Optimization Models and Algorithms for Structure Based Drug Design

by

Pınar Kahraman

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Industrial Engineering

Koc University

August 2008

ABSTRACT

Drug design is a highly expensive process that may take several years for a drug to be commercially available. Most of the effort for this purpose is spent on unsuccessful ligands wasting the resources and considerable amount of time. Therefore, developing computational tools for drug design has become one of the most widely studied areas in the past two decades. The aim of these studies is coming up with increased number of candidates while eliminating the deficient candidates beforehand. Structure-based drug design is a branch of such computational methods that is based on the fact that a drug has to be structurally and electrostatically compatible with its target protein's active site.

This thesis proposes various methods and algorithms for different stages of the structure-based drug design process. An algorithm called SLICE for the mathematical representation of active sites of proteins is presented. With this method, active sites can be represented as a union of convex hulls. The outputs are planned to be used as geometrical constraints for an optimization model for fragment-based drug design. For this model, ideas and propositions were presented in this thesis, including the necessary parameters, variables and constraints to be used. These are yet open to discussion and are merely offered as a starting point for the development of the model. In addition, a QSAR methodology for early prediction of drug activities is proposed. This study proposes a series of methods to be applied composed of a regression study, a home-made classification method based on hyper-boxes and significance testing.

The active site representation is applied on four different proteins and is observed to be covering the active site successfully for buried active sites, while a modified version of the algorithm is proposed for sites open to outer space. The QSAR method is applied on a dataset of 45 DHP derivatives and the methodology is seen to be providing more accurate results than the reference methods available in the literature.

ÖZETÇE

Bir ilacın tasarlanıp piyasaya sürülmesi, yıllar alabilen oldukça pahalı bir süreçtir. Emeğin çoğu başarısız molekülleri incelemek için harcanmakta; bu da kaynakların ve zamanın boşa gitmesine neden olabilmektedir. Bu nedenle, ilaç tasarımı için hesaplamalı metotlar geliştirme alanı son yirmi yıldır üzerinde çokça çalışılagelen bir konu olmuştur. Bu çalışmaların amacı, hem daha fazla ilaç adayı oluşturabilmek, hem de başarısız olacak adayları önceden belirleyip elemektir. Yapı bazlı ilaç tasarımı da bu tip hesaplamalı yöntemlerin bir dalı olup bir ilacın hedef enziminin aktif bölgesiyle yapısal ve elektrostatik uygunluğa sahip olması gerekliliğine dayanmaktadır.

Bu tez, yapı bazlı ilaç tasarımının çeşitli aşamaları için çeşitli metotlar ve algoritmalar önermektedir. Proteinlerin aktif bölgelerinin matematiksel ifadesi için SLICE adlı bir algoritma sunulmaktadır. Bu metotla aktif bölge, bir dışbükey örtüler birleşimi olarak ifade edilmektedir. Bu algoritmanın çıktıları, ileride ilaç tasarımı için oluşturulacak bir eniyileme modelinin geometrik kısıtları olarak kullanılabilecektir. Bu tezde, bahsedilen eniyileme modeli için fikirler ve öneriler sunulmakta, gerekli olacağı düşünülen değişkenler, parametreler ve kısıtlar sıralanmaktadır. Bunlar sonradan geliştirilmek üzere bu model için bir başlangıç noktası olması amacıyla oluşturulmuştur. Bunlara ek olarak, erken ilaç aktivitesi tahmini için bir "hesaplamalı yapı-aktivite ilişkisi" metodolojisi önerilmektedir. Bu çalışma, regresyon araştırması, çok boyutlu kutu bazlı bir sınıflandırma metodu ve anlamlılık sınamasından oluşan bir dizi metottan oluşmaktadır.

Aktif bölgelerin ifadesi dört ayrı protein üzerinde uygulanmış ve gömülü bölgelerin başarıyla kaplanabildiği görülmüş, tam gömülü olmayan bölgeler için ise algoritmanın modifiye edilmiş hali önerilmiştir. Hesaplamalı yapı-aktivite ilişikisi metodu 45 DHP türevi üzerinde uygulanmış, referans metotlardan daha doğru sınıflandırmalar elde edilmiştir.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis advisor Assoc. Prof. Metin Türkay for providing me the opportunity to work in an exciting and inspiring project and for his continuous support and guidance. I also thank my lab members: Uğur Kaplan for his bitter-sweet comments and his knowledge, Fadime Yüksektepe for her advices and beautiful personality, Pelin Armutlu for her friendship and assistance and Ali Öztürk for being the voice of the truth. I greatly appreciate my mom and dad's patience and would like to thank them for always trying to make me accommodate with the real world yet supporting all my decisions. I would like to thank my sister Ceren for understanding me and offering her precious advice. Lastly, I thank my life partner Alibey Öztürk for always being there. This thesis was completed with the support of TUBITAK scholarship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$A_{(HULL)p}$ parameter of the system of linear inequalities Ax + By + Cz ≤ D of plane *p* belonging to the $HULL^{th}$ convex hull.

$B_{(HULL)p}$ parameter of the system of linear inequalities Ax + By + Cz ≤ D of plane *p* belonging to the $HULL^{th}$ convex hull.

$C_{(HULL)p}$ parameter of the system of linear inequalities Ax + By + Cz ≤ D of plane *p* belonging to the $HULL^{th}$ convex hull.

$D_{(HULL)p}$ parameter of the system of linear inequalities Ax + By + Cz ≤ D of plane *p* belonging to the $HULL^{th}$ convex hull.

$V^{(i)}$ value of an ionic bond

$V^{(h)}$ values of a hydrogen bond

$V^{(vdW)}$ value of a van der Waals interaction

$ats_{i(at)}$ atom type *at* of surface atom *i*. (N,C,O,H,S…)

$atf_{jl\,(at)}$ atom type at of fragment j atom l. (N,C,O,H,S…)

$vdW_{(at)}$ vdW distances of every type of atom

mw_g molecular weight of fragment type *g*

Xa_i coordinates $\begin{bmatrix} xa_i \\ ya_i \\ za_i \end{bmatrix}$ of the surface atoms

$Ta^{(n)}{}_i$ binary value regarding surface atom *i*'s neutrality

$Ta^{(i)}{}_i$ binary value regarding surface atom *i*'s charge

$Ta^{(h)}{}_i$ binary value regarding surface atom *i*'s being a hydrogen bond acceptor or donor

$Tf^{(n)}_{gw}$	binary value regarding w^{th} atom of fragment g's neutrality
$Tf^{(i)}_{gw}$	binary value regarding w^{th} atom of fragment g's charge
$Tf^{(h)}_{gw}$	binary value regarding w^{th} atom of fragment g's being a hydrogen bond acceptor or donor
$mind^{(i)}$	minimum distance of ionic bonds
$mind^{(h)}$	minimum distance of hydrogen bonds
$mind^{(vdW)}$	minimum distance of van der Waals bonds
$maxd^{(i)}$	maximum distance of ionic bonds
$maxd^{(h)}$	minimum distance of hydrogen bonds
$maxd^{(vdW)}$	minimum distance of van der Waals bonds
$Dist^{(c)}_{ghcwx}$	distance between two bound atoms w and x of conformation c formed by fragment types g and h
θ_{ghcwxy}	bond angle between the three connected atoms s t and u of conformation c formed by fragment types g and h
$\omega_{ghcwxyz}$	torsional angle formed by the bound atoms s t u and v within conformation c formed by fragment types g and h
Rep_{ghcwx}	binary value regarding whether w^{th} atom of fragment g and x^{th} atom of fragment h is replaced to form c^{th} conformation of g and h or not
ffb_{ghcwx}	binary value regarding whether w^{th} atom of fragment g and x^{th} atom of fragment h make bond or not
$fb_{gw(w')}$	binary value regarding whether w^{th} and $(w')^{th}$ atoms of fragment type g are connected or not
D_{μ}	demand of atom μ

Chapter 1

INTRODUCTION

1.1. The Drug Design Process

According to the 16th report of World Health Organization (WHO) on drug dependence, a drug in the pharmacological sense is "any chemical agent that alters the biochemical physiological processes of tissues or organisms [1]. Drugs are substances used as corrective measures against deviances from normal biological processes. In general, they are designed to bind proteins (in most cases enzymes) having important roles in the metabolic pathway of a disease and to inhibit their activities.

Drug discovery is a cumbersome and expensive practice. To make a drug be commercially available, on the average, 10 drug candidates among millions of alternatives can survive to be tested on humans [2]. Before computational techniques were realized, the amount of time that was needed to commercialize a single drug was in the order of 5 years and the amount of money invested was in the order of $2,000,000,000 per drug approved.

The search for new drugs has considerably paced in the past decades thanks to the enhancing number of three dimensional protein structures on hand obtained by X-ray crystallography and NMR (nuclear magnetic resonance) spectroscopy methods, and to the advances in computational and visualization methods. Observing the structures of the active sites of proteins provides an understanding as to exploiting the Fisher's lock and key model for drug design purposes [3], where the "locks" correspond to the active sites and

"keys" are the ligands to be designed, which will fit the lock geometrically and electrostatically, and the complex should also have a favorable energy [4]. Therefore, as the number of structurally available proteins increases, the amount of knowledge gained by the structural information also increases, which allows researchers to be able to reduce the size of the molecular structure sets in their quest for ligands [5].

With the help of improved computational methods, rational drug design efforts considerably decrease the amount of time and resources used in the drug design process. Therefore, although there is need for improvement to perfect the process, structure-based drug design is widely used in academia and by industrial drug developers. In fact, it has been estimated that the computational efforts for this purpose necessitate exhausting more than 50% of the computer resources used in scientific research today [6].

In rational drug design, it is critical to develop an effective methodology to mimic the criteria regarding issues such as specificity and efficacy as to choose the right drug candidates among a large set of molecules. There are numerous computational drug design techniques that help diminishing the number of unsuccessful drug candidates prior to laboratory tests, which use different sets of such criteria in their algorithms. Methods such as QSAR (quantitative structure-activity relationship) and molecular docking are widely used by the pharmaceutical industry for this purpose [7]. All these computational methods aim to minimize the number of false positives and the number of missed successful candidates so that the time spent in lab phases is diminished and finally to have candidates that are successful enough for further clinical trials at hand. This study will focus on structure based computational drug design, since the developed design method falls in this area of research.

1.2. Structure Based Drug Design

1.2.1. The Role of the Active Site

A protein's function depends on the interactions it makes with other proteins and ligands [8]. A drug molecule is designed to bind to its target protein. The binding process takes place in the "active site" of the target, which is in general a pocket at the surface and holds the residues that are accountable for the substrate specificity (in terms of charge, steric interference, hydrophobicity) and the catalytic residues that are accountable for cofactor binding or that contain protons donors or acceptors [9]. Generally, the drug binds to the active site in order to inhibit another important binding process, and this course of action lies in the basis of structure based drug design. In order to efficiently utilize the "lock and key" model that mimics the structural compatibility of the binding partners, firstly, the structure of the "lock" must thoroughly be examined. X-ray crystallography and NMR spectroscopy methods yield structures of proteins in high resolutions, allowing researchers to have detailed models of the active sites [3].

Before the actual design process begins, there are two crucial steps: one of them is the procedure explained above, i.e. the identification of the potential active site of the target molecule, and the other is the description and visualization of these sites. The target may be an enzyme whose active site is already known. But if this is not the case, discovering new sites and studying their appropriateness for ligand binding may be necessary [10]. It is suggested that on a certain protein, there may be more than one active site to target. This is due to the fact that most proteins have more than one function and more than one binding partner with different active sites [11]. Therefore, for drug design purposes, the selected site may be an active site, in which catalytic activity takes place, such as the ones in

enzymes. The selected site may also be a site used for communicating with other molecules necessary for the molecule to perform its activities [3,5,7]. Therefore, it is important to wisely decide which site to choose.

Once the target site is selected, the geometrical representation of this pocket is another significant step in the structure based drug design process. The volume and the accessible surface area of the pockets should be precisely defined and represented, since they define the three dimensional space that the drug molecule should bind. For instance, representing the molecule in the stick and ball mode, where atoms are represented by hard spheres and the connections are represented by lines, provides very little information about the shape, and the volume of the active site [4]. There are a number of different representations of accessible surfaces in the literature all of which trying to capture the three dimensional space as closely as possible, since the drug molecule should fit in this space like a hand in a glove so that the affinity is maximized [10]. Active site representations in the literature is discussed in the Literature Review section.

1.2.2. Methods of Structure Based Drug Design

It can be said that there are three approaches to in silico drug design: inspection, virtual screening and de novo drug design. In all these methods, the yields are not finalized drug products, but are compounds that have at least micro-molar affinity for the target molecule [7]. The lead is then modified in order to get rid of its undesired properties such as toxicity or insolubility, and finally is subjected to clinical trials [2]. In the first approach, generally a realized drug molecule that is known to fit the active site of the target protein is slightly modified for maximizing the affinity by inspection, hoping that the new compounds will also have favorable properties [7]. In the second method, a database of available small molecules, which are generally modifications of certain template models, are docked into

the active site. Although virtual screening used to utilize libraries containing thousands of molecules, the trend today is more and more towards designing small libraries that are focused and specially designed for purpose, i.e. towards non-random screening [5,12]. The third approach, de novo drug design, is one of the most widely studied areas in drug discovery today. In this approach, the knowledge on the structure of the target molecule is used in order to rationally develop novel "lead" molecules by bringing small molecules called "fragments" together and optimizing their conformations. In addition, there are a few algorithms in the literature that handle de novo design by using atoms as building blocks [13].

In general, a main challenge in structure based drug design is predicting the orientation of the ligand, called "molecular docking", and calculating the affinity of the complex. Therefore, docking algorithms with different scoring and optimization approaches are used in both virtual screening and de novo design methods. Assuming that the three dimensional structure of the target molecule is known, docking methods aim to design and optimize a ligand structure that will fit in this active site [14]. Today, there are many docking algorithms, most of which sharing similar methodologies bearing original extensions [13]. Major algorithms, their methodologies, similarities and differences will be discussed in the Literature Review section.

1.2.3. Apriori Analyses

Prediction of designed ligands biological activities before they are subjected to lab tests is an important problem in drug design. Since the amount of resources spent on a drug to be commercialized is really high, the possibility of wasting these resources on likely unsuccessful candidates should be decreased. QSAR (quantitative structure-activity relationship) is one of such methods and aims to build regression models that describe the

activities of drugs based on experimental data and to extract activities of molecules based on these models [15]. The method is based on data-mining and is used frequently in the literature for early activity prediction purposes.

In this thesis, a new QSAR methodology is presented. Studies that have conducted QSAR research on the data set that the proposed methodology uses are explored in the Literature Review chapter.

1.3. Contributions

This thesis presents new methods and algorithms for the different stages of structure based drug design. For the initial stage, a novel mathematical representation of active sites is presented. Then, for the design stage, a fragment library is proposed, which is to be used as the input of a proposed but yet-to-be-tested MINLP model for de novo design. The offered model aims to minimize the energy of the binding process. Finally, as the final stage, a QSAR method based on a home-made classification algorithm for drug effectiveness studies for elimination of unpromising drug candidates before the laboratory phase is presented.

As explained in previous sections, the first step in structure based drug design is to identify and represent the active site of the target protein. For this purpose, this thesis presents a novel mathematical representation of target active sites based on convex hulls is presented. There are different approaches used for representing the active site in the structure-based drug design literature, such as alpha-shapes, Delaunay triangulation or dot surfaces [4]. However, to our knowledge, no algorithm has utilized convex hulls as representative blocks. The method in this study uses a greedy heuristic algorithm that aims to cover the accessible surface of the site by conjoint convex hulls, and characterizes the

space by linear inequalities, which are used as the geometric constraints of the energy minimization model.

In drug design studies, once the representation of the target pocket is accomplished, the method to be used in the design must be decided. Therefore, following the representation, the main idea about a new fragment based optimization model is proposed and the fragment library built for this purpose is presented. It should, however, be noted that the proposed model is in the form of a concept, which is basically a list of ideas that the MINLP model can be based on.

The proposed drug design method is aimed to be an optimization model itself, which can be claimed to be a new approach in structure based drug design. Although operations research has been widely used for drug design purposes, an example to which is the genetic algorithm (GA) used by many docking algorithms to prune genetically undesired random instances of docking, all such uses of OR make iteration decisions within the main algorithm of the drug design method, whereas this study proposes usage of optimization as the ligand building process itself. The idea of the MINLP model proposed in this thesis seeks to construct energetically favored leads by selecting and binding a set of molecular fragments subject to a set of geometric and chemical constraints, and having an objective function that tries to maximize the affinity and minimize the energy of the complex. The fragment set that is composed for this purpose is composed of bound couples of small fragments that are claimed to be found frequently in ligands. The inter-atomic distances, bond distances and torsional distances were also calculated to be used in the proposed optimization model.

After developing candidate ligands, the scientist may want to computationally test the compounds activities to lower the number of molecules to be sent to laboratory. Thus, lastly, this study puts forward an early prediction method to be implemented once the candidate molecules are designed for further elimination of molecules with undesired

chemical and biological characteristics so that the number of molecules to be clinically tested is reduced. For this purpose, a new QSAR approach is proposed, which uses an integer programming algorithm [17] based on hyper-boxes for the classification phase. The approach comprises an iteratively run series of optimizations and is a novel procedure that has achieved good accuracy results [15].

Chapter 2

LITERATURE REVIEW

2.1 Representations of the Active Site

Drugs are small molecules designed to inhibit certain undesired reactions within our body. Although they might have various types of targets in molecular level, most try to bind a protein, be it an enzyme of a virus breaking down the DNA, or a protein that performs an important function in the pathway that the disease attacks. Therefore, revelation of protein structures has great importance for drug design efforts. A lot of resource and time are dedicated to solely revealing these structures [7].

Some proteins are the building blocks of cells and some carry out important functions in metabolism. The various functions that proteins perform are dictated by their three dimensional conformations. However, although having various shapes, all proteins are composed of the same group of units called amino acids. Amino acids are organic molecules that are composed of an amino, a corboxyl, a hydrogen atom and a radical side chain group that are bound to a central alpha carbon atom (see Figure 2.1). There are 20 types of amino acids and this variety is due to the types of radical groups (side chains). In a protein, amino acids are joined together by peptide bonds to form a polypeptide chain. In the drug binding process, these bonds do not break, since they are covalent bonds; rather, noncovalent bonds are formed between the atoms of the drug molecule and the atoms of the side chains lying on the *active site* of the protein [18].

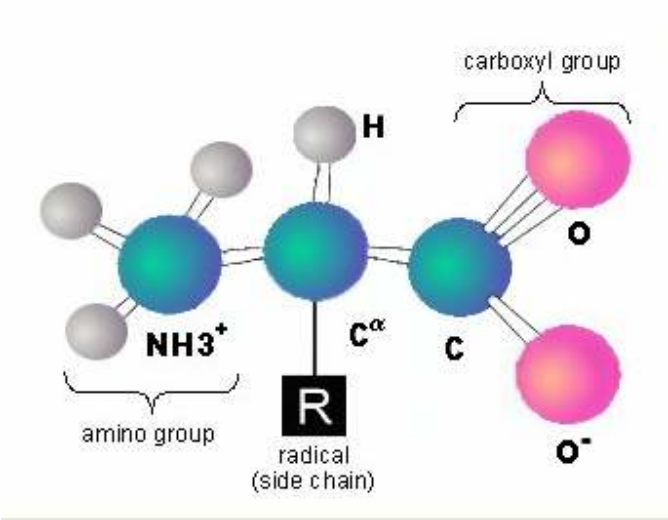


Figure 2.1: Sketch of an amino acid.

Active sites of proteins are generally small pockets on the surface of proteins. As explained in the Introduction section, a drug should fit in the active site of the target protein like a hand in a glove to perform its expected functions. The ligand and protein molecules should recognize each other by chemical attractions and also by compatibility of their structures. Thus, considering the significance of this recognition process, one can easily acknowledge the fact that the starting point of any structure based drug design is the active site representation process [19].

There are many algorithms that aim to represent the active site as accurately as possible. As explained below, most of these algorithms divide the space and form a grid to see where atoms are present and where not. Some algorithms are based on purely geometric criteria while others aim to add some physical meaning to the representation process.

Active site representation methods mostly rely on crystallized structures of macromolecules and their ligands obtained by X-ray crystallography and NMR

spectroscopy methods [20]. These structures are deposited in the Protein Databank (PDB) [21] developed by Rutgers University, New Jersey, and University of California, San Diego, which provides three dimensional biological macromolecular structures and tools to examine their associations to sequence, function, and disease. There are 51155 structures as of June 3rd, 2008 in the database.

There are three sources of information that can be used to infer ligand active site locations and shapes: protein structure, evolutionary information (sequence alignments) and ligand/substrate information [22]. If the structure of the target molecule is not known, homology modeling can be used. In this procedure, a 3D model of the target protein is inferred from amino acid sequences of structurally similar (homologous) proteins [23]. However, if the structure of the macromolecule is available but the crystal of its ligand is absent, the active site needs to be predicted. There are various methods that accomplish this task by different approaches. One approach may be identifying the "hot spots" on the surfaces, which certain functional groups may favor in binding [19]. Other algorithms may calculate pockets by geometrically finding out spaces larger than a certain threshold on the surface.

In this section, major active site representation software and their methodologies are summarized.

2.1.1 Methods for Representation of Active Sites

The program SURFNET [24] provides three dimensional contour surfaces using the density distributions of three dimensional data points. The algorithm was developed by Roman A. Laskowski in 1995 and is one of the first predictive models for active site determination. The program can be used to determine the locations and shapes of pockets and cavities of a protein as well as the space between two interacting macromolecules, such as a protein and its ligand.

Gap regions between and within molecules are identified by fitting spheres in spaces between atoms. For two molecules, a sphere between each atom pair (one from one molecule and one from the other) is placed such that it is tangent to the spheres formed by the van der Waals volumes of the atoms. For an atom pair A and B, first the sphere tangent to this pair is formed, then, if the sphere overlaps with neighboring van der Waals volumes, it is shrunk so that it does not collapse with any atom but touches at least one. Spheres having radii smaller than 1A° are rejected. For each pair, this procedure is repeated. The result is a set of spheres that represent the gap between the two molecules. For representation of the cavities of one molecule, the same procedure is valid, only the pairs now belong to the same molecule.

After forming the gap spheres, a three dimensional grid is formed, which contains the molecule, and grid density values for an atom are assigned to every grid point using the Gaussian density function:

$$\rho = \rho_0 e^{-kr^2} \qquad (2.1)$$

where ρ is the density of a grid point that is r away from the atom, ρ_0 is the density of the atom center and k is a constant calculated such that the ρ value is 100 at the van der Waals radius and 200 at the center, i.e.

$$k = \frac{\ln 2}{r^2_{vdW}}. \qquad (2.2)$$

If there are covalently bonded atoms in the immediate neighborhood of the grid point, whose van der Waals volumes overlap, then the grid density of a grid point is the density value of the atom that has the maximum density value among all atom spheres that fall in that grid point.

The program takes this density map to build the contours, and the surface of the gap region is obtained by forming the 3D contour surface built by polygons made up of all the interpenetrating gap spheres. In this way, the size, and the shape of pockets and cavities are identified.

POCKET [25] is one of the first algorithms that can identify pocket sites and cavities without the need for specifying some seed points that indicate the locations of the indentations. The program automatically finds and displays the cavities as a collection of triangles and identifies their surrounding amino acids.

The method commences by assigning densities to every (*x,y,z*) position by moving a sphere of radius *r* in the x direction, y direction and z direction separately in discrete steps of size δ and checking if the center of any atom in the protein falls into the sphere in each position. Then, the regions with contact and no contact are checked to see the path and regions of no contact surrounded contact regions are assigned a value of 1, and the remaining space is assigned a value of 0. The pocket regions are accepted to be the regions with assignments of 1.

After the density map is obtained, a modified version of the marching cubes algorithm of Knox [26]. The macromolecule is divided into cubes of size δ x δ x δ, and then the vertices are checked to identify different surface indentations. Each cube is assigned a value to indicate to which indentation the cube belongs to. A density value of 0 in one of the vertices indicates that a surface passes through the cube. If there is such a vertex, other

vertices are checked to see if there is a value of 0, and if there is such a vertex, value of the neighbor cube, which share the same vertex is checked to see if the neighbor belongs to another surface. In this case, the value of the neighbor is assigned to the current cube. If there is no such neighbor, a new value is assigned. At later iterations, some surfaces that belong to the same pocket will merge. Such a case is observed when a cube has more than one vertex that is neighbors of cubes of different assignments. Let there be three different assignments a, b and c of the cube. Then, a merging procedure is applied, which is changing the values of all the cubes that have values a, b and c to a same value. Having assigned the cube vertices, the surface is modeled as a set of triangles. Also, the volume of the macromolecule is provided using the following formula:

$$V = \int_S x.nds \qquad (2.3)$$

where x is the x component of vector defining the center of a triangle, and n is the outward normal of the triangle

The program is used for identifying and visualizing the pockets and cavities, calculating their volume and finding the amino acids surrounding them. These are all accomplished without the need of any reference point, i.e., the only input that the algorithm needs is the PDB coordinates of the molecule.

LIGSITE [27], a predictive program developed in 1998 by Hendlich et al., is an extension of POCKET [25]. The prediction procedure of LIGSITE also begins with the scanning algorithm of POCKET. However, in POCKET, the grid is scanned only in x, y and z axes, making the program miss the pockets with an orientation of 45° to these three axes. Therefore, it can be said that the success of POCKET depends on the orientation of the protein in the grid. LIGSITE gets rid of this problem by scanning the grid also along the four cubic diagonals.

At the beginning, the program assigns all of the cubes in the grid a 0 value, indicating solvent accessible areas. With the scanning, the cubes containing a protein atom is then

given a value of -1, whereas a solvent accessible cube's value is increased by 1 when the cube is within two solvent inaccessible regions in each axis or diagonal scan. Therefore, the solvent accessible areas have values ranging from 0 (not buried in the molecule) and 7 (deeply buried). Pockets are determined as neighbor cubes having values more than a threshold value (2 is the default threshold of LISGSITE). Cavities are special regions which are composed of 7-valued cubes.

The procedures for the geometrical representation of the surfaces and the calculation of the volumes are the same as in POCKET. The main advantages of this program compared to POCKET are its speed and a more accurate prediction algorithm. LIGSITE can distinguish pockets from cavities, and also has the adjustable threshold and step-size features, which enable the user to avoid or encourage identifying small bumps and to choose between a better representation and a faster response, respectively.

Being an implementation of LigSite, Q-Site Finder [8] also aims to find out cavities of a given macromolecule for docking and drug design purposes. The program especially focuses on the problem of dependence of the output pockets' sizes on the protein's size, which is claimed to be present in LigSite, POCKET, and in some other algorithms. As in LigSite, Q-Site Finder divides the protein into grid points and the van der Waals energy of a methyl probe in each grid point is calculated. What differentiates the program from its references is the clustering algorithm embedded in the main algorithm. The probes that have energies above a threshold are selected and are clustered according to their closeness in space. Then the total energy of each cluster is calculated, the clusters are ranked according to these values, and the most favorable cluster is picked as the first predicted active site.

The aim of Q-Site Finder is to keep the volumes of predicted pocket sites as small as possible while maintaining the accuracy. Thanks to the new clustering algorithm and better choice of parameters, Q-Site Finder is a more accurate and a more realistic program than its

reference algorithms in the sense that it catches a high percentage of the pockets while minimizing the number of false positives. Purely geometrical algorithms tend to find larger pockets as the size of the protein gets larger, which decreases accuracy because the majority of the ligands are small molecules. Q-Site Finder takes care of this problem with its energy-based algorithm.

PASS (Putative Active Sites with Spheres) [28] is a computational tool that was developed by G.P. Brady Jr. and P.F.W. Stouten for the DuPont Pharmaceuticals Company. The main aim of the tool is to predict and visualize all potential active sites of a protein. The software is used as a front-end to virtual screening studies, predicting different active sites. Therefore, it can be used when the active site is unknown, or alternative ones are sought. The algorithm is fast, taking 20 seconds for a moderate sized protein to come up with a visual output of the active site.

This method describes the active site as a set of probe spheres. First, the PDB file of the protein is read assigning elemental atomic radii and coordinates to the atoms. Second, probe spheres are placed by solving the following problem for each unique tripartite atom set of the protein: given the three atoms i, j, k with positions P_i, P_j, P_k, and radii σ_i, σ_j, σ_k, find position R_p of a probe sphere and its radius σ_p such that the sphere is exactly tangent to all three atoms. With this step repeated for all possibilities, the protein is covered with probe spheres. Third, the spheres are filtered according to three rules: the sphere should not overlap with the accretion substrate's atoms, it should not collide with the protein's atoms, and it should be buried. The third rule is tested by counting the number of atoms that sits within a 8°A radius, and the spheres having this value below an empirically determined threshold are filtered out. The remaining spheres are accreted onto a scaffold built by the previous probe spheres, new spheres are added to fill the empty spaces opened due to filtration, and then filtered again. This procedure is repeated until no newly added sphere survives the filter. Finally, the algorithm ends having filled all cavities and the

empty spaces lying in the protein with a set of evenly separated, buried probe spheres, none of which sterically clashes with the protein. The program then smoothes the spheres; renders and colors the protein, the spheres, and ligands.

PASS is a geometric visualization program, which allows the user to gain an insight of the volume and the three dimensional shape of the protein cavities, later to be used in docking or be interfaced with functional genomics tools.

CAST [29] was developed by Liang et al. in 1997, which is a program that extends related programs of Edelsbrunner and colleagues and aims to identify pockets and cavities of proteins and to gain insight about quantitative measures of the three dimensional shapes of these structures. The output of the program is comprised of the volume, the surface area of each concavity, and the atoms that build them, as well as the atoms that construct the mouths of pockets, the surface area and the perimeter of these mouths. Features that were newly developed and not present in previous methods of Edelsbrunner et al. are the surface area and volume calculations.

The algorithm uses the Delaunay triangulation [30], the alpha shape (or dual complex) of the triangulation, and the discrete flow method. The difference between the Delaunay tessellation and the alpha shape of atom centers of the protein in three dimensions gives the cavities. Having found the occupied and empty tetrahedra with these methods, the discrete flow method pours the “obtuse” empty tetrahedra into “acute” empty tetrahedra, merges them, and provides the pocket sites. The difference between obtuse and acute empty tetrahedra is that none of the surfaces of the acute empty tetrahedra contain an edge on one of the lines that unite atom center couples, which are drawn during the alpha shape construction phase, while some of the surfaces of the obtuse empty tetrahedra contain such edges. The mouths of the pockets are also found in this phase, which are the outer surfaces of the pockets that face the empty space. The details of the alpha shape and the discrete flow methods can be found in [31].

The program makes use of available programs. Atomic radii are assigned by using PDB2ALF, 3D weighted Delaunay tessellation is made by DELCX, and to compute the alpha shape (dual complex), MKALF is utilized [31]. Then CAST takes the output and computes the volume and surface area of pockets and cavities, surface area and circumference of the mouth regions. A visualization program can be used to visualize these structures by using the output that CAST provides.

To conclude, CAST can identify and measure most probable active sites of proteins without ligands as well as the unoccupied space in active sites for drug design purposes, and the description of these sites is in the form of a union of three dimensional tetrahedra.

Coleman and Sharp from the University of Pennsylvania developed the algorithm TravelDepth [32] in 2006, which quantitatively describes the shapes of the surface indentations. The algorithm finds the "non_Euclidean macromolecule avoiding" minimum distance of every surface point to a reference point. The output is a discretized approximation of the actual three dimensional shape of the protein. Although they are unbounded in one direction, the algorithm can be used to describe tunnels and also DNA grooves; nevertheless, disregards the cavities that are closed empty spaces in proteins.

"Travel depth" is defined as the minimum distance a solvent probe has to travel from a surface point to the reference point through the solvent. Finding this distance is equivalent to the shortest path problem, which is a very well known NP hard problem in literature. The protein is turned into a graph by discretizing the structure into grid cubes. However, before discretizing, all cavities are removed, since the probe should travel through the molecule to reach to the reference point. The grid spacing should be small enough such that every atom of the macromolecule is represented by at least one grid point and any concave indent in the surface is represented by at least one grid center. The TravelDepth algorithm takes the grid spacing to be 1A° for this purpose.

The convex hull of the macromolecule is calculated by using the readily available algorithm QuickHull [33] and then, the center of every grid cube is checked to see whether it is inside the convex hull or not. The ones outside the convex hull are classified as the O class, and the ones inside the convex hull is checked once again to classify them into I, S and B classes, which represent the cubes inside the molecular surface and not containing any points of the surface, the cubes inside the molecular surface but containing a surface point, and the cubes between the surface and the convex hull borders respectively. After this classification, a careful definition of neighbors is made for each class type to make sure that depth spreads into the molecular surface, i.e. into the S class cubes, but not through the molecule, that is into the I class cubes. The algorithm then assigns a distance of zero for the O class and indefinitely large distances for B and S classes. Then, for every neighbor j, distance of a cube i (d_i) is calculated by calculating the sum of the distance of its neighbor j (d_j), plus the distance between i and j, and choosing the minimum among them. That is:

$$d_i = \min_j (d_j + dist(i, j)) \qquad (2.4)$$

where j is an element of all neighbors of i.

The calculation begins with the O classes, assigning them a distance of zero, and recursively assigning distances to their neighbors and then to the neighbors of their neighbors and so on. For the assignment process, the program utilizes Dijkstra's shortest path algorithm [34].

The algorithm is fast and reliable, which was tested on various known structures. The method quantitatively defines the binding pocket shapes and therefore can be used as a preprocessor for library screening purposes or for ligand docking.

DrugSite [22] is another geometry-based active site identification program, which was developed by An et al. in the year 2004. The major contribution of the program is that it can differentiate "druggable" active sites from the other pockets according to certain threshold parameters, which can be tuned by the user based on observations. The

algorithm computes the van der Waals potential map on a grid, adding a physical sense to the process while not requiring any chemical structural information of the ligand.

In the algorithm, first the ligand and the water molecules are excluded from the PDB structure. Then, a grid is formed and for each grid point, the algorithm first calculates the cumulative potential using the Lennard-Jones potential formula:

$$P_i^0 = \sum_{j=1}^{N} \left(\frac{A_{jC}}{r_{ij}^{12}} - \frac{B_{jC}}{r_{ij}^{6}} \right) \qquad (2.5)$$

where i is a member of the set of grid points, j is a member of the set of atoms, r_{ij} is the distance between point i and atom j, and A_{jc} and B_{jc} are calculated according to the ECEPP/3 molecular mechanics force field. The motivation of this calculation is based on the fact that van der Waals forces are lower close to the pocket area, since there are no or a few atoms in the neighborhood of the grid points in this area. After the energy map is created, it is space-averaged ten times to stress the regions with higher values. The pockets are then obtained by contouring the map according to the following contouring level:

$$level = mean(map) - threshold * rmsd(map) \quad (2.3)$$

where the threshold is an empirically designed value that affects the size of the pockets obtained, and rmsd stands for the root mean square deviation.

After coming up with the pockets, the program deletes the ones smaller than 100Å, since the authors have examined from available data in the PDB that ligands tend to be larger than this size. DrugSite characterizes the pocket sites by triangulating the grid space. The output then can be utilized in docking and de novo drug design studies. The algorithm also finds alternative druggable active sites for a protein.

The authors came with another algorithm called the PocketFinder [35] in 2005, one year after they developed the DrugSite. The program identifies the pocket sites by utilizing only the protein structure as an input. In the algorithm, the van der Waals map is calculated just like DrugSite does, and the pocket envelopes are created according to the same criteria.

The novelty in this algorithm is the capability to predict the actual active sites of proteins by classifying the pockets according to their shapes and physiochemical properties.

The latest algorithm to our knowledge that identifies and describes active sites of proteins is the PocketPicker [36], which was developed by Schneider et al. in December 2006. The program is a grid based detection method that is able to translate the shape information and the buriedness of an identified pocket into correlation vectors.

Schneider et al. use grids to calculate how buried the concavity is by observing the neighborhood of every grid. The shape and buriedness information is stored in descriptors, which are specifically designed for easy comparison of various pocket site conformations. The method first takes the protein in a rectangular grid having mesh size of 1Å . For each probe, the algorithm searches for neighbor atoms within 30 rays having a size of 10Å x 9Å, originating from the probe and equally dividing the sphere, calculates the number of atoms that fall in these regions to classify the probe into six clusters with different buriedness indices.

Being able to compare the shape descriptors is one of the most useful functions of the algorithm. The method compares two structures by creating 20 distance bins that range from 1Å to 20Å for every 21 bipartite combinations of the two molecules' six classes A, B, C, D, E and F. Therefore, the method comes up with shape descriptors having 20 x 21 = 420 dimensions. Then the pocket shapes are compared by calculating the distance *d* between the two structures *r* and *s* using the following formula:

$$d = \sqrt{\sum_{i=1}^{420} (r_i - s_i)^2} \qquad (2.6)$$

where *i* is an element of the 420-dimensional descriptor. The algorithm is proved to be a promising tool for the representation and the comparison of active sites, providing better resolutions with the 30 directional scanning algorithm and outperforming most of the

algorithms that rely on computational geometry, such as CAST, PASS and SURFNET in terms of reliability.

2.2 Structure Based Drug Design

2.2.1 Drug Design Principles

Designing a successful drug is not a trivial effort, since it has to meet many conditions. Firstly, the molecule should exhibit favorable target selectivity, i.e. the designed drug should not bind to molecules that have important metabolic duties, but only to its target. If the drug binds to enzymes that carry out vital tasks, side affects occur. Also, the drug should be "orally bioavailable", that is the molecule should not be too large, should have membrane permeability, and should be soluble to be absorbed. Moreover, the toxicity levels of the drug should be minimized [37]. These characteristics of a *drug-like* molecule are called the ADME (absorption, distribution, metabolism and excretion) properties

Lipinski et al. [38] published threshold values for parameters of four characteristics marking the absorbability and permeability of the molecules in 1997, and these rules have been accepted by the drug design literature as the starting filter of drug candidates. According to the Lipinski's "rule of five", a drug-like molecule should have:

- not more than 5 hydrogen bond donors
- not more than 10 hydrogen bond acceptors
- a molecular weight under 500 g/mol
- a partition coefficient log P less than 5

It should be noted that Lipinski's properties are not adequate for a molecule to be considered as a successful drug, but are the necessary conditions for a molecule to be

regarded as a drug *candidate* [39]. The first two rules are about the atom types, the third rule does not allow the molecule to be very large and the last one is about the solubility of the molecule.

After the pioneering work by Lipinski, many other properties that affect oral bioavailability are discussed such as molecular flexibility, or polarity of the surface area [40].

Another feature called the *binding affinity* has major importance for computational drug design, which is correlated with the spontaneity and energetical favorability of the binding process. Affinity is strongly related to the enthalpy change (ΔH) and the entropy change (ΔS) due to the binding process, and is directly related to the Gibbs free energy of binding (ΔG), where

$$\Delta G = \Delta H - T\Delta S \qquad (2.7)$$

Here, T refers to the temperature that the process takes place in.

For a favorable binding process, affinity needs to be maximized, i.e. ΔG needs to be minimized. Ernesto Freire from the Johns Hopkins University reports in Protein Reviews that this value can be decreased by arbitrary values of enthalpy and entropy, but that a "strong favorable binding enthalpy is crucial" based on the data on HIV-1 protease inhibitors, implying that the weight of enthalpy change is more than the entropy change in binding affinity [37]. That is, in most cases, a minimized enthalpy is preferred over a maximized entropy. This makes sense when the stability issue is contemplated. Maximized entropy means more flexibility and spontaneity, however, this value should not be too high, since the complex should be stable to some extent; otherwise, the affect of the drug will not be in the desired levels. In fact, in rational drug design, more practical properties begin to enter the picture, such as the easiness of formulation, availability, stability and crystallinity of the molecule [40].

2.2.2 Molecular Interactions

As explained above, a successful drug has a strong affinity with its target. One way of increasing affinity is to maximize the number of noncovalent bonds between the target and the ligand. In theory, the more atoms the ligand molecule has, the more is the stability; however, the ADME properties say that a molecule that large cannot be a good drug candidate. Therefore, it is crucial for a drug design method not to violate the Lipinski constraints while maximizing affinity.

There are three types of noncovalent interactions: ionic, hydrogen and van der Waals interactions. Ionic bonds are built by an electron transfer between oppositely charged atoms. Hydrogen bonds, on the other hand, are formed by polarization between a hydrogen donor (or a hydrogen atom bonded to a N, F or I atom) and a hydrogen acceptor, which is electronegative in nature. Van der Waals interaction occurs between every atom couple that are close enough. This proximity is different for every atom type, and is called the van der Waals distance [41].

The strength of a bond is measured by the energy needed to break the interaction: the more this energy, the stronger the bond. The energies of the noncovalent bonds vary from atom to atom, however it can be claimed for an “average atom” that the strongest attraction among these is formed by an ionic bond, with an approximate energy of 3 kcal/mole in water. Hydrogen bonds bear about one third of this energy, and van der Waals interactions are quite weak, having energy approximately one tenth of ionic bonds have [42]. It can be said that the stability of the complex depends on the total strength of the non-covalent bonds that it bears.

The draft model that is presented in this thesis is attempted to be built on these interactions, aiming maximized affinity between the designed ligand and the active site of

its target protein. On the other hand, by taking Lipinski's rule of five into consideration, the designed ligand is supposed to be prevented from being an unrealistic candidate.

2.2.3 Fragment-Based de Novo Drug Design Algorithms

The idea of an optimization model for drug design purposes that is aimed to be initiated with this thesis is based on designing ligands by combining fragments together by optimizing the affinity of the whole structure. Therefore, it is seen necessary to present here some of the major fragment-based design methods from the literature.

In the late 80s, a new approach based on computational de novo ligand design was born and was considered as an alternative to the high throughput screening methods, which involve exhaustively docking a database of molecules and searching for good drug candidates among them. Today, there are numerous de novo design algorithms that use various types of building blocks, search methods and scoring functions [43]. In all these algorithms, the purpose is to come up with ligands having favorable pharmacokinetic properties from scratch using atoms or more frequently molecular fragments, which is a completely different approach than screening, and is more flexible than modifying some existing drug molecule. However, this flexibility comes with the price of increased combinatorial difficulty. Not only there exist multiple topologies, but also a single one has multiple conformations [44]. Therefore, the criteria and the optimization techniques used in the filtration process has major importance in de novo design.

This section takes a look at a selection of fragment-based de novo algorithms and the methods they use.

LUDI [45] is a fragment-based design algorithm, which was developed by Böhm in 1991. The program basically aims to build hydrogen bonds between the active site atoms and fragments from a fragment library first, and then tries to connect the fragments together

by different kinds of fragments called “bridges” from another library. It requires the coordinates of the protein from its PDB structure, the position of the active site, and the possible bonding partners within the active site as input. Having these at hand, the algorithm separates the different types of possible bonding partners into aliphatic, aromatic, hydrogen donors and acceptors. Then it draws their influence regions in many possible conformations, which are limited in number and are predefined according to statistical data about bond distance and angle preferences. The program also take advantage of another program called GRID, and uses it to determine the energetically favorable sites on the surface that fragments might be placed. These too are taken into consideration as well as the outcome achieved from the rule based approach. Then all are checked if there is a van der Waals trespass. The regions that pass this test are called “active sites”.

The algorithm then initiates the fragment binding process. It first checks for couples, triplets or quadruples of these active sites that are close enough to each other for multiple atoms of a fragment to bind. The fragment library is composed of fragments in their energetically optimized conformations, which are calculated using available energy field CVFF [46]. However, the program allows new fragments to be added to the library if needed. Fragments are treated as rigid bodies, but different conformations of a single fragment are included in the library, which provides a certain level of flexibility. Then the program fits the fragments that suits to each candidate active site group from their pre-defined “promising” atoms. The algorithm allows fitting multiple fragments to a region, and then the user needs to select the most desirable ones by hand.

The last step is to connect the bound fragments. The program accomplishes it by marking the close fragments and their closest hydrogen atoms with their bound heavy atoms and uses these to select the bridge fragment to be connected to these heavy atoms. The bridge fragments are selected from a separate library through a similar fashion.

HOOK [47] requires the coordinates of the active site, the description of the functional regions on the active site and a database of “skeletons”, whose conformations are known, as inputs. The main idea of the program is to place fragments called skeletons in the active site and then “hook” these skeletons to the surface by building non-covalent interactions. Skeletons have predefined hooks on themselves to be bound to the functional groups.

The program uses three types of probes in order to discover possible attraction sites: a hydrogen acceptor, a hydrogen donor and a hydrophobic group. The grids achieving attractions below a certain level for each type of probes are labeled as “vacant”. The grids achieving favorable attractions are labeled as “donor”, “acceptor” and “hydrophobic”. Each group has a certain CHARMM [48] potential and this is used for later assessment of fragment bindings. The algorithm compares the geometry of the hooks on skeletons and the geometry of the functional sites, then overlaps these and computes an overlap score by using Lennard-Jones potential, where a certain threshold should be met for the interaction not to be rejected. It checks for possible clashes of the skeleton with the protein, and rejects the conformation if there exist such collusions. Then, finally, the unbound regions of the skeleton are bound to the functional groups by using extra carbon atoms.

Another fragment-based de novo design algorithm is LigBuilder [20] developed in Peking University in 1999. Initially, the algorithm uses POCKET [25] to analyze the protein and to determine the binding, which is then divided into grids. The method used for detecting the functional areas of the active site is the same as HOOK uses.

After detecting the attractive sites, the algorithm starts to select fragments to be bound. The structure should have an initially bound molecule in its active site in order for the program to start the fragment selection procedure. Hydrogen atoms on this molecule and on the fragments are determined, and a selected fragment is bound to the molecule by building a covalent bond between the heavy atoms connected to the hydrogen atoms. Then an energetically favorable conformation is searched by turning the bond by 15° increments.

All conformations having local minima are kept in memory and are treated as different molecules. Then the resulting structure is checked if it bumps into the protein, and is rejected if so. The molecule is also subjected to a knowledge based set of rules about the reasonability of its chemistry, and also to Lipinski's rules.

Because every favorable conformation is treated as a different molecule, the number of possible structures built by the program is huge. The growing procedure should therefore be controlled, and this is achieved by a genetic algorithm. The new generation is created by the elitism algorithm, where 10% of the best members of the old population is kept in the new generation, and the remaining is achieved by randomly selecting fragments from the library. With this approach, the average fitness of the population is increasing in each iteration. The favorability of the structures is measured by the ΔG value. The program then stops when a user defined number of iterations is reached. The last generation is the candidate ligands at hand.

CONCERTS [49] by Pearlman et al. (1995) has a different methodology then the algorithms above to some extent. The main difference is that CONCERTS conduct molecular dynamics on a database of fragments together within the active site, where the fragments do not "see" each other, i.e. they do not interact with each other, but only with the active site. At predetermined intervals, fragments that are in favorable conformations are tried to be bound to each other. However, these connections are not final; they can be broken for building more energetically favorable connections in later iterations. In the case that all fragments have been checked for possible more favorable interactions, the algorithm stops when a certain number of steps is reached.

2.3 QSAR

A lot of resource is spent for a drug candidate to be tested in laboratory, and all is wasted if the molecule turns out to be toxic or having adverse pharmacokinetic features. Therefore, predicting the activities of drug candidates before lab phases is an important problem in computational drug design, since there may be a tremendously large number of candidates among which many unsuccessful instances exist [50].

Data-driven methods used for this purpose derive their prediction models from experimental data [51]. QSAR (quantitative structure-activity relationship) is one of such methods that aim to build correlations between the chemical structure of the molecule and its activities like reaction ability, solubility and target activity [52] assuming that similar chemical structures will lead to similar activities. The correlation model is built on the molecules, whose activities are known through experiments. The objective is to eliminate the molecules that have high levels of toxicity based on this model. Usually the numeric values of biological activities are unknown; however, activities of molecules can be classified into two classes based on their toxicity levels as high-activity and low-activity [53].

The QSAR approach proposed in this thesis is applied on calcium channel antogonists (dihydropyridine derivatives), which is a subgroup of a class of drugs that are used for the treatment of cardiovascular diseases by inhibiting the Ca^{2+} flux into the cells. Therefore, here, QSAR methods in the literature that are applied on the same data set are studied.

One of the earliest studies on this group of drugs was a conformational analysis applied on 45 2,6-dimethyl-3,5-dicarbomethoxy-4-X-phenyl-1,4-dihydropyridine derivatives [54]. The model relating structure and activity was built by multiple linear regression (MLR) based on the conformations of the two rings found in the molecules.

Takahata et al. [55] also conducted QSAR analysis on 1,4-dihydropyridine calcium channel blockers and compared the model building success of PCA (principle component analysis) with MLR's. The variables of the model were adopted from the QSAR study of Gaudio et al. and they obtained the most predictive variables by calculating the weights of each variable in the model derived by PCA, and these predictive variables were used to classify the data set into high and low activity classes.

Following this study, another QSAR algorithm with the same data set was applied by Viswanadhan et al. [83] called PCANN, which was also based on PCA. Predictive variables were calculated by PCA and a back-propagation neural network trained by cross validation was used for activity prediction, whose outputs were compared to the outputs gained by using MLR and a hologram QSAR model. It was exhibited that the best prediction results were achieved by the PCANN method.

Another study by Takahata et al. [53] conducted in 2003 compared NN (neural networks) with PCA by classifying the same 45 DHP derivatives into high and low activity classes first by using classical descriptors and then using the calculated ones. The outcome at the end of the study was that NN outperformed PCA in classifying the data and the proposed descriptors achieved just as good accuracies as the classical descriptors obtained.

A genetic algorithm was applied to regression analysis by Shamsipur et al. [52] on a different set of the same kind of drugs. Again, relevant descriptors were selected by building a linear model once by MLR and once by PLS (partial least squares) methods for comparison purposes. Genetic algorithm was used to prune the unsuccessful regression models among a large number. PLS was concluded to be outperforming MLR for regression purposes.

Another study conducted in 2003 by Schleifer et al. [56] compared the toxicity prediction strengths of three different QSAR methods. The prediction methods (comperative molecular field analysis), CoMSIA (comparative molecular similarity indices

analysis), and GRID/GOLPE (generation of optimal linear PLS estimations) were based on probe-ligand interaction energies. With this study, functional groups on the molecules were revealed and all methods succeeded in predicting the toxicity levels of the data set with favorable R^2 values.

The same 45 DHP derivatives were studied by Yao et al. [57], where LSSVM (least square support vector machines) method was used to obtain a seven descriptor regression model and to classify the molecules. The results were compared with reference methods from the literature and were found to be achieving the best accuracy and regression results among these.

Si et al. in 2006 conducted QSAR study on the same data set [58] in which GEP (gene expression programming) was used to build the regression model that described the $\log(1/IC_{50})$ values of the drugs and to extract the most important descriptors. Descriptors were calculated in CODESSA [59] and the heuristic method built in CODESSA was compared with GEP. A successful six-descriptor model was obtained and GEP was proved to be providing satisfactory QSAR models.

Chapter 3

MATHEMATICAL DESCRIPTION OF THE ACTIVE SITE

The success of computational drug design efforts considerably depends on the effectiveness of the algorithm that expresses the structure of the active site of interest. Coming up with an algorithm that will describe the inner volume of the active site of any given protein is a highly complex problem. The difficulty is mostly due to the fact that the problem is in three dimensions and that proteins may have any shape, which is often irregular and asymmetric. However, the major challenge is to turn the site interior, which is a virtual volume surrounded merely by "points", into a space that is enclosed by a continuous surface [60], which is vital for the design algorithm to be able to recognize the feasible region that the ligand will be placed in. It is almost impossible to represent the space *exactly*, since any computer algorithm will inevitably involve a certain degree of digitizing. Therefore, the mathematical representation of this three dimensional geometry will be an approximation.

It can be said that the most widely used surface model to represent the active site is the Delaunay triangulation model [4], which forms triangles from neighboring three atoms, whose circumcircle does not involve any other points in the set. Popular algorithms such as DOCK [61], AutoDOCK [62] and FlexX [63] are examples to the docking methods that use triangulation.

In the proposed approach, a heuristic algorithm is used that aims to cover the inner surface of the site as closely as possible for a good representation, with as few convex hulls as possible for a low computational complexity. In the literature, there are many

algorithms that simplify a nonconvex polyhedron into a set of convex polyhedra [64]. However, all these algorithms either utilize the knowledge about the adjacency of the points that build up the polyhedron, or the boundary of the polyhedron is already distinguished and defined. The algorithm offered in this study, on the other hand, has no information but the coordinates of the points, therefore differs from those algorithms.

Other than such generalized algorithms, there are many algorithms specialized in finding the active site of a given protein as studied in the Literature Review chapter. However, to our knowledge, none of them describe the active site as a set of convex hulls, a representation technique that would come in handy when building an optimization program for drug design purposes.

The motivation behind the choice of convex hulls as building blocks in order to represent a binding pocket's shape is the wish to utilize this mathematical representation as a part of an optimization model for drug design purposes. This shape defines the feasible region that the ligand atoms will be placed in by the optimization algorithm. Active sites have extremely random shapes, but it is very probable that a given pocket's shape will have a non-convex nature. To describe a non-convex shape in terms of a set of convex hulls helps decreasing the complexity of the proposed model, since it will already be nonlinear in nature due to distance and angle calculations of the atoms.

A convex hull is defined as the minimal convex set of a set of points in an N dimensional space. In computational geometry, however, it usually refers to the boundary of the minimal convex set of a set of points in three dimensional space [65-66]. Mathematically, a convex hull in 3D is represented by a series of linear inequalities that defines an intersection of half spaces. The left hand side of each inequality describes a 2D façade of the hull, while the direction of the inequality defines the feasible half space.

$$Ax + By + Cz + D \leq 0 \qquad (3.1)$$

A series of such inequalities describes a three dimensional convex structure.

3.1 Assumptions

In order to simplify this problem, a number of assumptions are made. First, atoms are assumed to be dimensionless and are represented by points, each of whose coordinate in the 3D Euclidean space is indicated by the center of mass of the atom that it represents. (It should be noted here, however, that van der Waals distance of every atom should be taken into account later on while placing the candidate ligand's atoms by the optimization model in the form of constraints prohibiting trespass upon these borders.) The coordinates of the protein atoms are obtained from their crystal structures stored in PDB. Second, proteins are assumed to be rigid bodies, whose atom coordinates does not shift. This means that it is assumed that the proteins stay in the conformation. Another assumption about the nature of a common active site's geometry essentially marks what the algorithm is built on. The assumption is that the "interspace" all along the active site is always larger than the space between neighboring atoms. The use of this assumption is explained below.

3.2 Representation of the Active Site

The algorithm proposed in this thesis is based on a sweeping method that slides a thin slice that is formed by two parallel planes and the region between them. The motivation behind using such a method can be understood by explaining some methods, which were studied as candidate algorithms in this thesis but were found to be inefficient and ineffective.

3.2.1 First Attempts

Methods that first come to mind that involve building triangulations from neighboring points or that are based on dividing the space into grids and checking for points in those grids have already been applied in the literature; however, they are not based on building convex hulls. Rather, they either merely build surfaces but not describe the volume of the pocket or define the volume by a collection of grids, as explained in detail in the Literature Review section. Therefore, to represent the active site in terms of convex hulls, a new algorithm had to be developed.

First, a sphere enlargement algorithm was attempted. According to this algorithm, in each iteration, the active site was supposed to be inspected by a series of spheres that share the same central point, which is in the "middle" of the pocket's most buried surface. At the beginning, a very small sphere was to be built and then this sphere would be enlarged until it hits an atom. Then it would search for the second atom and so on until there were four atoms that could build a convex hull. The algorithm would go on in the same respect, but after the fourth atom, a new atom would be considered within the convex hull only if the convex hull that was built by the already accepted atoms and the new one contained no atoms inside. Then once there remained no atoms to be included, the center would be placed on the "top" of the newly built convex hull and the next iteration would begin with a small sphere without considering the atoms below the center atom. The reason for not considering those atoms is the following: if they were considered, the next convex hull could never be built since it would always contain the points already added to the previous convex hull. The top of the convex hull was defined to be the surface having the largest area, based on the assumption that the space in the pocket is larger than the spaces between neighboring atoms.

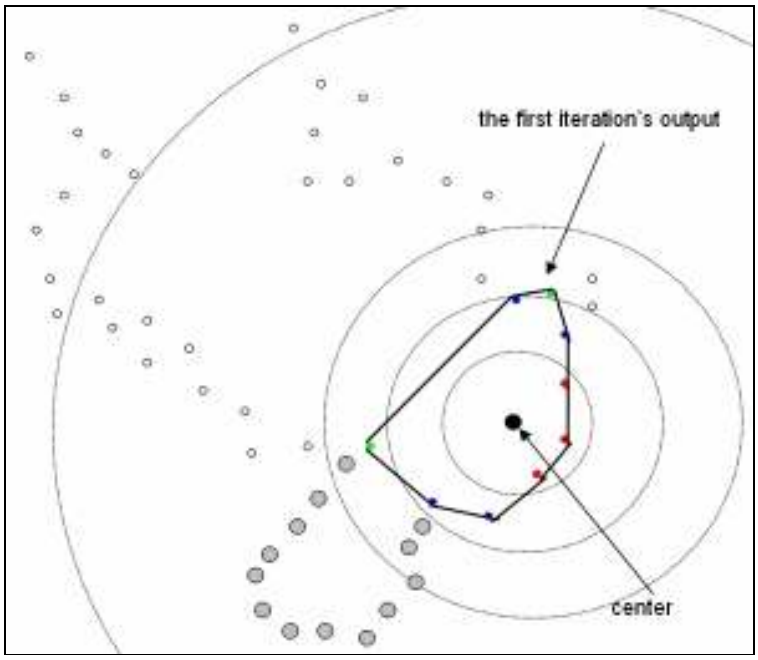


Figure 3.1: The sphere algorithm

This algorithm looked promising for some smooth shaped pockets, but as it was told before, proteins tend to have non-smooth shapes and so do their active sites. First of all, the choice of the location of the center affected the representation tremendously, and there was no optimal location to place this center, since the best place differed for every other protein in terms of the output obtained. Secondly, the algorithm missed the spaces that branched sideways as shown in 2D in Figure 3.1 by large grey points. Here, the largest convex hull in 2D that could be built by the first iteration can be seen. The color differences of the points that build the shape indicate that they are included in the convex hull in different sphere instances. The next iteration will be started from the upper part of the convex hull facing the inner space of the active site by placing the next center in the middle of the upper surface of the convex hull. Since the points below this surface cannot be added to the next convex hull, the space surrounded by the larger grey points is simply

missed by the algorithm, and much work has to be done for the algorithm to detect the missed spaces and to conduct iterations in these missed spaces. However, it would amplify the complexity tremendously and also its feasibility was questionable, since pocket shapes are highly random.

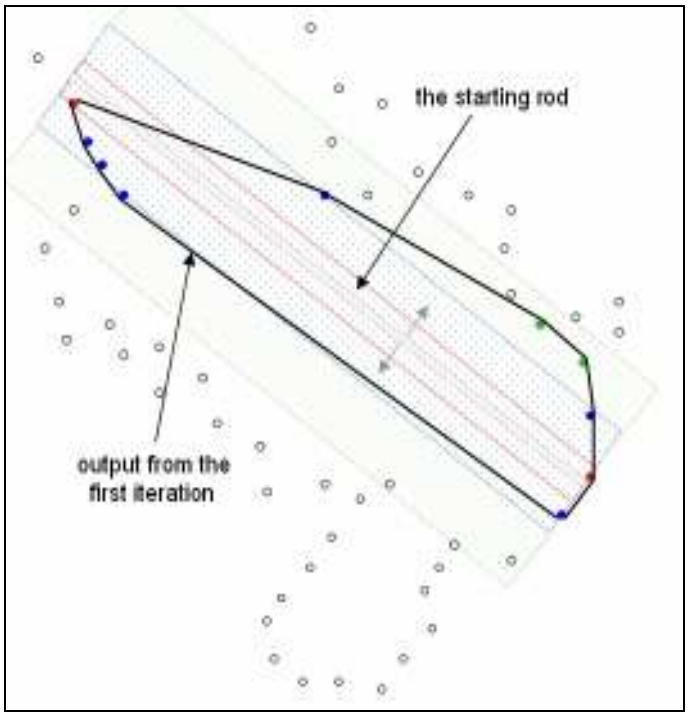


Figure 3.2: The rod algorithm

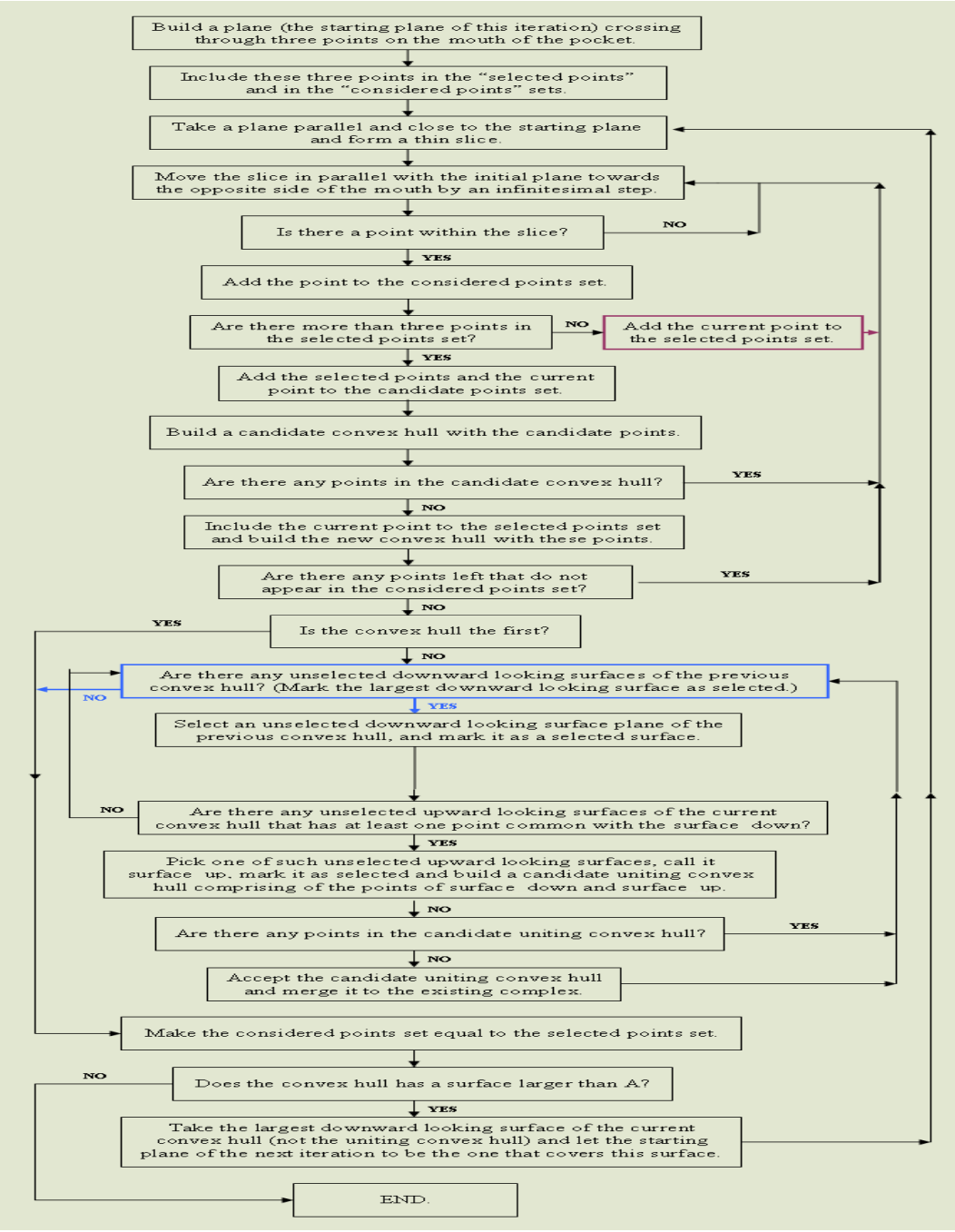
Other algorithms based on other three dimensional shapes led to problems very similar to the sphere algorithm, and will not be explained here. However in the algorithm based on rod enlargement, the output was different. In that algorithm, a rod was placed "all along" the pocket as high as possible, and was enlarged sideways to include points in convex hulls. One rod could cover one turn of the pocket if it contained turns. After an iteration was complete, the next rod was placed above the largest surface of the most recent convex hull

and another iteration was to be conducted (see Figure 3.2). This algorithm looks more efficient than the sphere algorithm, because it requires fewer iterations to cover the active site, but it still has the problem of missing spaces, and it needs to be branched from every surface just like the sphere algorithm. Moreover, the height of the rod has to be modified in each iteration, and the feasibility of an algorithm that would come up with these optimum heights is, again, questionable.

These algorithms do not provide effective and efficient methods for pocket coverage, since in each, every enlargement was constrained by the shape of the three dimensional form. The rod algorithm, on the other hand, reaches better solutions since it already covers the pocket site along with its height, and is enlarged only sideways; however, as explained above, this height needs to be optimized in each iteration. The location of the initial rod placed in each iteration is also an important decision to be taken, since it determines the fate of the final space covered, and this does not have an optimum answer for all pockets either. Thus, another idea is applied, which is not based on 3D forms and would not require decisions throughout the iterations that would lead to random consequences. Such an algorithm is presented here as the sweeping slice algorithm.

3.2.2 The SLICE Algorithm

The proposed algorithm SLICE that covers a given active site by convex hulls is a greedy heuristic method that is based on sweeping a slice. The only inputs of the algorithm is the information that indicates which atoms are the three outmost atoms that build up the “mouth” of the pocket, that is the opening face of the pocket to the exterior, and the coordinates of the atoms that are present in the protein. The steps of the algorithm can be seen in Flowchart 3.1.



Flowchart 3.1: The SLICE algorithm

Each main iteration of the program includes many iterations, where in each the “slice” is slid a little bit inwards of the pocket. On the other hand, each main iteration corresponds to a new convex hull: the new main iteration is started and a new convex hull is sought when the new points met by sliding the slice do not contribute to the current convex hull anymore. The details are presented below.

There are three point sets in the program. One is the “to-be-considered points” set, which, in a certain main iteration, contains the points that are not part of the representation. These points are yet to be considered and decided whether they should be in the representation or not in the following iterations. The other set is the “selected points” set, and the third is the “considered points” set. The prior set includes the points selected to be building the convex hulls that are part of the representation, whereas the latter includes the points that are met by the program in each iteration of a main iteration. Whenever the program meets a point in an iteration, it includes the point into the “considered points” set and then decides whether the point should be included in the current convex hull or not, i.e. whether it should be included in the “selected points” set. The “considered points” set adds all of the members of the “to-be-considered” set one by one to itself in each iteration of a main iteration, and is updated to be empty at the beginning of every main iteration. The selected points can never be excluded from the “selected points” set, only a new member can be added, whereas the size of the “to-be-considered points” set keeps diminishing in each main set, since the selected points are excluded from this set except for the three points that are part of the starting surface of the next convex hull. The details of these selections will be explained later.

The main idea of the algorithm is to build adjacent convex hulls that cover the inner space of the pocket as closely as possible. Given the inputs, the algorithm takes the three atoms that are present on the mouth of the pocket and builds the 2D simplex with these points as the starting surface of the initial convex hull. These first three points are taken

into both the “selected points” and the “considered points” set. Then, the algorithm takes a thin slice into account, which is formed by the starting surface and a surface parallel to it. This parallel surface is a very small distance away from the starting surface and is placed closer to the “bottom” of the pocket.

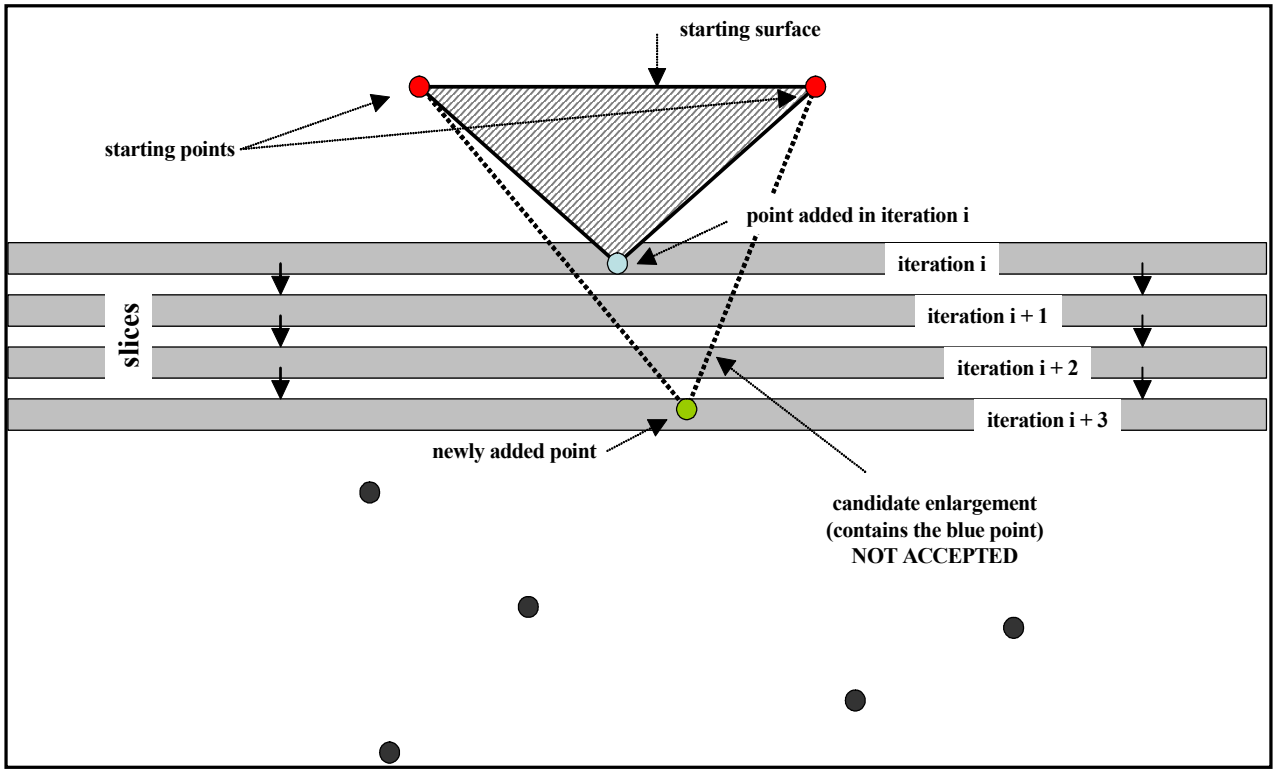


Figure 3.3: Convex hull enlargement attempt in the SLICE algorithm

Once the two parallel surfaces are built, it is checked if there is an atom that falls into this slice. It should be noted that the slice is taken thin enough so that only one atom at a time can be present in it. If there is an atom in the slice, a convex hull is built with the three starting atoms and the current atom found in the slice. If not, the slice is slid lower until there is an atom within. The algorithm goes on in a similar fashion, such that the next

atom is sought by iterating the slice lower and lower, and once found, the convex hull is enlarged by adding the atom into the selected atoms set. However, in every time corresponding to a possible enlargement of the convex hull, the algorithm checks whether an atom is present inside the newly formed hull. This step is crucial since the algorithm is to present *hollow* convex hulls to express the inner-space. The case in which there is an atom in the newly enlarged convex hull is illustrated in Figure 3.3.

If this is the case, the new convex hull is not accepted, the point is not included in the "selected points" set and the algorithm goes on with the search of the next atom. At some point, if the whole active site is not convex by nature, which is an extremely rare occasion, the algorithm will not accept any other point to enlarge the current convex hull. At this point, the algorithm builds the convex hull with the selected atoms and goes on to build the next convex hull, which is to be placed adjacent to the previous one. Each façade of a convex hull is a 2D simplex build up of three atoms. Therefore, in order to build an adjacent convex hull, the next set of starting points should be composed of the three points of one of the façades of the previous convex hull. Choice of these starting points has major importance for the algorithm, since according to this choice; the heuristic will determine the direction of the object to grow. The reason why this choice is significant is illustrated in Figure 3.4 and Figure 3.5.

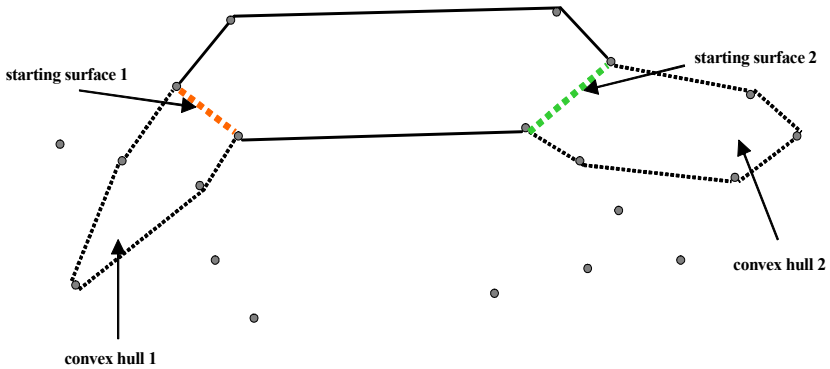


Figure 3.4: Two cases of wrong choice

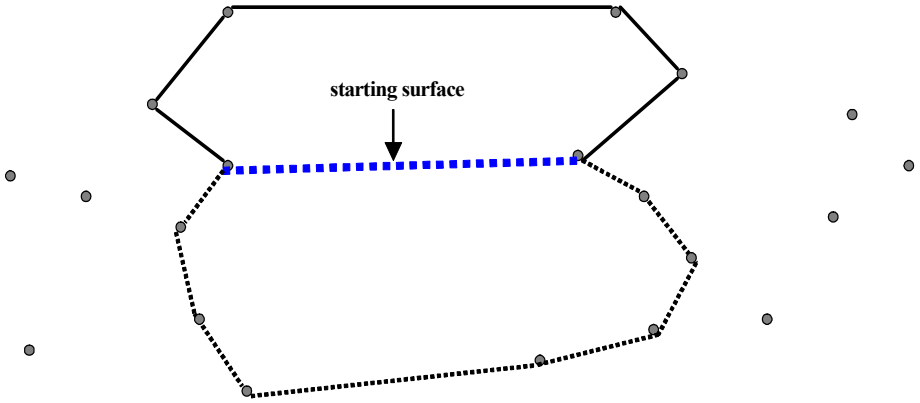


Figure 3.5: The right choice

Here, there is an important issue to be noted. Since active sites have highly random shapes, the pocket may exhibit even "u-turns", and in such a case, "downward" becomes "upward". Therefore, the algorithm should have an updatable direction rule. This is handled through updating a point called the "ceiling" every iteration j to be the middle point of the j-nth starting surface that is the starting surface of the convex hull, which is n hulls before the current one. This "n" is a number that can be changed by the user according to the shape of the active site. If the algorithm needs to keep track of every move because there are many turns and cavities, this "n" should be small. However, keeping "n" very small in every case may not work. A convex hull may attempt to discover a cavity, but may need to go on towards the original direction afterwards. For smooth shapes, a small "n" is not recommended. In this study, n was taken to be 3 due to inspection.

In the design phase of this study, only the atoms building up the active site were taken into account to lower the time needed for the iterations. For this purpose, we utilized methods that provide the active site of a protein called ConSurf [67] and Q-Site Finder [8].

These programs present the active site as a set of amino acids that construct it and different applications offer different sets of amino acids for the same pocket site, which required taking the union of these into account. Therefore, the input, which is comprised of the atoms building up the active site, did not include only the atoms present on the surrounding contour of the pocket site, but it included many of the neighboring atoms as well. One can venture the cumbersome effort of distinguishing only the contour atoms among hundreds by hand; however, this algorithm does not require such an input. Our purpose was to develop an automated algorithm that is able to detect the active site given only the coordinates of the protein atoms and the three atoms on the mouth of the active site. Thus, for our purposes, the challenge illustrated by Figures 3.4 and 3.5 is inevitable.

To deal with this problem, a greedy approach is used. It is assumed that the direction of the enlargement should be through the *largest* space that is "*below*" the previous convex hull, since the other neighboring spaces, which are indicated by convex hull 1 and convex hull 2 in Figure 3.4, are formed by atoms that are in interaction, thus the interspace between these are usually smaller than the interspace formed by the active site.

However, this assumption may not hold in some cases, although such cases are estimated to be rare. First, sizes of the spaces within interacting atoms change depending on types of interactions, thus on the atom types. Also, active sites have tremendously varying shapes. Moreover, only a ratio of the 3D shapes of proteins is known today. Therefore, the smallest "diameter" of the pocket space of the protein may be of the same size as or even smaller than the "diameter" of the space lying between the surrounding atoms. Such a case may occur if surrounding atoms have large van der Waals radii, building weak bonds and not sitting very close to each other; while the atoms in the referred part of the active site have small van der Waals radii building close interactions.

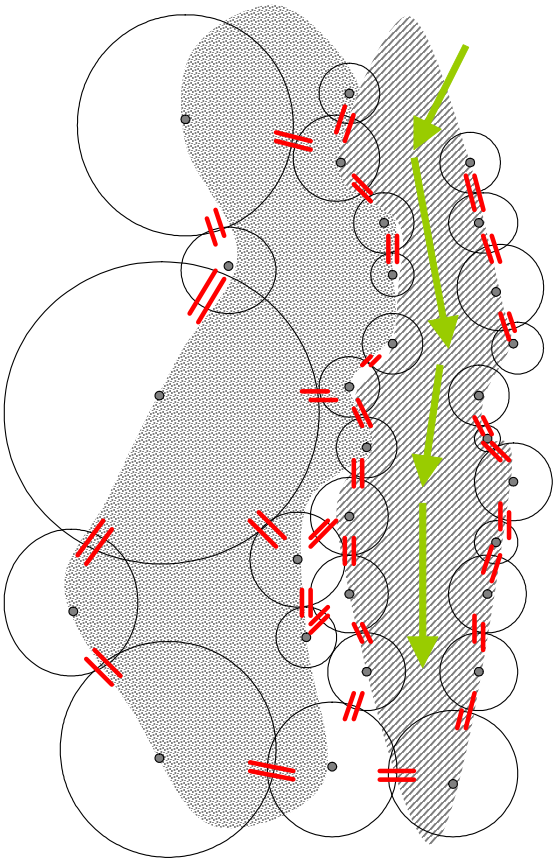


Figure 3.6: The case of having a smaller inter-space in the pocket than the inter-space between surface atoms.

In such a situation, the heuristic will pick the wrong direction and the algorithm will fail to describe the active site accurately. This is due to the fact that the program considers atoms as points, and does not take their van der Waals radii into consideration. To solve this problem, knowledge about the atom types and the binding relationships can be fed to the program and the algorithm can be extended so that it calculates the vdW radii by exploiting this information and thus more precisely determines the space that the ligand can

go into. Nevertheless, it is obvious that such an extension would increase the complexity of the algorithm and also it can be said that the probability of such an occurrence is considerably low looking at the population of known active sites.

According to the stated assumption, the starting three points of the next convex hull are decided to be the points that sit on the corners of the largest simplex that faces the space "below" the previous convex hull. Here, two remarks should be made. First, the reason of stressing the need of the simplex facing below is because the largest simplex of the previous convex hull may be facing upwards, which will take the algorithm to the wrong direction while the correct direction is "downwards", which is towards the interior of the protein. Here, it should be stated that the direction towards the interior of the pocket is taken by the algorithm as the opposite direction of the one pointing towards the middle of the previous convex hull. Second, the reason to put "below" in quotation marks is not only because where "below" indicates depends on the reference points we take, but also the active site may exhibit turns and loops, thus changing the direction to be followed continuously.

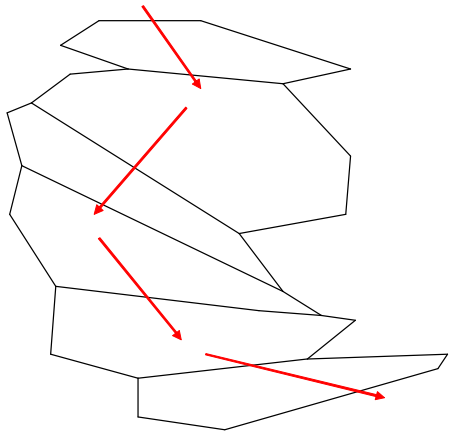


Figure 3.7: Directionality in the SLICE algorithm

The next convex hull is started if there are no points left that are not included in the "considered points" set. It is started from this selected surface and is formed by applying the same procedure used in building the first one. In each main iteration that a new convex hull is to be built, the points of the previous convex hull are removed from the set of "to-be-considered points" except for the three points comprising the current starting surface. Initially, the starting surface is automatically included as one of the facets of the next convex hull, and then the sweeping algorithm takes over.

After building the first two convex hulls, it is observed that there is some space left out that need to be included in the complex (see Figure 3.9). This is true for every consecutive couple of convex hulls (i^{th} and $i+1^{st}$ convex hulls) to be built. These spaces are between the simplexes of the $i+1^{st}$ convex hull that look "upwards" and the simplexes of the i^{th} convex hull that look "downwards" having common points. Convex hulls are built by the atoms of these coupled simplexes if there are no atoms in the resulting convex hulls. This is accomplished by considering each one of the downward looking surfaces of the i^{th} convex hull and each upward looking surface of the $i+1^{st}$ convex hull that shares one or more common points with one of the former set. Each surface couple facing each other and sharing common points is considered to see if they need to be added to the complex and is united if the convex hull comprising of the points that construct them has no point inside it. This newly built uniting convex hull is merged to the existing complex.

The algorithm goes on by building convex hulls and uniting the spaces between the i^{th} and the $i+1^{st}$ convex hulls until there are no points left to be considered.

The active site is then described as a set of convex hulls, thus as a set of linear inequalities representing the border and interior of the convex hulls. The number of convex hulls depends on the shape of the active site. The more turns there are, the more convex hulls are likely to be. Also, this number increases with increasing number of straits in the active site. The number of facets of each convex hull may also differ, but the minimum

number of facets that a convex hull may have by definition is four. Therefore, each convex hull will have equal to or more than four simplexes and there will be as many linear inequalities as there are simplexes.

The algorithm is coded in MATLAB [68], a high level programming language. In the development phase of the algorithm, visual inspection was used to measure the effectiveness of the algorithm. MATLAB has the feature to draw graphs in three dimensions, thus the input points and the resulting convex hulls can be inspected together, and it can be seen whether the active site is covered closely enough. The transparency level of the 3D shapes can be tuned by the user, allowing examining to control if any points were included in the output.

3.2.3 Case Studies and Comments

This algorithm does not miss any spaces that *are* determined by points; however, it has its own shortcomings. The algorithm works perfectly fine for active sites that are completely buried within the protein, that is a pocket surrounded by atoms, such as the case study presented in the Appendix for the protein "v-kit" (homo sapiens v-kit Hardy-Zuckerman 4 feline sarcoma viral oncogene homolog) that goes by the PDB name 1T46 [69]. However, the algorithm fails to cover the space closely when the pocket is not surrounded by atoms. Since each move in the algorithm depends on the existence of points, the method struggles for such active sites and builds thinner and thinner convex hulls since it can find points that are far away from each other. Such a case is represented with a dihydrofolate reductase, (PDB id: 1VJ3) [70]. The two other case studies are for 3BGY [95] and 1IKT [96].

Moreover, the uniting algorithm that is embedded into the main method is open to further development. In certain instances, the surfaces that are united are very small, yet

there are larger ones beside them left disunited since uniting them would cause inclusion of points within the structure. In fact, these points that are included in the structure by uniting algorithms can be ignored, since the main body of the structure accomplishes to draw the "inner" space that does not contain any points. The points that will be included can be negligible in some cases, but it depends on the case as it is always with the proteins. Therefore, the user can decide if he/she should ignore these points or not.

The Case Study with 1T46

The protein "V-kit" is a part of the provirus of Hardy-Zuckerman 4 feline sarcoma virus and is claimed to be the cause of gastrointestinal stromal tumor. This protein causes mutation on the gene "c-kit" and thus causes encoding of a transmembrane receptor for a growth factor named "stem cell factor" which leads to the development of gastrointestinal cancer [71].

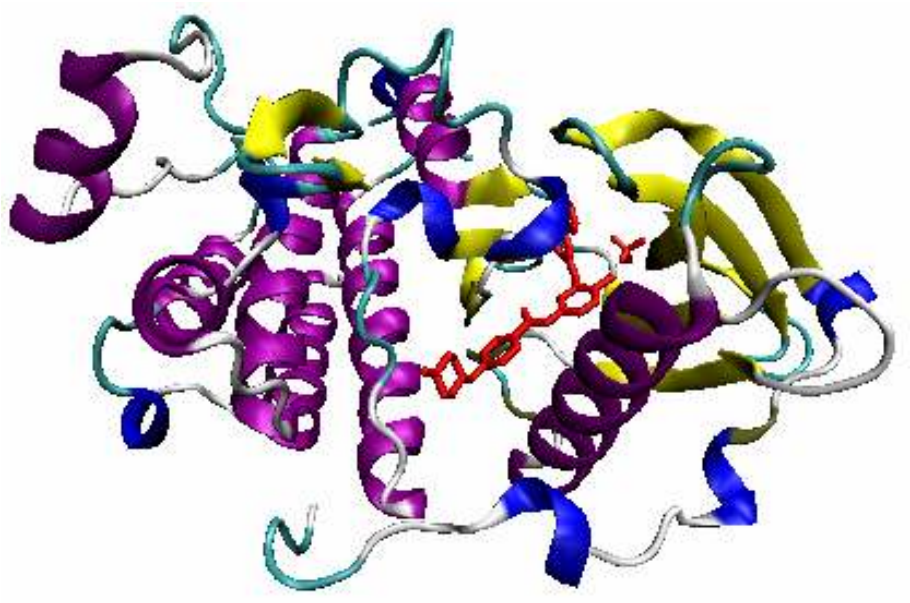


Figure 3.8: 1T46 and Gleevec docked into its active site

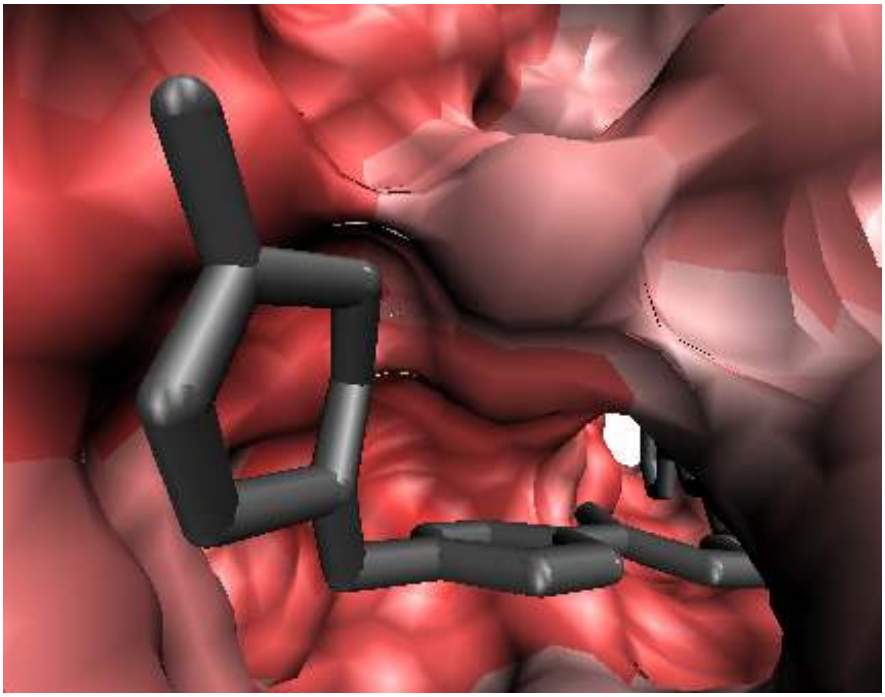


Figure 3.9: The tunnel active site of 1T46 zoomed

As it can be observed clearly from Figure 3.9, the active site of 1T46 is a tunnel, which means that the site is totally covered with atoms (points). This allows the algorithm to detect the site more closely.

The output active site representation of this protein can be seen in Figure 3.10. In this figure, the very thin red structure whose a little part is seen in the marked space is the structure formed by the coordinates of the atoms of Gleevec, a ligand that was designed for this active site to suppress binding of this protein to the "c-kit" gene. The active site representation is supposed to be covering the ligand structure while not including any of the atoms inside. In the figure, we see a certain area seems to be uncovered, where there are few atoms that describe the shape.

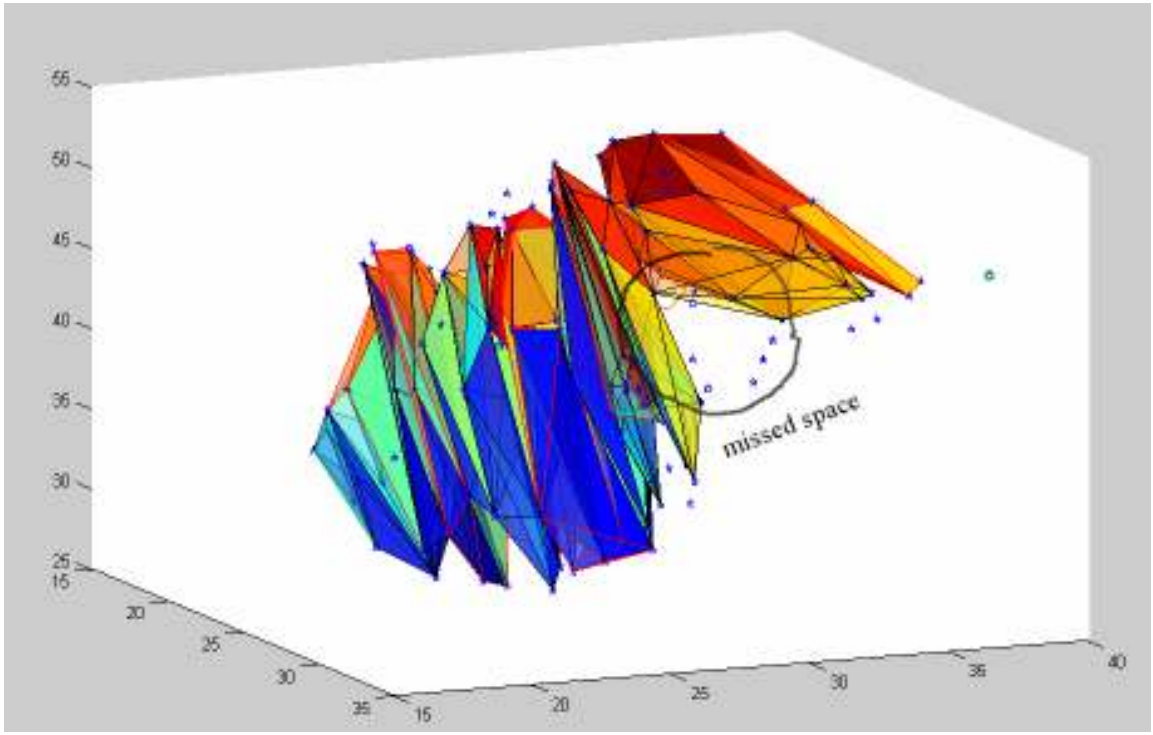


Figure 3.10

This problem may be addressed by the to-be-designed mathematical problem with a relaxation of the constraint that all points should be in one of the convex hulls. A penalty function can be added to the objective function instead. This issue will be addressed in the following section, where a draft optimization model is presented. Another solution is to ignore the points included by the "unite" algorithm. Below are the figures displaying that case.

Figure 3.11 is a screenshot from the 4th iteration. As it can be seen from the figure, the missed space is smaller than the case before and the part of Gleevec that could be seen in the previous figure is completely covered by the convex hulls.

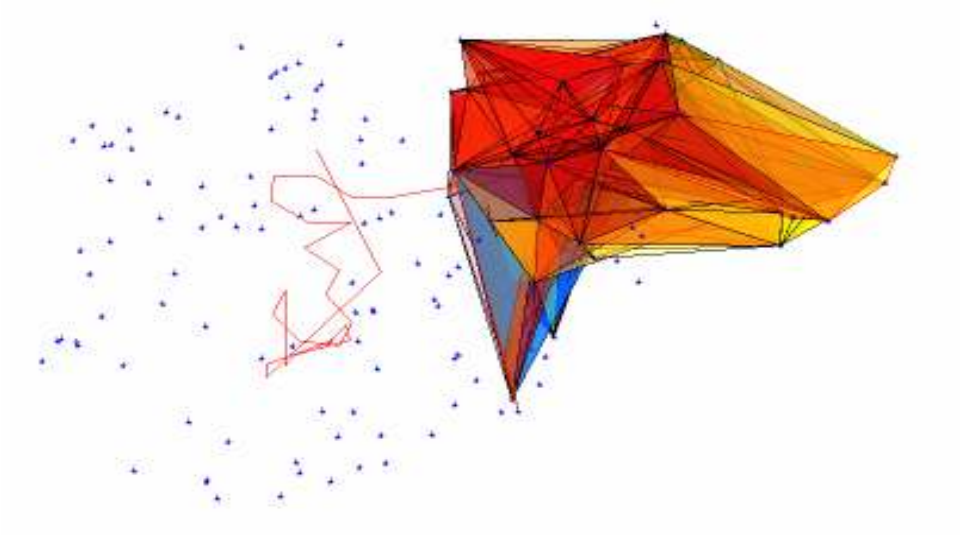


Figure 3.11

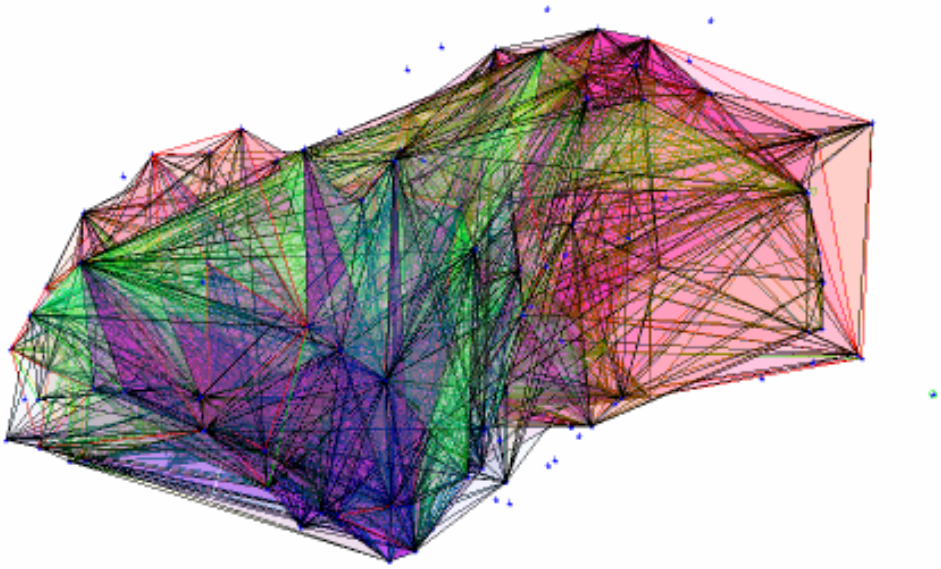
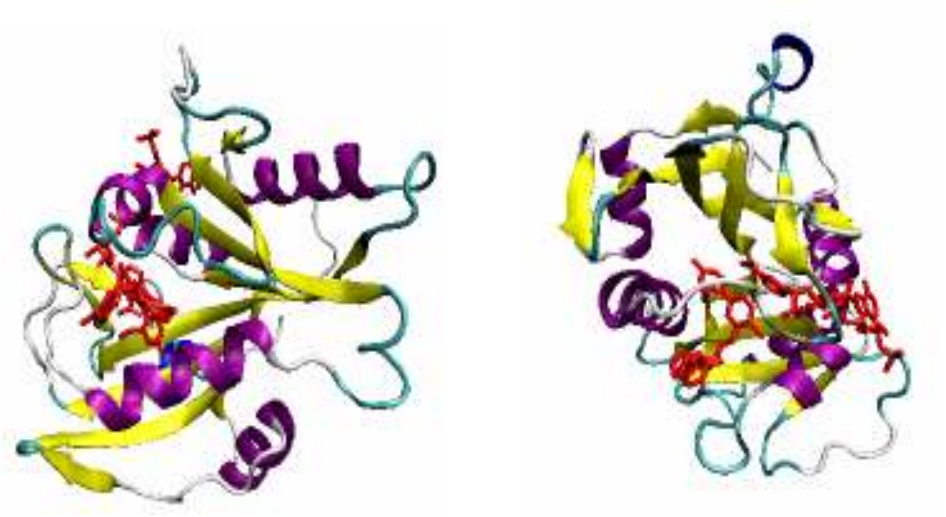


Figure 3.12

In the figure above, 1T46 is covered completely ignoring the points included by the "unite" algorithm. It can be observed that there are many points within the structure close to the walls of the convex hulls; however, this problem can be handled by the optimization model with the help of a constraint telling the model that the van der Waals sphere around of surface atoms should not be violated. Since points that build up the protein are at most that far from each other, this may solve the problem. Also, as told before, the structure succeeds in capturing the inner space, which is the main goal of the study.

The Case Study with 1VJ3

Another protein called pneumacystis carinii dihydrofolate reductase cofactor (PDB id: 1VJ3) [94] was taken into consideration and its active site was represented by the SLICE algorithm. The protein and the ligands of 1VJ3 (Tab and Ndp) are presented from different points of view in Figures 3.13-a and 3.13-b



Figures 3.13-a and3.13-b: 1VJ3 with Tab and Ndp docked into the active site.

As it can be seen from the figure below, the active site has wide openings to the outer space. This brings the result that the site is partially covered with points, which creates a challenge for the SLICE algorithm.

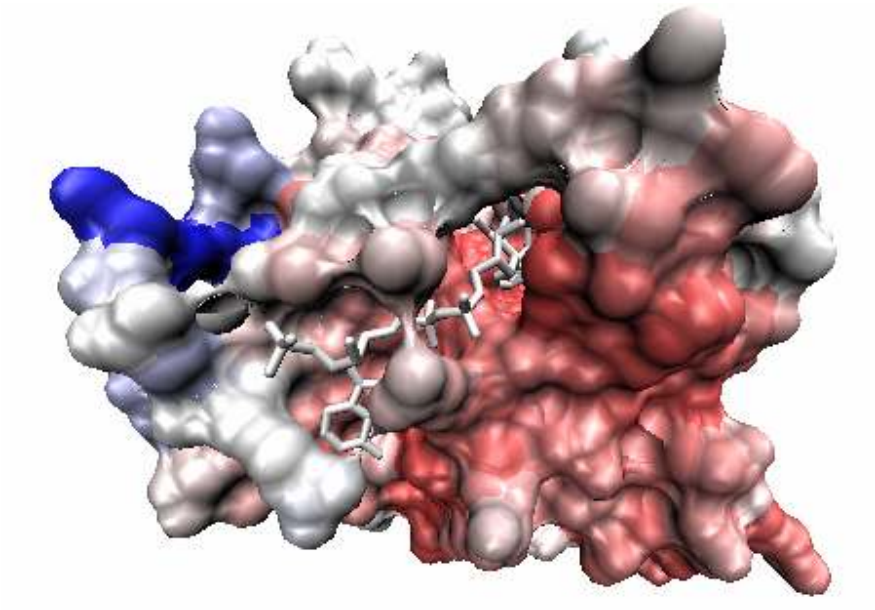


Figure 3.14: Active site of 1VJ3

Since a lot of space was missed with no points included in the "unite" algorithm, the following results are obtained by ignoring the existence of those points in the structure. Figures 3.15-a and 3.15-b are screenshots from the 9^{th} iteration with 1VJ3. It is observed that some parts of the ligands are still missed and not covered by the algorithm since there are no points of the protein in these areas.

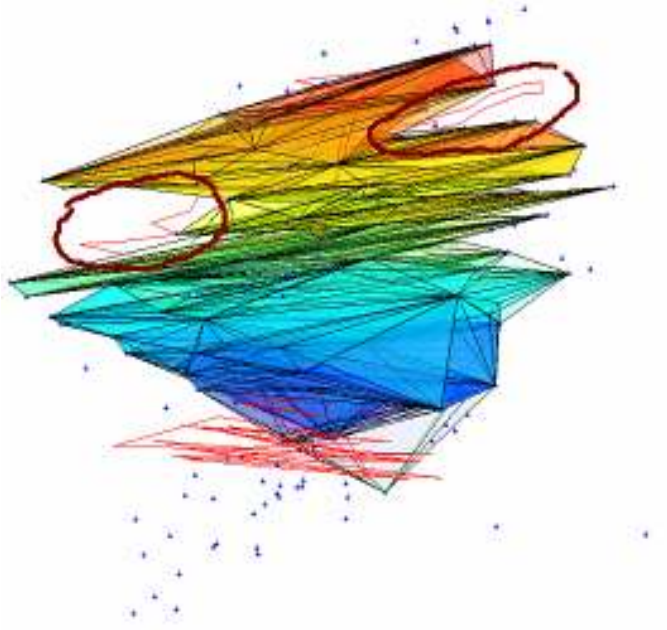


Figure 3.15-a: 9th iteration. Missed spaces observed from side view.

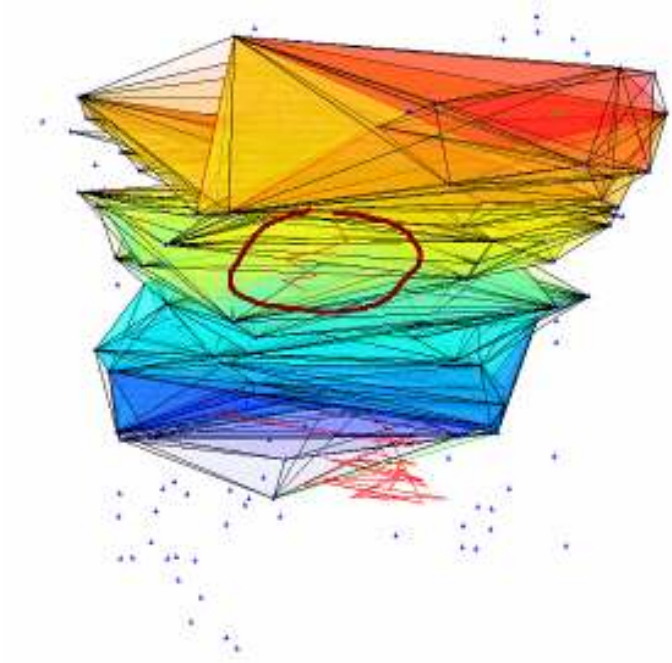


Figure 3.15-b: 9th iteration. Missed space observed from top view.

Apart from the missed spaces observed in iteration 9, there is one more missed part at the end of the run. Also, there are points included in the output structure since the "unite" algorithm did not take the points included during the uniting process. However, these are again tolerable, since the points are close to the walls of the convex hulls and can be handled by the optimization model.

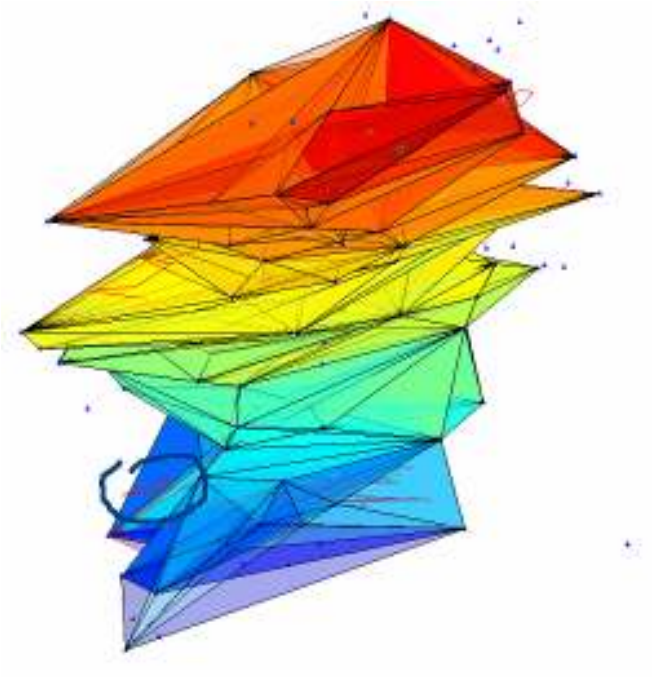


Figure 3.16: The end result of the run

The Case Study with 3BGY

The SLICE algorithm was also tried on a beta barrel structure found on an mRNA capping viral protein: 3BGY [95] with acetate ions attached. The structure of the protein composed of two chains can be seen below. The two chains have exact beta barrel structures, and the algorithm is run only with the B chain, which has two acetate ions as ligands bound at the middle of its barrel to see the active site clearly in the figures.

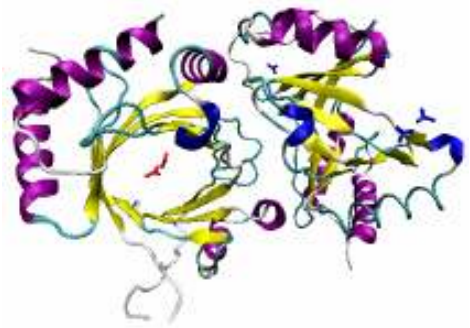


Figure 3.17: 3BGY. The beta barrel studied belongs to the chain seen on the left.

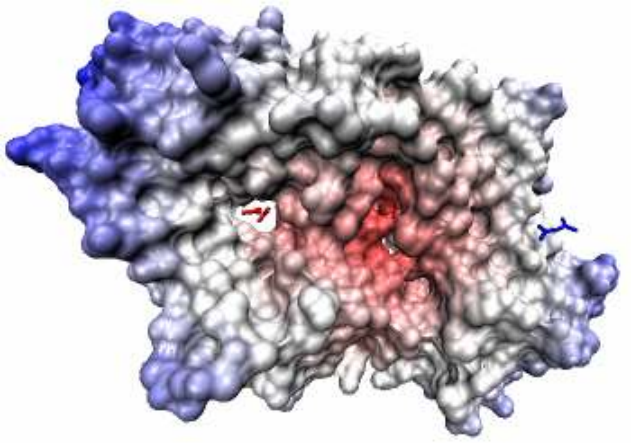


Figure 3.18: Solvent accessible surface of 3BGY. The tunnel is seen on the left.

Figure 3.18 clearly exhibits the tunnel that the barrel forms. The algorithm is expected to provide good results, since the active site is completely surrounded with points. The output's success was measured in terms of capturing the space comprised by the barrel this time. The ligand's coverage was not important in this case, since its size was very small.

The active site was represented once to obtain a structure with no points inside, i.e. the "unite" algorithm did not ignore the points included while uniting the convex hull. The output is seen in Figure 3.19 with a missed space that is marked.

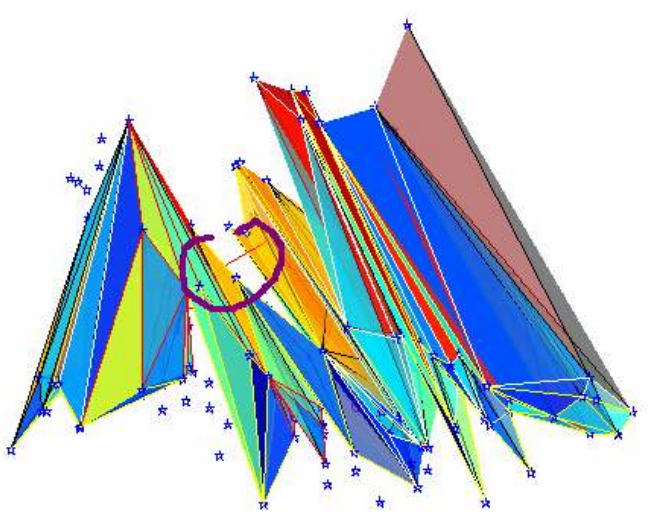


Figure 3.19: Representation without points inside

The output of the run ignoring the points included within the "unite" algorithm is presented below. The tunnel was successfully covered by the algorithm. (See Figures 3.20-a and b.)

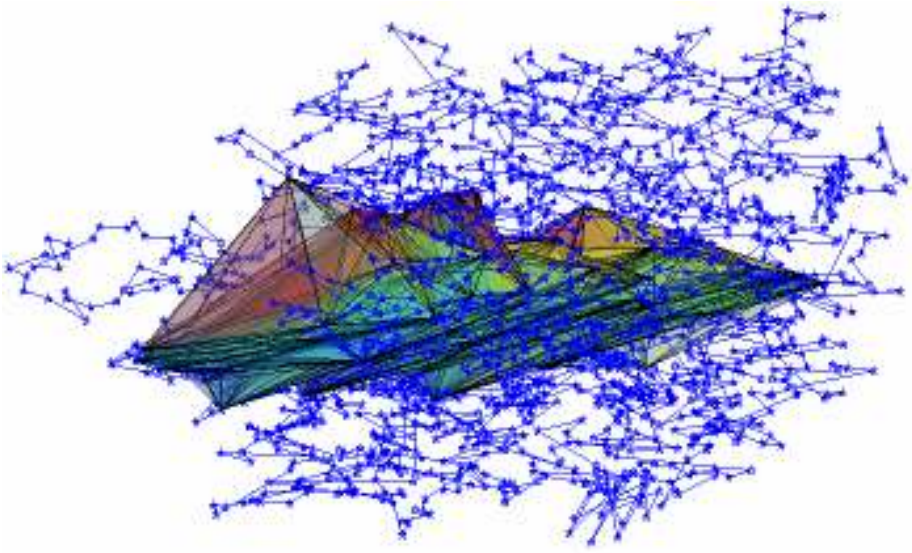


Figure 3.20-a: The representation of the tunnel in the B chain of 3BGY.

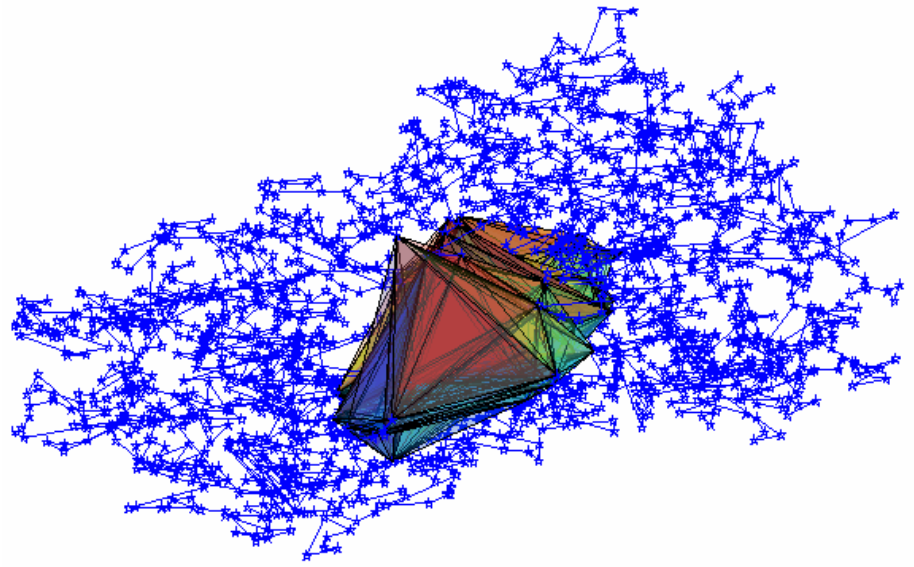


Figure 3.20-b: The representation of the tunnel in the B chain of 3BGY.

The Case Study for 1IKT

The last case was the sterol carrier protein type 1(SCP-2) like domain of human multifunctional enzyme type 2 (MFE-2) with a PDB id of 1IKT. The active site's shape is like a tube with half of it cut open. The structure can be seen below.

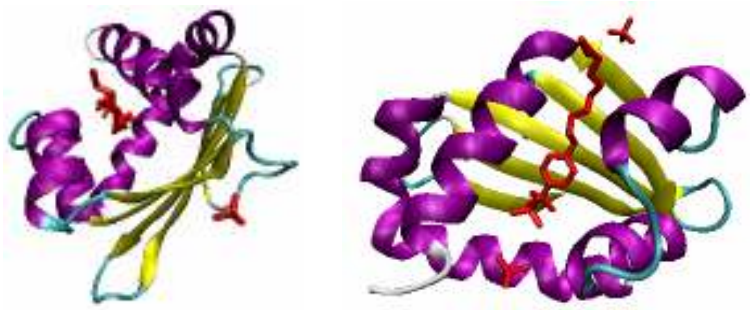
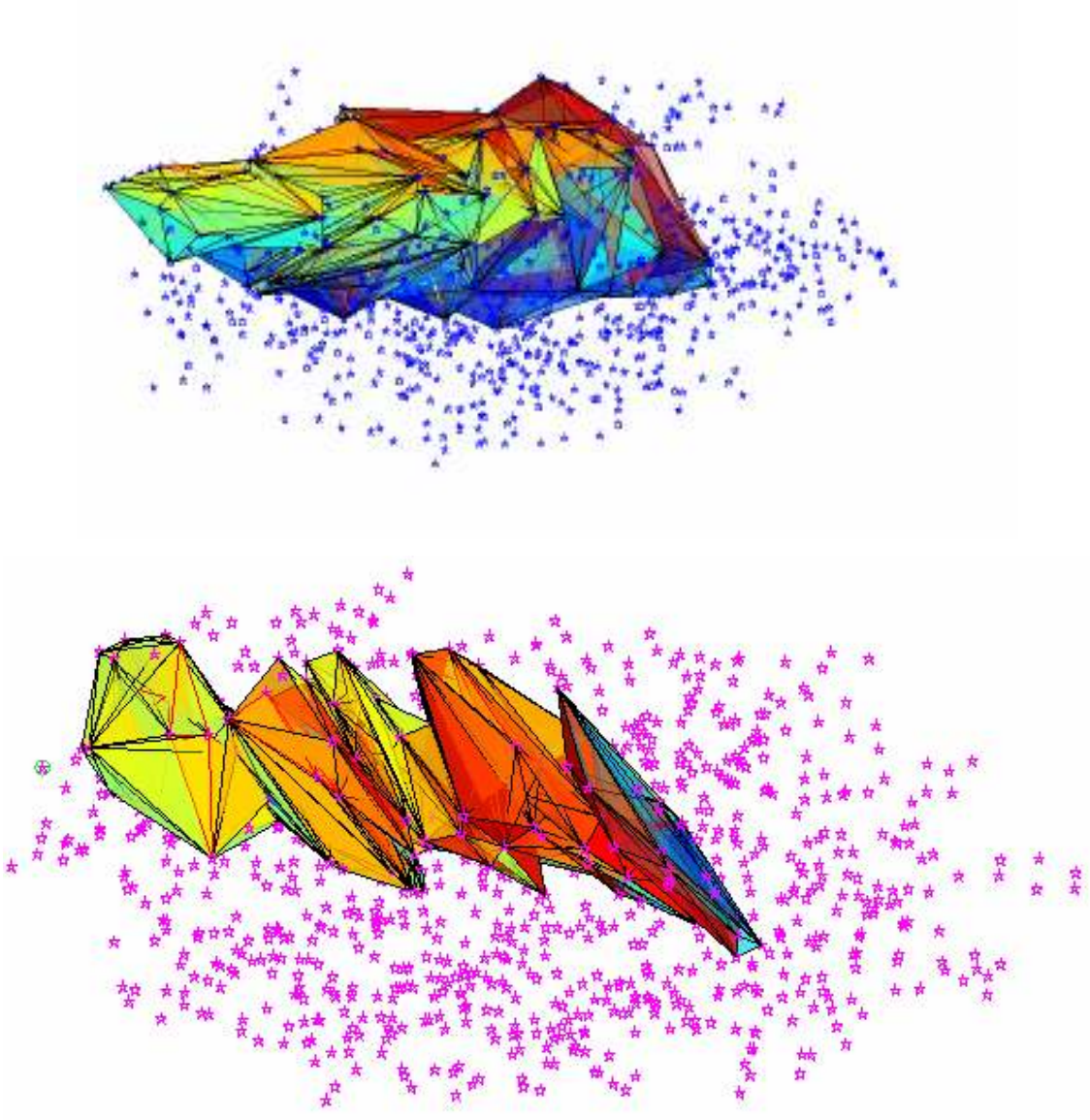


Figure 3.21-a and 3.21-b: 1IKT with its ligand from different points of view

The active site was represented ignoring the points included with the “unite” algorithm. The result can be seen in Figures 3.21-a and b. The algorithm covered the active site completely.



Figures 3.22-a and 3.22-b: The representation of the active site of 1IKT

Chapter 4

CONCEPTUAL MODEL FOR FRAGMENT ASSEMBLY

In this section, the basis of a conceptual MINLP model idea for fragment-based drug design is presented. The motivation behind presenting this draft model is merely to provide a starting point for future studies. Also, a fragment library developed for this purpose will be presented.

4.1 Assumptions

As explained in section 2.2.1, one of the major goals of structure based drug design is to design ligands that have maximized binding affinity with its target. In this draft, maximum affinity is sought by maximizing the number of non-covalent bonds between the target and the ligand. The aim is to minimize the enthalpy change of the binding process, which is claimed to be a more important task than maximizing the entropy by Freire [37]. Also, as it was with the active site representation, atoms are considered to be dimensionless; however, their van der Waals regions are aimed to be presented as inaccessible by the related constraints. Both the active site and the fragments are accepted as inflexible; however, although their values are yet to be revealed, intervals are allowed for the distances, bond angles and torsional angles of the fragments, which aim to provide a certain degree of flexibility to the design of the ligand molecule.

4.2 The Proposed Idea and the Conceptual Model

The claim that optimization can be used to design ligands by optimally selecting and binding fragments together is a bold one, and needs a lot of research before coming up with the model itself. One of the main problems about this idea is incorporating flexibility of the molecules into the model. Proteins are in fact not rigid molecules, and they have different modes of movements. Nevertheless, this challenge is not special to optimization. There are flexible screening and docking algorithms such as FlexX [63] and DREAM++ [73]; however, no method claims to be incorporating this effect exactly in their algorithms. Others incorporate many possible conformations of fragments in order to add flexibility to some degree. Another challenge is to come up with the objective function, which is supposed to maximize affinity, but the definition of affinity itself can be a vague concept. What is usually done in structure based drug design is to come up with a model that will represent the energy of the whole structure. As was studied in the Literature Review chapter, most de novo design methods merely use energy fields, which are essentially approximations, and some others use rule based algorithms.

The proposed preliminary optimization modeling idea aims to design a ligand that will fit in the mathematically described active site and will have favorable interactions with the surface atoms. The model is supposed to select fragments among a set to form a ligand molecule and assigns their binding configurations. The resulting molecule is supposed to make non-covalent bonds with the active site side chains that contain hydrogen acceptors and donors, negatively or positively charged atoms. In the absence of hydrogen bond acceptors/donors, it builds van der Waals interactions with the surface atoms.

The feasible ligand for the drug to be placed in is represented by the union of convex hulls incorporated by the SLICE algorithm. This region is represented by a series of linear inequalities, which are convex in nature. An optimization model that is supposed to set the

locations and types of the fragments to be bound together and placed in the active site forming a candidate ligand for the protein in consideration needs to decide in which convex hull a particular fragment atom will be placed. Since this space is defined as the union of a set of convex hulls, which is defined by "or" operators, disjunctive logical relations [74] need to be used to make this decision to set. Because of this and due to many other decisions to be taken, the model is an IP in nature.

Moreover, the model should calculate distances between the atoms, bond angles and torsional angles of the whole structure, since these values will be constrained by physical laws. These calculations involve nonlinear operations, which leads the model to be nonlinear. Therefore, it can be said that the model of the proposed idea is an MINLP model.

4.2.1 The Objective Function

The objective function of the model may contain the following elements:

$$\max z = \sum_{l}\sum_{j}\sum_{i}(V^{(i)} * t^{(i)}_{ijl} + V^{(h)} * t^{(h)}_{ijl} + V^{(vdW)} * t^{(vdW)}_{ijl}) - \lambda \sum_{l}\sum_{j}(v_{jl} - \sum_{HULL} ch_{HULL.jl}) \quad (4.1)$$

where $V^{(b)}$ indicates the value of the non-covalent bond type b: ionic for $b = i$, hydrogen for $b = h$ and van der Waals for $b = vdW$. $t^{(b)}_{ijl}$ is the variable indicating if the surface atom i and the l^{th} atom of fragment number j builds this type of a non-covalent bond, v_{jl} is the variable indicating if l^{th} atom of j^{th} fragment exists in the ligand, $ch_{(HULL)\,jl}$ is 1 if this atom is present in the convex hull indexed as *HULL*, and λ is the penalty function parameter. The objective function aims to maximize the affinity that is the total value gained by the non-covalent bonds and to minimize the number of atoms in the ligand that are not placed

in one of the convex hulls. This relaxation was inspected in Chapter Three as a proposal in the case that the mathematical representation misses some space in the active site.

4.2.2 Variables

The model has to take many decisions leading to existence of many binary variables. The following paragraphs are dedicated to the binary variables.

There should be a variable indicating if a surface atom of the protein is interacted with the ligand or not.

$$p_i = \begin{cases} 1, \text{ if surface atom i is interacted} \\ 0, \text{otherwise} \end{cases} \tag{4.2}$$

The number of fragments to be used in the ligand is not definite, so a variable as the following may be useful.

$$q_j = \begin{cases} 1, \text{ if fragment number j is used} \\ 0, \text{ otherwise} \end{cases} \tag{4.3}$$

The maximum number of fragments that can be placed is calculated as follows based on Lipinski's weight constraint saying that the ligand should not exceed 500g/mol.

$$j_{\max} = \left\lfloor \frac{500 g/mol}{mw_{\min}} \right\rfloor \tag{4.4}$$

Here, mw_{min} indicates the minimum molecular weight among all the fragments present in the fragment library.

If each fragment in the library is accepted as a certain type, then the type of fragment number j used in the ligand has major importance. It should be noted here that the fragment instances in the libraries themselves are not used in the ligand. Rather, one instance among the multiple copies of all fragment types is selected to be the j^{th} fragment in the ligand.

$$q^{(t)}{}_{gj} = \begin{cases} 1, \text{ if fragment number j is of type g} \\ 0, \text{ otherwise} \end{cases} \quad (4.5)$$

Selection of the fragments depends on the conformation of the ligand in a whole and the number of interaction it makes. Therefore, the model should decide which fragment should be selected and bound to which fragment and if it can interact with a surface atom.

$$qq_{jk} = \begin{cases} 1, \text{ if fragment number j and k are covalently bound} \\ 0, \text{ otherwise} \end{cases} \quad (4.6)$$

$$pq_{ijl} = \begin{cases} 1, \text{ if surface atom i makes non - covalent bond} \\ \quad \text{with } 1^{\text{th}} \text{ atom of fragment number j} \\ 0, \text{ otherwise} \end{cases} \quad (4.7)$$

The fragment library is composed of coupled fragments that are covalently bound to each other. Two fragments can be bound from different atoms and thus the outputs have different conformations. The model should decide which conformation to use if fragments j and k are to be building covalent bonds.

$$qq^{(c)}{}_{jkghc} = \begin{cases} 1, \text{ if } c^{\text{th}} \text{ conform. of bound fragment types g and h is used} \\ \quad \text{as the template for bound fragment numbers j and k} \\ 0, \text{ otherwise} \end{cases} \quad (4.8)$$

The model then selects based on the previous variable between which atoms of the bound fragments takes this covalent bond place.

$$qq^{(cb)}{}_{jklm} = \begin{cases} 1, \text{ if } l^{\text{th}} \text{ atom of fragment number k makes covalent bond} \\ \quad \text{with } m^{\text{th}} \text{ atom of fragment number k} \\ 0, \text{ otherwise} \end{cases} \quad (4.9)$$

$$lig^{(cbw)}{}_{\mu\nu} = \begin{cases} 1, \text{ if ligand atoms } \mu \text{ and } \nu \text{ make covalent bond due to} \\ \quad \text{a covalent bond within a fragment} \\ 0, \text{ otherwise} \end{cases} \quad (4.10)$$

$$lig^{(cbb)}{}_{\mu\nu} = \begin{cases} 1, & \text{if ligand atoms } \mu \text{ and } \nu \text{ make covalent bond due to} \\ & \text{a covalent bond between two fragments} \\ 0, & \text{otherwise} \end{cases} \quad (4.11)$$

$$lig^{(cb)}{}_{\mu\nu} = \begin{cases} 1, & \text{if ligand atoms } \mu \text{ and } \nu \text{ makes covalent bond} \\ 0, & \text{otherwise} \end{cases} \quad (4.12)$$

When building covalent bonds, it is evident that one or more atoms of each fragment are replaced with the atom of the partner fragment with which the bond is made. This should be taken into account, since this replaced atom cannot be used for future bonds anymore, and does not take place in the ligand.

$$r_{jl} = \begin{cases} 1, & \text{if } l^{\text{th}} \text{ atom of fragment number j is replaced for bond making} \\ 0, & \text{otherwise} \end{cases} \quad (4.13)$$

A fragment atom is either existing in the ligand or not, either because the atom is replaced, or because the fragment number *j* is not selected to take place in the ligand at all.

$$v_{jl} = \begin{cases} 1, & \text{if lth atom of fragment number j exists in the ligand} \\ 0, & \text{otherwise} \end{cases} \quad (4.14)$$

For understanding the nature of the interaction between the surface atom and the fragment atoms, each fragment's capability of making a non-covalent bond should be known.

Fragments may either be neutral, which means that they cannot make a hydrogen or ionic bond but only van der Waals bonds, or they may have atoms that are hydrogen donors, hydrogen acceptors, negatively charged or positively charged. The importance of the nature of the non-covalent bonds can be seen in the objective function, since it calculates the value of each type of interaction separately.

$$t^{(i)}{}_{ijl} = \begin{cases} 1, & \text{if } l^{\text{th}} \text{ atom of fragment number j makes} \\ & \text{ionic bond with surface atom i} \\ 0, & \text{otherwise} \end{cases} \quad (4.15)$$

$$t^{(h)}_{ijl} = \begin{cases} 1, \text{ if } l^{th} \text{ atom of fragment number j makes} \\ \quad \text{hydrogen bond with surface atom i} \\ 0, \text{ otherwise} \end{cases} \tag{4.16}$$

$$t^{(v)}_{ijl} = \begin{cases} 1, \text{ if } l^{th} \text{ atom of fragment number j conducts} \\ \quad \text{van der Waals interation with surface atom i} \\ 0, \text{ otherwise} \end{cases} \tag{4.17}$$

The following variable indicates to which atom of the ligand a selected fragment atom corresponds.

$$ca_{jl\mu} = \begin{cases} 1, \text{ if } l^{th} \text{ atom of fragment number j is used} \\ \quad \text{as the } \mu^{th} \text{ atom of the ligand} \\ 0, \text{ otherwise} \end{cases} \tag{4.18}$$

Along with all these decisive variables, the coordinate of each atom of the ligand is to be determined.

$$Xc_{\mu} = \begin{bmatrix} xc_{\mu} \\ yc_{\mu} \\ zc_{\mu} \end{bmatrix} : \text{coordinates of the } \mu^{th} \text{ atom of the ligand} \tag{4.19}$$

The coordinates of the ligand atoms will be equal to their corresponding fragment atoms

$$Xf_{jl} = \begin{bmatrix} xf_{jl} \\ yf_{jl} \\ zf_{jl} \end{bmatrix} : \text{coordinates of the } l^{th} \text{ atom of the } j^{th} \text{ fragment} \tag{4.20}$$

Then, one more binary variable should be devoted to the convex hulls indicating if a fragment atom is placed in the $HULL^{th}$ convex hull or not.

$$ch_{HULL.jl} = \begin{cases} 1, \text{ if } l^{th} \text{ atom of fragment number j is in convex hull HULL} \\ 0, \text{ otherwise} \end{cases} \tag{4.21}$$

Finally, since all atoms should be connected to each other in the ligand, we introduce a flow $F_{\mu\nu}$.

$F_{\mu\nu}$ *:* flow from atom μ to ν (4.22)

The reason for the existence of this variable will be explained in 4.2.4.

4.2.3 Parameters

The variables being introduced, the necessary parameters are to be presented.

$\boldsymbol{A_{(HULL)p}}$, $\boldsymbol{B_{(HULL)p}}$, $\boldsymbol{C_{(HULL)p}}$ and $\boldsymbol{D_{(HULL)p}}$ **:** parameters of the system of linear inequalities Ax + By + Cz ≤ D of plane *p* belonging to the $HULL^{th}$ convex hull.

$\boldsymbol{V^{(i)}}$, $\boldsymbol{V^{(h)}}$, $\boldsymbol{V^{(vdW)}}$ **:** values of an ionic bond, hydrogen bond and van der Waals interaction respectively

$\boldsymbol{ats_{i(at)}}$ **:** atom type *at* of surface atom *i*. (N,C,O,H,S…)

$\boldsymbol{atf_{jl\,(at)}}$ **:** atom type at of fragment j atom l. (N,C,O,H,S…)

$\boldsymbol{vdW_{(at)}}$ **:** vdW distances of every type of atom

$\boldsymbol{mw_g}$ **:** molecular weight of fragment type *g*

$\boldsymbol{Xa_i} = \begin{bmatrix} xa_i \\ ya_i \\ za_i \end{bmatrix}$ **:** coordinates of the surface atoms

$\boldsymbol{Ta^{(n)}_i}$ = 1 if surface atom *i* is neutral, 0 otherwise.

$\boldsymbol{Ta^{(i)}_i}$ = 1 if surface atom *i* is positively charged, 0 if negatively charged.

$\boldsymbol{Ta^{(h)}_i}$ = 1 if surface atom *i* is hydrogen donor, 0 if acceptor.

$\boldsymbol{Tf^{(n)}_{gw}}$ = 1 if w^{th} atom of fragment *g* is neutral, 0 otherwise

$\boldsymbol{Tf^{(i)}_{gw}}$ = 1 if w^{th} atom of fragment *g* is negatively charged, 0 if positively charged.

$Tf^{(h)}_{gw}$ = 1 if w^{th} atom of fragment *g* is a hydrogen acceptor, 0 if donor.

$mind^{(i)}$, $mind^{(h)}$, $mind^{(vdW)}$ and $maxd^{(i)}$, $maxd^{(h)}$, $maxd^{(vdW)}$ **:** min and max distances of for each type of non-covalent bonds

$Dist^{(c)}_{ghcwx}$ = distance between two bound atoms *w* and *x* of conformation *c* formed by fragment types *g* and *h*

θ_{ghcwxy} **:** bond angle between the three connected atoms *s t* and *u* of conformation *c* formed by fragment types *g* and *h*

$\omega_{ghcwxyz}$ **:** torsional angle formed by the bound atoms *s t u* and *v* within conformation *c* formed by fragment types *g* and *h*

Rep_{ghcwx} = 1, if w^{th} atom of fragment g and x^{th} atom of fragment h is replaced to form c^{th} conformation of g and h, 0, otherwise.

ffb_{ghcwx} = 1 if w^{th} atom of fragment g and x^{th} atom of fragment h make bond

$fb_{gw(w')}$ = 1 if w^{th} and $(w')^{th}$ atoms of fragment type g are connected

D_{μ} = demand of atom μ

4.2.4 Constraints

Existential constraints:

Firstly, an atom l of fragment j should exist if its fragment is selected and it is not replaced, and cannot exist if these two constraints do not hold.

$$v_{jl} \le r_{jl} \qquad \text{for } \forall j, l. \qquad (4.23)$$

$$v_{jl} \le q_j \qquad \text{for } \forall j, l. \qquad (4.24)$$

$$-q_j + r_{jl} + v_{jl} \ge 0 \qquad \text{for } \forall j\ , l. \qquad (4.25)$$

A fragment type g can be used as the j[th] fragment only if the j[th] fragment is used.

$$q^{(t)}{}_{gj} \le q_j \qquad \text{for } \forall \text{g, j.} \qquad (4.26)$$

Fragment j and k can make bond if both are selected.

$$q_j - qq_{jk} \ge 0 \qquad \text{for } \forall \text{j, k.} \qquad (4.27)$$

For the l[th] atom of fragment j to be used as the ligand's μ[th] atom, the l[th] atom of fragment j should exist.

$$ca_{jl\mu} \le v_{jl} \qquad \text{for } \forall \text{j, l, } \mu. \qquad (4.28)$$

To use the c[th] conformation of the fragment types g and h as the bound j-k couple, j should be of fragment type g and k should be of fragment type h.

$$\begin{aligned} qq^{(c)}{}_{jkghc} &\le q^{(t)}{}_{gj} \\ qq^{(c)}{}_{jkghc} &\le q^{(t)}{}_{hk} \end{aligned} \qquad \text{for } \forall \text{j, k, g, h, c.} \qquad (4.29)$$

Constraints related to non-covalent bonding:

Surface atom i can make non-covalent bond with l[th] atom of fragment j if fragment j and surface atom i both make interaction, and if l[th] atom of fragment j exists.

$$\begin{aligned} p_i - pq_{ijl} &\ge 0 \\ q_j - pq_{ijl} &\ge 0 \\ pq_{ijl} &\le v_{jl} \end{aligned} \qquad \text{for } \forall \text{i, j, l.} \qquad (4.30)$$

Surface atom i can make a non-covalent bond with l[th] atom of fragment j if both atoms are *not* of neutral type.

$$pq_{ijl} \le 1 - Ta^{(n)}{}_i \qquad \text{for } \forall \text{i, j, l.} \qquad (4.31)$$

The type of the l[th] atom of fragment j can be determined from the type of the atom it corresponds to in the fragment library. So, if the fragment type g corresponds to j and the

atom index of the fragment type w corresponds to l, which is controlled by the condition "l = w", this non-covalent bond cannot be built if this atom is of neutral type.

$$pq_{ijl} \le qt_{gj} * (1 - Tf^{(n)}{}_{gw}) \qquad \text{for } \forall \text{ w = l, g, i, j.} \qquad (4.32)$$

The type of the non-covalent bond between surface atom *i* and the l^{th} atom of fragment *j* is ionic if such a bond exists and if one of the atoms is negatively charged and the other is positively charged. The same is also true for the hydrogen bond, where one of the atoms should be a hydrogen donor and the other a hydrogen acceptor.

$$\begin{aligned} t^{(i)}{}_{ijl} &\le pq_{ijl} \\ t^{(h)}{}_{ijl} &\le pq_{ijl} \end{aligned} \qquad \text{for } \forall \text{ i, j, l.} \qquad (4.33)$$

$$\begin{aligned} t^{(i)}{}_{ijl} &\ge pq_{ijl} * \left[Ta^{(i)}{}_{i} * Tf^{(i)}{}_{gw} + (1 - Ta^{(i)}{}_{i}) * (1 - Tf^{(i)}{}_{gw})\right] \\ t^{(h)}{}_{ijl} &\ge pq_{ijl} * \left[Ta^{(h)}{}_{i} * Tf^{(h)}{}_{gw} + (1 - Ta^{(h)}{}_{i}) * (1 - Tf^{(h)}{}_{gw})\right] \end{aligned} \qquad \text{for } \forall \text{ w = l, g, i, j.} \qquad (4.34)$$

There will be a minimum and a maximum distance of the non-covalent bond between surface atom *i* and the l^{th} atom of fragment *j* in the case that such a bond is established.

$$d_{ijl}{}^{(1)} - M.(1 - pq_{ijl}) \le \left\| Xa_i - Xf_{jl} \right\| \le d_{ijl}{}^{(2)} + M.(1 - pq_{ijl}) \quad \text{for } \forall \text{ i, j, l,} \qquad (4.35)$$

where

$$\begin{aligned} d_{ijl}{}^{(1)} &= t^{(i)}{}_{ijl} * m.ind^{(i)} + t^{(h)}{}_{ijl} * m.ind^{(h)} + t^{(vdW)}{}_{ijl} * m.ind^{(vdW)} \\ d_{ijl}{}^{(2)} &= t^{(i)}{}_{ijl} * m.axd^{(i)} + t^{(h)}{}_{ijl} * m.axd^{(h)} + t^{(vdW)}{}_{ijl} * m.axd^{(vdW)} \end{aligned} \qquad (4.36)$$

Constraints related to covalent bonding:

To use c^{th} conformation of fragments *g* and *h* as the *j-k* complex, *j* and *k* should be covalently bound.

$$qq^{(c)}{}_{jkghc} \le qq_{jk} \qquad \text{for } \forall \text{ j, k, g, h, c.} \qquad (4.37)$$

l^{th} atom of fragment number *j* makes covalent bond with m^{th} atom of fragment number *k* if the l^{th} and m^{th} atoms are used in the fragment and if the corresponding atoms make bond in the selected conformation c of fragment types g and h, and not otherwise.

$$qq^{(cb)}{}_{jklm} + 2 \ge v_{jl} + v_{km} + ffb_{ghcwx} * qq^{(c)}{}_{jkghc} \qquad \text{for } \forall \text{ l = w, m = x, j, k, g, h, c} \qquad (4.38)$$

$$qq^{(cb)}{}_{jklm} \le v_{jl} \qquad qq^{(cb)}{}_{jklm} \le v_{km} \qquad \text{for } \forall \text{ j, k, l, m} \qquad (4.39)$$

$$qq^{(cb)}{}_{jklm} \le ffb_{ghcwx} * qq^{(c)}{}_{jkghc} \qquad \text{for } \forall \text{ l = w, m= x, j, k, g, h, c} \qquad (4.40)$$

If l^{th} atom of fragment number *j* has to make covalent bond in complex *j-k* but has been replaced, this complex cannot be selected.

$$qq^{(c)}{}_{jkghc} \le (1 - r_{jl}) * \sum_{x} ffb_{ghcwx} \qquad \text{for } \forall \text{ l =w, j, k, g, h} \qquad (4.41)$$

A covalent bond between the ligand's μ^{th} atom and ν^{th} atom should take place if they are used.

$$lig^{(cb)}{}_{\mu\nu} \le \sum_{j}\sum_{l} ca_{jl\mu} \qquad lig^{(cb)}{}_{\mu\nu} \le \sum_{j}\sum_{l} ca_{jl\nu} \qquad \text{for } \forall \mu, \nu \qquad (4.42)$$

If μ^{th} and ν^{th} atoms of the ligand correspond to l^{th} and m^{th} atoms of the same fragment j, and if l^{th} and m^{th} atoms correspond to the w^{th} and w'^{th} atoms of fragment type g and if w^{th} and w'^{th} atoms are connected within fragment g, μ^{th} and ν^{th} atoms should be connected too.

$$lig^{(cbw)}{}_{\mu\nu} + 2 \geq fb_{gw(w')} * (ca_{jl\mu} + ca_{jm\nu}) + q^{(t)}{}_{gj} \qquad \text{for } \forall \text{ l = w, m = w', g, } \mu, \nu \tag{4.43}$$

On the other hand, if μ^{th} and ν^{th} atoms of the ligand correspond to l^{th} atom of fragment j and m^{th} atom of fragment k, and if these atoms' corresponding library fragment atoms w of fragment g and x of fragment h make covalent bond, μ^{th} and ν^{th} atoms should be connected too.

$$lig^{(cbb)}{}_{\mu\nu} + 2 \geq ca_{jl\mu} + ca_{km\nu} + qq^{(cb)}{}_{jklm} \qquad \text{for } \forall \text{ j, k, l, m, } \mu, \nu \tag{4.44}$$

If neither of the two conditions is satisfied, μ and ν are not connected,

$$2 * lig^{(cb)}{}_{\mu\nu} \leq lig^{(cbb)}{}_{\mu\nu} + lig^{(cbw)}{}_{\mu\nu} \qquad \text{for } \forall \mu, \nu \tag{4.45}$$

and if either one of the two conditions is satisfied, μ and ν are connected.

$$\begin{aligned} lig^{(cb)}{}_{\mu\nu} &\geq lig^{(cbb)}{}_{\mu\nu} \\ lig^{(cb)}{}_{\mu\nu} &\geq lig^{(cbw)}{}_{\mu\nu} \end{aligned} \qquad \text{for } \forall \mu, \nu \tag{4.46}$$

Constraints related to atom replacements:

An atom l of fragment j is replaced if j and k complex is used and the atom has to be replaced to use this conformation.

$$r_{jl} = qq^{(c)}{}_{jkghc} * R.ep_{ghcwx} \qquad \text{for } \forall \text{ w = l, j, k, g, h, c, x} \tag{4.47}$$

Fragment j cannot make a complex with fragment number k if the corresponding to-be-replaced atom has been replaced before.

$$qq_{jk} \leq (1 - r_{jl}) + (1 - R.ep_{ghcwx}) + M(1 - qt_{gj}) \qquad \text{for } \forall \text{ w = l, j, k, g, h, c, x} \tag{4.48}$$

Constraints related to atom coordinates:

Coordinate of the μ^{th} atom of the ligand will be same as the coordinate of its corresponding fragment atom's coordinate

$$\begin{aligned} Xc_{\mu} &\le Xf_{jl} + M*(1-ca_{jl\mu}) \\ Xc_{\mu} &\ge Xf_{jl} - M*(1-ca_{jl\mu}) \end{aligned} \quad \text{for } \forall \text{j, l, } \mu \qquad (4.49)$$

If an existing l^{th} atom of fragment j is in one of the convex hulls, its coordinates should obey all inequalities representing the convex hull, where convex hull *HULL* is defined by the system of linear inequalities ax + by + cz ≤ d, each representing a plane and the direction of the interior space.

$$a_p * xf_{jl} + b_p * yf_{jl} + c_p * zf_{jl} \le d_p + M*(1-ch_{(HULL).jl}) + M*v_{jl} \qquad \text{for } \forall \text{ HULL, p, j, l} \qquad (4.50)$$

Van der Waals radii of the fragment atoms should not collide with the vdW radii of all atoms.

$$\begin{aligned} &\| Xa_i - Xf_{jl} \|. \le \sum_{at} vdW_{(at)} * ats_{i.(at)} + \sum_{at} vdW_{(at)} * atf_{gw.(at)} + M * qt_{gj} \\ &\| Xf_{km} - Xf_{jl} \|. \le \sum_{at} vdW_{(at)} * atf_{gw.(at)} + \sum_{at} vdW_{(at)} * atf_{hx} + M * qt_{gj} + M * qt_{hk} \end{aligned} \qquad \text{for } \forall \text{ l=w, m=x, g, h i, j≠k} \qquad (4.51)$$

Bond distances between connected atoms μ and ν should be equal to the corresponding atoms' distances of the corresponding conformation c formed by fragment types g and h. This relationship is built through atoms l and m, which are atoms of a dual complex built by fragments j and k. They may be members of j, members of k, or one of them may be a member of j and the other a member of k. On the other hand, if these atoms are not expressed or do not make a bond, no restrictions are put forward.

$$\| Xc_\mu - Xc_\nu \| . \geq Dist_{ghcwx} - M * ((1 - lig^{(cb)}{}_{\mu\nu}) + (1 - (ca_{jl\mu} + ca_{km\mu})) + (1 - (ca_{jl\nu} + ca_{km\nu})) + (1 - qq^{(c)}{}_{jkghc}))$$

$$\| Xc_\mu - Xc_\nu \| . \leq Dist_{ghcwx} + M * ((1 - lig^{(cb)}{}_{\mu\nu}) + (1 - (ca_{jl\mu} + ca_{km\mu})) + (1 - (ca_{jl\nu} + ca_{km\nu})) + (1 - qq^{(c)}{}_{jkghc}))$$

for $\forall$ l = w, m = x, g, h, c, j, k, μ, ν (4.52)

Similar constraints are also required for the bond angles:

$$ang(Xc_\mu, Xc_\nu, Xc_o) \geq \theta_{ghcwxy} - z_1$$
$$ang(Xc_\mu, Xc_\nu, Xc_o) \leq \theta_{ghcwxy} + z_1 \quad (4.53)$$

where the constraints have to be taken into consideration for the cases if two of the atoms are from fragment j and if two of the atoms are from fragment k. Therefore, the two constraints will exist two times for each z_1 value presented below:

$$z_1{}^{(1)} = M * ((1 - lig^{(cb)}{}_{\mu\nu}) + (1 - lig^{(cb)}{}_{vo}) + (1 - (ca_{jl\mu} + ca_{km\mu} + ca_{jl.'\mu})) + (1 - (ca_{jl\nu} + ca_{km\nu} + ca_{jl.'\nu})) + (1 - (ca_{jlo} + ca_{kmo} + ca_{jl.'o})) + (1 - qq^{(c)}{}_{jkghc}))$$

for $\forall$ l = w, m = x, l' = y, g, h, c, j, k (4.54)

$$z_1{}^{(2)} = M * (((1 - lig^{(cb)}{}_{\mu\nu}) + (1 - lig^{(cb)}{}_{vo}) + (1 - (ca_{jl\mu} + ca_{km\mu} + ca_{km.'\mu})) + (1 - (ca_{jl\nu} + ca_{km\nu} + ca_{km.'\nu})) + (1 - (ca_{jlo} + ca_{kmo} + ca_{km.'o})) + (1 - qq^{(c)}{}_{jkghc}))$$

for $\forall$ l = w, m = x, m' = y, g, h, c, j, k (4.55)

Similar constraints for the torsional angles:

$$tors.ang(Xc_\mu, Xc_\nu, Xc_o, Xc_\pi) \geq \varpi_{ghcstuv} - z_2$$
$$tors.ang(Xc_\mu, Xc_\nu, Xc_o, Xc_\pi) \leq \varpi_{ghcstuv} + z_2 \quad (4.56)$$

where the same procedure is applied for each combinations of the four atoms for two fragments. Below is a z_2 for the case when two of the atoms are from fragment j and two are from fragment k.

$$z_2 = M * ((1 - lig^{(cb)}{}_{\mu\nu}) + (1 - lig^{(cb)}{}_{\nu o}) + (1 - lig^{(cb)}{}_{o\pi}) + (1 - (ca_{jl\mu} + ca_{km\mu} + ca_{jl'\mu} + ca_{km'\mu})) + (1 - (ca_{jl\nu} + ca_{km\nu} + ca_{jl'\nu} + ca_{km'\nu})) + (1 - (ca_{jlo} + ca_{kmo} + ca_{jl'o} + ca_{km'o})) + (1 - (ca_{jl\pi} + ca_{km\pi} + ca_{jl'\pi} + ca_{km'\pi})) + (1 - qq^{(c)}{}_{jkghc}))$$

for ∀ l = w, m = x, l' = y, m' = z, g, h, c, j, k (4.57)

There are cases when the three atoms whose bond angle are to be calculated and cases when the four atoms whose torsional angle are to be calculated are not from two fragments but more. The model does not take these cases into consideration with its current constraints. How these cases can be incorporated is a task proposed for future research.

The last but not the least is the constraint assuring that all atoms of the ligand are connected to each other. For this constraint to hold, the ligand should be considered like an undirected graph, where atoms act like the nodes and bonds act like the edges of the graph. For a graph to be connected there should be a path between every couple of nodes on the graph [75]. This can be assured if a certain demand is attached to every node and a supply to one of the nodes (denoted as ν' in the constraints), so that demands have to be supplied from the neighboring atoms and every neighbor should be communicating to at least one other node.

$$F_{\mu\nu} \le M * (1 - lig^{(cb)}{}_{\mu\nu}) \qquad \text{for } \forall\ \mu, \nu$$
$$F_{\nu\mu} \le M * (1 - lig^{(cb)}{}_{\nu\mu})$$

$$\sum_{\mu} F_{\mu\nu} - \sum_{o} F_{\nu o} = D_{\nu} \qquad \text{for } \forall\ \mu,\ \nu \neq \nu', o$$

$$\sum_{\mu} F_{\nu'\mu} = \sum_{\mu} D_{\mu} \qquad \text{for } \forall\ \mu, \nu$$

4.3 The Fragment Library

4.3.1 The Motivation

Fragment based drug design is based on the idea that a candidate ligand is to be built from a set of fragments, which are small molecules that can be found in the nature alone. It is assumed in this approach that organic compounds can be broken down into such basic fragments, and they can be combined together to form other organic structures. Fragment-based drug design is a more flexible approach than modifying a successful ligand or screening a database of available ligands [63]. The general method in fragment-based de novo design is to build hydrogen bonds with the surface atoms with fragments selected from a library and then to connect these with spacer fragments, which is an iterational approach in essence [45]. As studied in the Literature Review section, the main aim of fragment-based approaches is to obtain as many different structures as possible so that the possibility of achieving a successful ligand is increased.

This study proposes an idea about building a ligand by using optimization methods, which uses fragments as building blocks and forms energetically favorable structures. The model builds non-covalent bonds with the surface atoms and combine these fragments together according to an affinity maximization objective and subject to geometrical and chemical constraints. As explained in previous sections, the geometrical constraints are obtained from the convex hull algorithm, which will represent the feasible region into which the fragments can be placed. This section proposes a fragment library that can be used by such an optimization model.

4.3.2 Methods

The fragments used in this study are taken from the literature, a fragment set also used in a fragment-based drug design algorithm called LigBuilder developed by Wang et al. [20]. The fragments are presented in Figures 4.1, 4.2, 4.3 and 4.4. The set here aims to contain the fragments that are most commonly come across in available ligands. However, new fragments can be added to this set if the user sees the need.

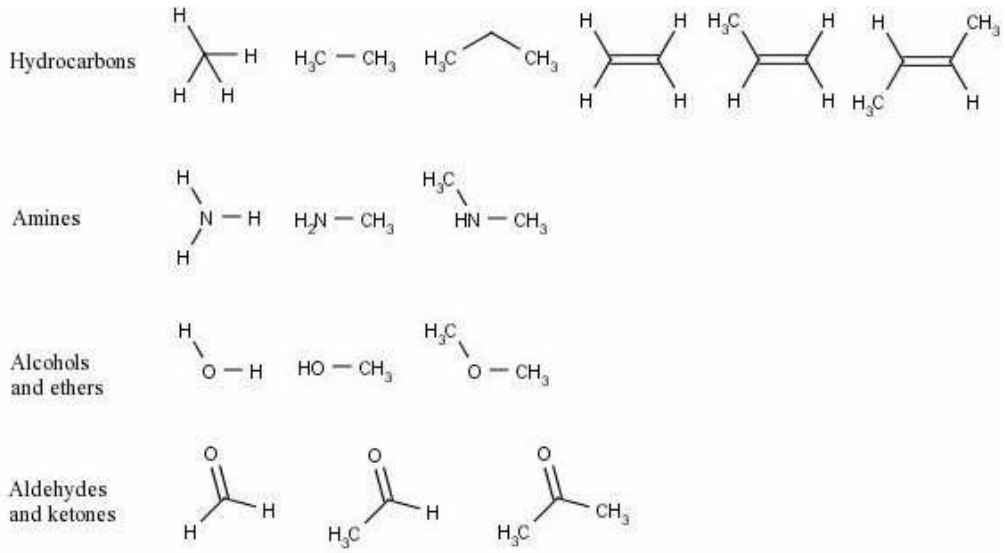


Figure 4.1: Hydrocarbons, amines, alcohols, ethers, aldehydes and ketones

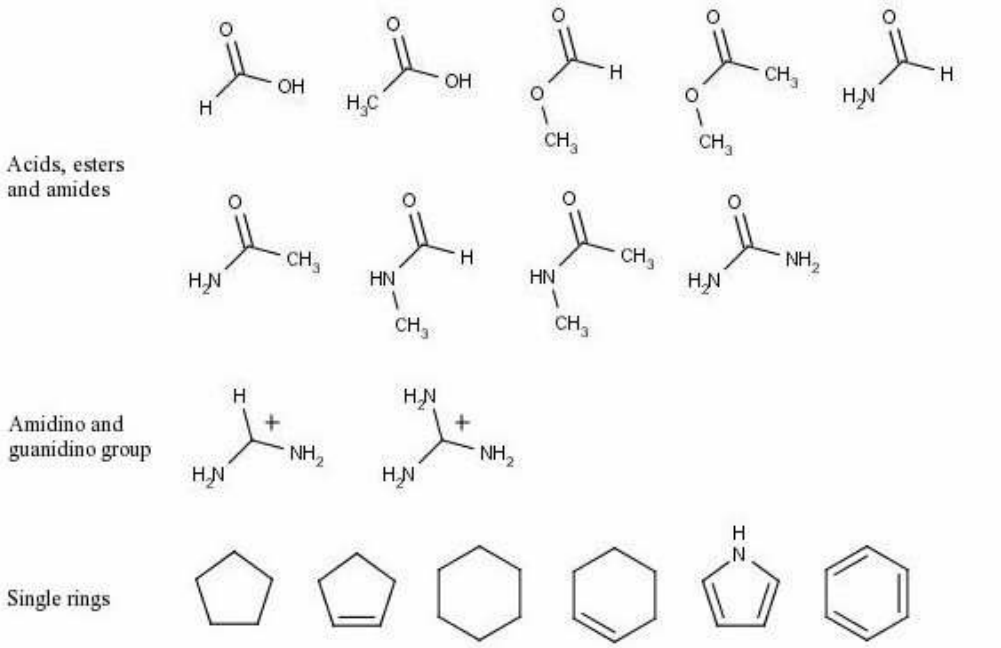


Figure 4.2: Acids, esters, amides, amidino and guanidine groups, single rings

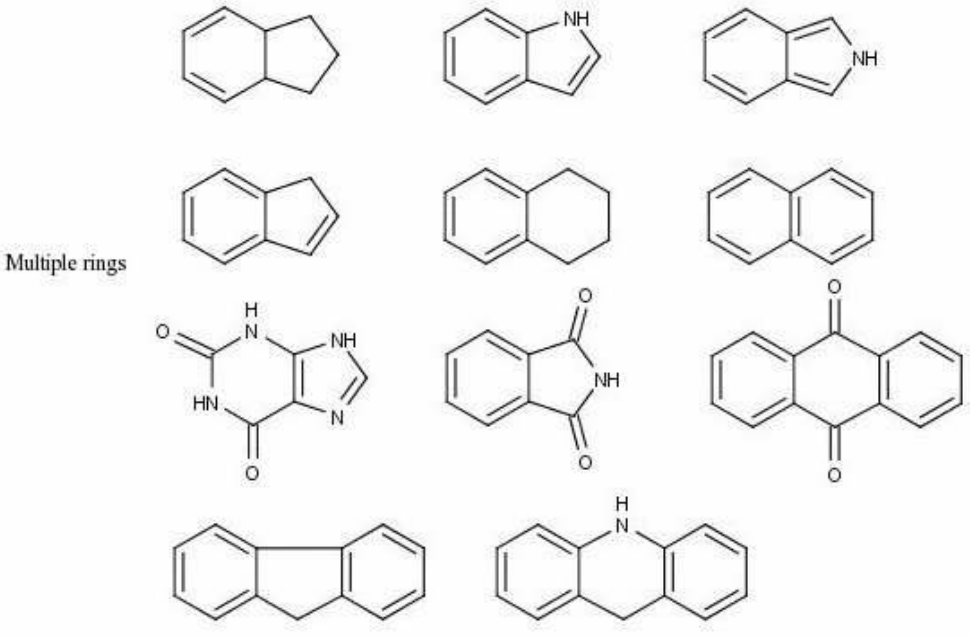


Figure 4.3: Multiple rings

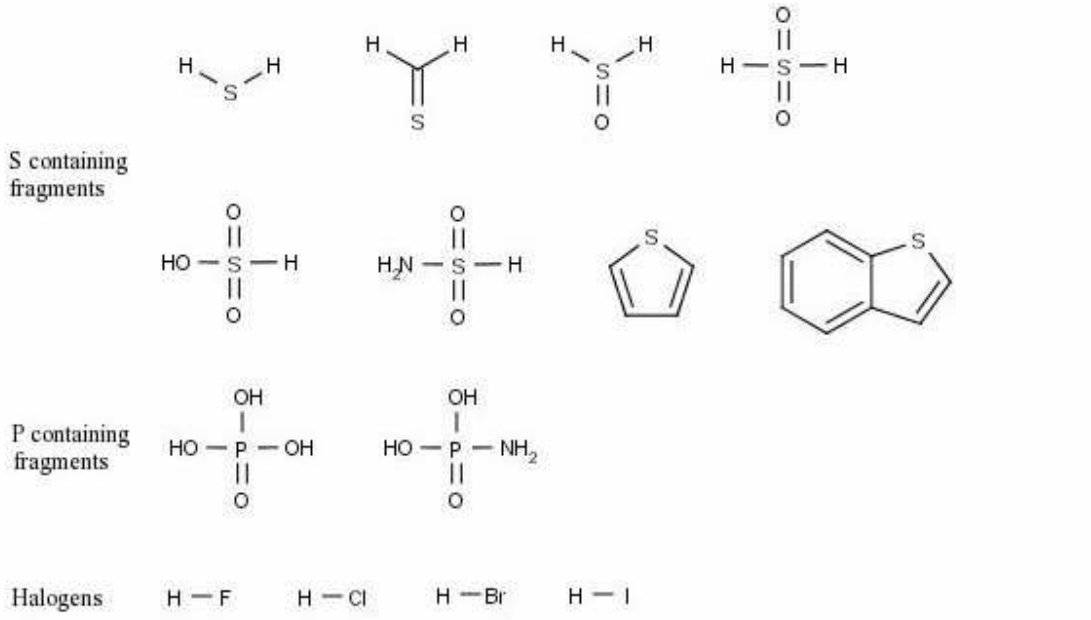


Figure 4.4: Fragments with sulphur, with phosphorus, and halogens

The fragments are drawn in MarvinSketch, an easy-to-use molecule editing program developed by ChemAxon that accomplishes some useful tasks such as molecular surface calculation or orbital electronegativity illustration. Then, the fragments are fed into an efficient library generation platform ILIB DIVERSE [72]. Chemical bonds between each pair of fragments are built by ILIB DIVERSE, which calculates the optimal 3D structures of possible binding conformations. These are then subjected to filtering for eliminating the outputs according to the criteria that are determined by the user. In this study, the molecules that are impossible to be found in nature, that are toxic and that have higher molecular weights than the default threshold of the program were eliminated.

Once all the feasible possible conformations of the combinations of fragment pairs, bond distances, bond angles and torsional angles were to be calculated. A bond angle is

defined as the angle formed by three bound atoms A, B, and C. . Given the center atom A, this angle is then the angle θ between two vectors $A\vec{B}$ and $A\vec{C}$, where

$$A\vec{B} = \begin{bmatrix} x_A - x_B \\ y_A - y_B \\ z_A - z_B \end{bmatrix} \text{ and } A\vec{C} = \begin{bmatrix} x_A - x_C \\ y_A - y_C \\ z_A - z_C \end{bmatrix} \qquad (4.1)$$

Since the dot product of two vectors $A\vec{B}$ and $A\vec{C}$ is calculated as:

$$A\vec{B} \bullet A\vec{C} = |A\vec{B}||A\vec{C}|\cos(\theta),$$

the angle θ can be calculated by the following formula:

$$\theta = \arccos\left(\frac{A\vec{B} \bullet A\vec{C}}{|A\vec{B}||A\vec{C}|}\right) \qquad (4.2)$$

Torsional angle (or dihedral angle) on the other hand, is the angle between two planes formed by consequently bound four atoms. As it can be seen in the figure below, the first plane is defined by atoms A, B and C, and the second one is determined by atoms B,C and D.

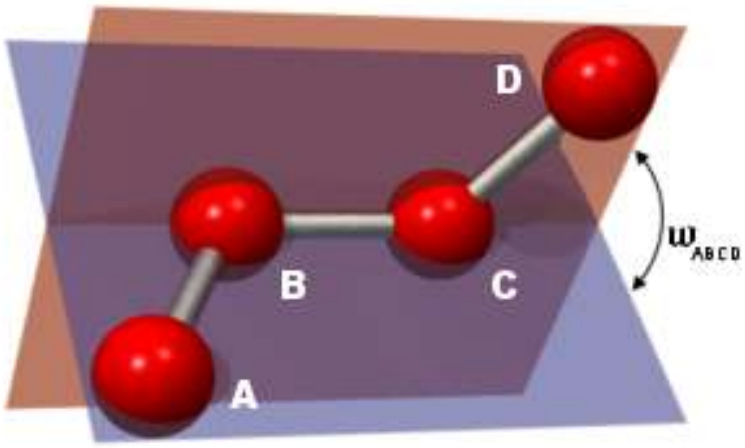


Figure 4.5: Torsional angle of four bound atoms [97]

The angle between two planes is equal to the angle between their surface normals. Normal vector of a plane is any vector that is perpendicular to the plane, i.e. a vector perpendicular to all vectors on the plane. For three points A, B and C on the plane, a normal vector is calculated by the cross product of the two vectors formed by these three atoms.

$$\vec{n} = A\vec{B} \times B\vec{C} \qquad (4.3)$$

To find the torsional angle, first the normal vectors of the two planes are calculated, and then the angle between two vectors can be found by using Formula 4.2.

The calculations for the bond distances, bond angles and torsional angles were coded in MATLAB to extract these parameters for all dual combinations of fragments. These calculations were conducted to be used in the optimization model for setting constraints on the fragment locations, thus on the whole structure's conformation.

4.3.3 Results

There are 57 fragments in the fragment set that led to a 1600 coupled fragments. Since each couple may have multiple conformations, it is not possible to present all outputs here. Nevertheless, one such example will be investigated.

The combination of amine (NH_3) and benzene (C_6H_{12}) fragments can be seen below. This combination led to one type of conformation for every possible N-C bond.

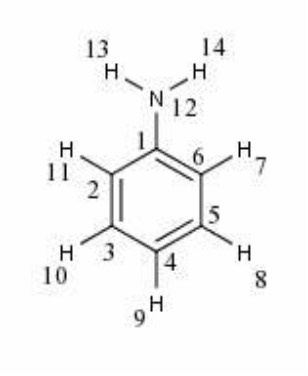


Figure 4.6: Amine bound to a benzene ring

center atom	side atoms		bond angle
1	2	6	119.653
1	2	12	120.172
1	6	12	120.175
2	1	3	120.17
2	1	11	120.45
2	3	11	119.38
3	2	4	120.04
3	2	10	119.992
3	4	10	119.968
4	3	5	119.949
4	3	9	119.98
4	5	9	120.071
5	4	6	120.107
5	4	8	119.933
5	6	8	119.96
6	1	5	120.078
6	1	7	120.506
6	5	7	119.414
12	1	13	120.214
12	1	14	120.158
12	13	14	119.628

Table 4.1: Bond angles of the amine-benzene complex

atoms forming plane 1			atoms forming plane 2			torsional angle between planes
1	2	6	1	2	12	179.982
1	2	6	1	6	12	0.017
1	2	6	2	1	3	179.671
1	2	6	2	1	11	0.003
1	2	6	6	1	5	0.139
1	2	6	6	1	7	179.838
1	2	12	1	6	12	179.982
1	2	12	2	1	3	0.346
1	2	12	2	1	11	179.979
1	2	12	12	1	13	170.948
1	2	12	12	1	14	895.044
1	6	12	6	1	5	0.122
1	6	12	6	1	7	179.820
1	6	12	12	1	13	903.449
1	6	12	12	1	14	171.067
2	1	3	2	1	11	179.674
2	1	3	2	3	11	0.322
2	1	3	3	2	4	0.517
2	1	3	3	2	10	179.454
2	1	11	2	3	11	179.677
2	3	11	3	2	4	0.195
2	3	11	3	2	10	179.776
3	2	4	3	2	10	179.972
3	2	4	3	4	10	0.028
3	2	4	4	3	5	0.235
3	2	4	4	3	9	179.842
3	2	10	3	4	10	179.972
3	4	10	4	3	5	0.263
3	4	10	4	3	9	179.814
4	3	5	4	3	9	179.923
4	3	5	4	5	9	0.077
4	3	5	5	4	6	0.233
4	3	5	5	4	8	179.684
4	3	9	4	5	9	179.923

Table 4.2: Torsional angles of the amine-benzene complex

atoms forming plane 1			atoms forming plane 2			torsional angle between planes
5	4	6	5	6	8	0.082
5	4	6	6	1	5	179.579
5	4	6	6	5	7	179.878
4	5	9	5	4	6	0.310
4	5	9	5	4	8	179.606
5	4	6	5	4	8	179.917
5	4	8	5	6	8	179.917
5	6	8	6	1	5	179.496
5	6	8	6	5	7	179.795
6	1	5	6	1	7	179.698
6	1	5	6	5	7	0.298
6	1	7	6	5	7	179.700
12	1	13	12	1	14	179.898
12	1	13	12	13	14	0.101
12	1	14	12	13	14	179.899

Table 4.2 cntd.: Torsional angles of the amine-benzene complex

Chapter 5

THE QSAR STUDY

Early prediction of activity-related properties of drug candidates is an important step in the drug design process. Characteristics such as toxicity and undesirable pharmacokinetic features often surface during clinical studies wasting time and resources. Therefore, there is need for computerized methods to predict such effects of drug candidates before the laboratory phase.

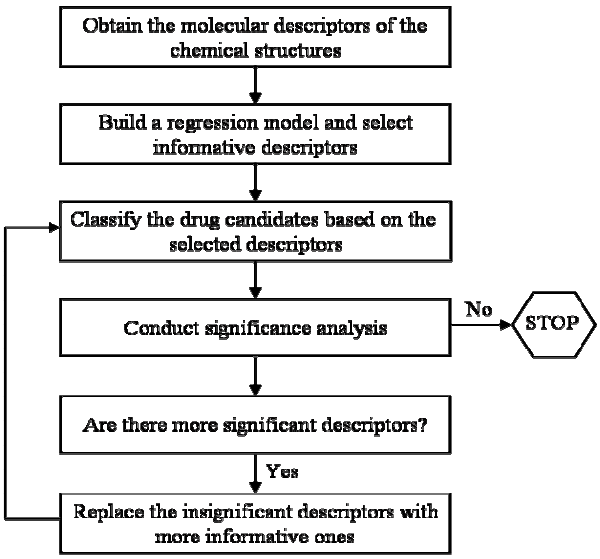
QSAR (quantitative structure-activity relationship) is one of the widely used methods for this purpose [15]. The method is a data mining procedure, which utilizes available experimental data to draw the correlation between the chemical structure of candidate molecules and their biological and chemical activities such as the biotransformation and reaction abilities, solubility and target activity [52]. The assumption here is that similar structures have similar chemical activities.

In this chapter, a QSAR method is used to predict the effectiveness levels of a class of drugs called calcium channel blockers (antagonists) for comparison purposes, since this class is widely examined in early prediction studies as presented in the Literature Review chapter. Activities of the molecules are classified into two classes based on their effectiveness levels: high-activity and low-activity. Drug effectiveness levels are based on their experimental IC_{50} values, i.e. the concentration of an inhibitor that inhibits 50% of the enzymatic reaction in consideration [76].

5.1 Strategies, Methods and Models

QSAR studies comprise a number of steps, and the methods used in each step have importance as to contributing the accuracy of the activity prediction. Traditionally, first, a set of numeric molecular attributes that describe various properties of the molecules, such as the number of double bonds or the molecular charge, is generated. These attributes are called molecular descriptors, and one may generate hundreds of such attributes for a given molecule. Therefore, there is need for a regression model that will select the most important descriptors that significantly affect the molecule's behavior. Once these major attributes are selected, the molecules are classified by a classification method that is trained on the selected descriptors' values of the available experimental data [53, 56-58].

Here, however, although following the idea of the traditional methodology, a different QSAR approach is presented. This approach is composed of a series of *iteratively* operated steps and involves a significance test procedure for the first time. The flowchart of the proposed methodology is presented below:



Flowchart 5.1: Steps of the proposed QSAR methodology

As it can be seen from the flowchart, after building the initial regression model for selecting the informative descriptors, the initial classification analysis is conducted. If the accuracy is not satisfactory, a significance test is run to see if there are more significant descriptors in terms of classification. The details of the significance test will be presented later on in this chapter.

5.1.1 Calculation of the Molecular Descriptors

Application of the QSAR method starts with the feeding procedure of the structures to the computer environment. The 3D structures of the molecules are drawn through energy minimization in HyperChem [77], and the molecular descriptors of the optimized structures are generated by CODESSA [78]. CODESSA calculates eight types of descriptors: constitutional, topological, geometrical, electrostatic, CPSA (charged partial surface area), MO- (molecular orbital-) related, quantum chemical, and thermodynamic, which sum up to 172 descriptors.

5.1.2 Regression Analysis

As the regression model to select the most informative descriptors, we used the PLS (partial least squares) method [79], which is used for the first time in a QSAR study on 1-4 DHP derivatives. PLS is an MLR (multiple linear regression) method, which describes a set of dependent variables Y as a linear combination of the set of independent variables X minimizing the distance between the actual Y values and the function values. The reason that PLS is selected in this study is that the method is especially successful in cases having many descriptors but few instances [15]. In this study, we have 172 descriptors and only 45 molecules.

The statistics software MINITAB [80] is utilized for regression runs. The PLS algorithm in MINITAB generates coefficients for the independent variables (molecular descriptor values) and a linear model for the dependent variable (the experimental IC_{50} value). The coefficients indicate how much the corresponding descriptors contribute to the IC_{50} level in the following way: the higher the absolute value of a descriptor's coefficient is, the more the descriptor is affecting the molecules behavior and vice versa. However, the coefficients are standardized first not to be affected by the very different magnitudes of the descriptors. Therefore, the descriptors that have the largest absolute valued standardized coefficients are selected as the most significant ones.

After building the regression model and selecting the most relevant descriptors, the classification study is to be conducted. However, here, the difference between regression and classification should be illustrated, since the regression coefficients that indicate the correlation between the independent variables and the dependent variable may not be that explanatory of the descriptors' effectiveness in terms of classifying the molecules. This issue is studied through the following example.

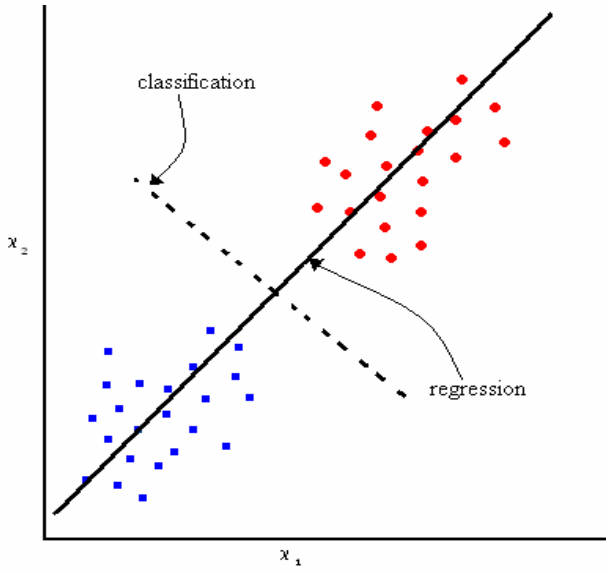


Figure 5.1: The difference between classification and regression

The figure illustrates a two-dimensional data having two classes represented by the squares and the circles. The black line represents the linear regression model of the data, and the slope of the line corresponds to the coefficient of the independent variable; whereas the dashed line is the line that separates the two classes, which has a completely different slope. This may also be implemented to higher dimensions.

Although there is such a difference between classification and regression, this study uses PLS to obtain a preliminary model that describes the effectiveness. Also, since there are many descriptors, PLS is useful for selecting a few of them initially to perform the initial classification analysis. Like the two dimensional case, some of the significant descriptors obtained from the regression analysis may not be very informative for the classification study, and there may be other descriptors that are more informative than these. Then these more informative ones should replace the less significant ones. One note should be made here: although there may be a replacement of descriptors, these new ones should also describe the dependent variable, thus cannot have very low standard coefficients. Moreover, as the number of selected descriptors increase, the probability that these really significant descriptors are selected in the regression analysis increases.

5.1.3 Classification

The classification of the molecules using the selected descriptors is carried out by utilizing the hyper-box classification method, which is a mixed-integer linear programming based model and is a very reliable classifier that can solve hard problems with high accuracy rates [17]. The algorithm encloses multidimensional data in hyper-boxes by solving an MILP, where hyperboxes are assigned to classes and data are assigned to hyperboxes. The strength of the algorithm is that it can assign multiple hyperboxes to a class, and thus allowing hyperboxes covering as few as a single instance each, which

considerably increases accuracy by minimizing overfitting. In this study, the hyperbox classification model is used to classify the 45 molecules into high and low activity classes. Initially, the model is run using the descriptors coming from the initial regression analysis, and then rerun after conducting the significance test. The classification runs will be repeated iteratively as long as there is an insignificant descriptor in the model to be replaced.

The classification method by itself is compared with other 51 classification methods available in the WEKA software [81]

5.1.4 Significance Test

The above explained problem of overrating the contributions of the descriptors to classification of the molecules into high and low active classes is solved by making a significance test after the classification run. The procedure followed in this test is as follows: Let *X* be the set of molecules in the data set and *D* be the set of selected descriptors for the classification analysis. Let *X* be separated into two classes after the classification process: *A* and *B*. If the classification run is successful, variances of the descriptor values for each class have to be smaller than the variance of the descriptor values of the whole set *X*. For a descriptor *i* in *D*, the sample variance of that descriptor's value for a molecule set *j* is represented by *Sij*. Then the following value for molecule sets *j* and *k* obeys the *F* distribution:

$$\frac{S_{ij}^{\ 2} / \sigma_i^{\ 2}}{S_{ik}^{\ 2} / \sigma_i^{\ 2}} = S_{ij}^{\ 2} / S_{ik}^{\ 2} = f_{\upsilon\eta} \qquad (5.1)$$

The $f_{\upsilon\eta}$ value is distributed under the F distribution with degrees of freedom υ and η. Here, $\upsilon = n - 1$ and $\eta = m - 1$, where *n* is the number of values descriptor *i* takes in drug set *j* and *m* is the number of values descriptors *i* takes in the drug set *k*.

In the significance test, the aim is to accept or reject a hypothesis by posing an alternative hypothesis and calculating the corresponding p-value. P-value of an outcome value is the probability of obtaining a statistic value at least as contradictory to the original hypothesis H_0 as the value of the outcome [82]. If the p-value is lower than the significance level, the null (original) hypothesis is rejected. Here, our null hypothesis is that the variance of the descriptor *i* values for the whole drug set *j* is smaller than or equal to the variance of the values for a molecule set *k*. Then, the alternative hypothesis H_a becomes the opposite of this claim, which is what we expect to happen from the results of a successful classification. Analytically:

$$H_0 = S_{ij} \leq S_{ik} \qquad (5.2)$$

and

$$H_a = S_{ij} > S_{ik} \qquad (5.3)$$

The significance level for this test is 0.1, i.e. the $f_{\upsilon\eta}$ value needs to have a p-value smaller than 0.1 for the null hypothesis to be rejected.

Three models are constructed with the regression analysis composed of 7, 10 and 15 of descriptors. The reason for building different models with different number of descriptors is to see the more significant descriptors in the larger models to replace the less significant ones in the smaller models. As it has been explained, the probability of selecting the significant descriptors and both the representative power of the regression analysis and the accuracy of the classification analysis improve as the number of selected descriptors increases [15]. This strategy is seen to be a successful approach and to be increasing the accuracy of the classification analyses.

5.2 Implementation to DHP Derivatives

The new QSAR approach was implemented on 45 variants of 1,4-d,hydropyridine calcium channel antagonists (DHP).

5.2.1 The Data Set

The 45 1,4-dihydropyridine calcium channel antagonists are taken from the literature for comparison purposes. The data set, used in this application was obtained from a template structure containing two ring structures. The diagram of this template is presented in Figure 5.2. The 45 DHP derivatives are constructed by attaching various fragments to the numbered positions of the X ring. These fragments and their positions can be seen in Table 5.1. Also, $\log(1/IC_{50})$ values of the molecules are provided in this table, which are used as a measure of drug efficacy. Since the IC_{50} value represents the concentration of the inhibitor necessary to reach a threshold inhibition level, a high IC_{50}, thus a low $\log(1/IC_{50})$ indicates low effectiveness. In this study, classification of the data set was conducted according to the cutoff value used in the literature [53,56,57,83] acquired in laboratory tests. Molecules that have $\log(1/IC_{50})$ values lower than 5.72 were categorized as low-activity drugs and are indicated by asterisks in Table 5.1 and the molecules having higher values were set as high-activity drugs [57].

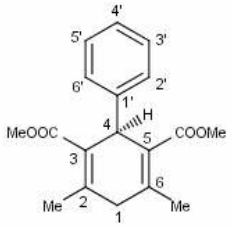


Figure 5.2: DHP derivatives template molecule

antagonist	X	log (1/IC50)	antagonist	X	log (1/IC50)
1	3'-Br	8.89	**23***	3'-NMe_2	6.05
2	2'-CF_3	8.8.2	**24***	3'-OH	6.00
3	2'-Cl	8.66	**25***	3'-NH_2	5.70
4	3'-NO_2	8.40	**26***	3'-OAc	5.22
5	2'-CH=CH_2	8.35	**27***	3'-OCOPh	5.20
6	2'-NO_2	8.29	**28***	2'-NH_2	4.40
7	2'-Me	8.22	**29**	4'-F	6.89
8	2'-Et	8.19	**30***	4'-Br	5.40
9	2'-Br	8.12	**31***	4'-I	4.64
10	2'-CN	7.80	**32***	4'-NO_2	5.50
11	3'-Cl	7.80	**33***	4'-NMe_2	4.00
12	3'-F	7.68	**34***	4'-CN	5.46
13	H	7.68	**35***	4'-Cl	5.09
14	3'-CN	7.46	**36**	2',6'-Cl_2	8.72
15	3'-I	7.38	**37**	F_5	8.36
16	2'-F	7.37	**38**	2'-F,6'-Cl	6.12
17	2'-I	7.33	**39**	2',3'-Cl_2	7.72
18	2'-OCH_3	7.24	**40**	2'-Cl,5'-NO_2	7.52
19	3'-CF_3	7.13	**41**	3',5'-Cl_2	7.03
20	3'-CH_3	6.96	**42**	2'-OH,5'-NO_2	7.00
21	2'-OC_2H_5	6.96	**43**	2',5'-Me_2	7.00
22	3'-OCH_3	6.72	**44***	2',4'-Cl_2	6.40
			45*	2',4',5'-$(OCH_3)_3$	3.00

Table 5.1: The data set and their experimental –log(IC_{50}) values

As explained in the previous section, the molecular structures were generated and their energy optimizations were carried out in HyperChem, and the molecular descriptors were computed by CODESSA. The program calculated 172 descriptors for the 45 molecules; however, 45 of these descriptors had the same or very close values for all of the members of the data set and thus were eliminated, since they could not provide any information regarding classifying these molecules. After the elimination process, the calculated number of descriptors reduced to 127. Because the number of attributes were too large compared to the number of instances, PLS was selected as the regression method, and for the

regression runs, the software MINITAB was used. The classification runs were made using the Hyper-Box model of Türkay and Üney [17].

5.2.2 Results

To explain the -log(IC_{50}) values of the data set, regression analysis was conducted. This model was used to obtain the starting descriptive attributes, which were used in the initial classification runs. Three models were built by the PLS runs: 7, 10 and 15 attribute regression models. The motivation to construct three models is have many descriptors at hand so that insignificant descriptors of the 7 and 10 attribute models can be replaced with the significant ones from the 15 attribute model to increase the accuracy of classification.

The PLS regression study was conducted in MINITAB. The program does not explicitly hand over a regression model and the selected attributes. Instead, it provides parameters such as the loadings, fits and coefficients. As explained before, the relevant attributes are the ones that have the largest standardized absolute value coefficients. Table 5.2 presents the initial selected descriptors, their standardized absolute value coefficients and their contributions to the regression model's R^2. Regression models having fewer than 7 attributes had very low R^2 values and thus were not considered.

Once the initial descriptors were obtained from the regression studies, three binary classification runs were made with the 7, 10 and 15 descriptors with the hyper-box classification algorithm. The molecules were classified as high active and low active by randomly selecting 66% of the data (29 descriptors) as the training set and the remaining as the test set. The results of the initial classification runs are presented in Table 5.3. The 10 and 15 attribute model reached an accuracy of 100%, whereas the 7 attribute model's accuracy needs to be improved, where the hyperbox method placed one molecule in both classes, hence the "half placements" observed in Table 5.3. As claimed before, it can be

inferred from the results that increased number of descriptors increase the classification accuracy. This is due to the fact that the $-\log(IC_{50})$ value is better explained with more attributes. The low accuracy result of the 7 attribute classification run indicates the possibility that there are insignificant descriptors in terms of classifying the data set into high and low activity molecules.

descriptor	abs std coeff	R^2
7-Attribute Model		
moment of inertia c	0.200060	0.3208
zx shadow/zx rectangle	0.190382	0.5327
yz shadow	0.160966	0.7060
moment of inertia b	0.141894	0.7887
relative number of double bonds	0.129344	0.8222
min partial charge (Qmin)	0.109508	0.8436
xy shadow/xy rectangle	0.108719	0.8545
10-Attribute Model		
yz shadow	0.362224	0.3208
min partial charge (Qmin)	0.237551	0.5327
relative number of rings	0.222727	0.7060
gravitation index (all bonds)	0.194466	0.7887
momen of inertia c	0.191894	0.8222
avg information content (first order)	0.287006	0.8437
zx shadow/zx rectangle	0.174063	0.8545
moment of inertia b	0.155643	0.8663
xy shadow/xy rectangle	0.146396	0.8832
avg structural information (first order)	0.138678	0.8997
15-Attribute Model		
yz shadow	0.690157	0.3208
relative number of rings	0.328803	0.5327
relative number of double bonds	0.276340	0.7060
min partial charge (Qmin)	0.271366	0.7887
gravitation index (all bonds)	0.262456	0.8222
topographic electronic index (first order)	0.244745	0.8437
xy shadow/xy rectangle	0.238495	0.8545
moment of inertia c	0.235894	0.8663
avg information content (first order)	0.223088	0.8832
Kier and Hall index (third order)	0.194655	0.8997
relative number of O atoms	0.169296	0.9052
avg structural information (first order)	0.164614	0.9126
number of O atoms	0.163997	0.9212
zx shadow	0.163665	0.9273
RPCS (relative positive charges)	0.150060	0.9319

Table 5.2: Initially selected descriptors for the three models

7-attribute model			10-attribute model		
	high	low		high	low
high	8	1	high	9	0
low	3.5	3.5	low	0	7
accuracy	71.875%		accuracy	100%	

Table 5.3: Confusion matrix for the initial models obtained by the Hyper-Box classifier

Once the initial classification runs were complete, a significance test on the class variances were conducted, and p-values for the models were obtained as explained in Section 4.1.4. A p-value below a certain α value indicates that the null hypothesis, where the variance of the whole set is claimed to be smaller than or equal to the variance of the respective class, can be rejected with a 1-α confidence against the alternative hypothesis, where the variance of the whole set is claimed to be larger than the class' variance. The p-values for the 7-descriptor model are presented in Table 5.4. Here, it can be observed that certain descriptors are more significant for different classes. The p-value of the descriptor "minimum partial charge" for the high-activity class is notably low indicating that this descriptor is significant in terms of classifying the data in the high-activity class. Also, the p-values of the descriptors "xy shadow / xy shadow rectangle" and "moment of inertia" are smaller than 0.2 for the low-activity class, which is due to the similarity of the values that these descriptors took in this class. This being the case, "relative number of bonds" and "zx shadow / zx rectangle" descriptors are considerably large for both classes. To see other descriptors' significance values, the significance test was also performed on the 10 and 15 descriptor models. The results of these tests are presented in Tables 5.5 and 5.6.

class	sample var	p value
	moment of inertia c	
all data	1.86433×10^{-12}	
high-activity class	2.45364×10^{-12}	0.69553
low-activity class	2.14177×10^{-13}	0.02475
	zx shadow/zx rectangle	
all data	4.78054×10^{-9}	
high-activity class	5.43046×10^{-9}	0.59965
low-activity class	4.97706×10^{-9}	0.58132
	yz shadow	
all data	3.31775×10^{-5}	
high-activity class	1.70621×10^{-5}	0.13479
low-activity class	7.54847×10^{-5}	0.89063
	moment of inertia b	
all data	3.06989×10^{-12}	
high-activity class	3.66341×10^{-12}	0.63273
low-activity class	1.48766×10^{-12}	0.25311
	relative number of double bonds	
all data	7.13736×10^{-10}	
high-activity class	6.56676×10^{-10}	0.45377
low-activity class	1.42946×10^{-9}	0.85419
	minimum partial charge	
all data	3.4584×10^{-12}	
high-activity class	1.7292×10^{-13}	0.00001
low-activity class	1.03752×10^{-11}	0.94724
	xy shadow/xy rectangle	
all data	3.79668×10^{-9}	
high-activity class	4.22108×10^{-9}	0.58499
low-activity class	1.45006×10^{-9}	0.18189

Table 5.4: Results of the significance tests for the initial classification run for the 7-descriptor model

class	sample var	*p* value
	moment of inertia c	
all data	1.86433×10^{-12}	
high-activity class	1.16597×10^{-12}	0.25525
low-activity class	1.35135×10^{-12}	0.36348
	zx shadow/zx rectangle	
all data	4.78054×10^{-9}	
high-activity class	6.01771×10^{-9}	0.66672
low-activity class	3.42624×10^{-9}	0.35775
	yz shadow	
all data	3.31775×10^{-5}	
high-activity class	2.11057×10^{-5}	0.26358
low-activity class	5.47244×10^{-5}	0.79829
	moment of inertia b	
all data	3.06989×10^{-12}	
high-activity class	1.42173×10^{-12}	0.13663
low-activity class	4.34283×10^{-12}	0.72743
	gravitation index (all bonds)	
all data	52,852.91667	
high-activity class	76,877.77778	0.74696
low-activity class	26,295.23810	0.19943
	minimum partial change	
all data	3.44650×10^{-12}	
high-activity class	2.17778×10^{-13}	0.00025
low-activity class	7.40571×10^{-12}	0.89243
	xy shadow/xy rectangle	
all data	3.79668×10^{-9}	
high-activity class	3.99492×10^{-9}	0.55730
low-activity class	1.93239×10^{-9}	0.20762
	avg information content (first order)	
all data	0.020193333	
high-activity class	0.026119444	0.68250
low-activity class	0.012628571	0.29218
	relative number of rings	
all data	0.001135933	
high-activity class	0.001102111	0.50635
low-activity class	0.001158476	0.55053

Table 5.5: Results of the significance tests for the 10-descriptor model

class	sample var	*p* value
	moment of inertia c	
all data	1.12216×10^{-13}	
high-activity class	7.02339×10^{-14}	0.25570
low-activity class	8.13117×10^{-14}	0.36331
	zx shadow	
all data	2.72483×10^{-6}	
high-activity class	1.04685×10^{-6}	0.08732
low-activity class	5.03054×10^{-6}	0.84296
	yz shadow	
all data	1.99608×10^{-6}	
high-activity class	1.26980×10^{-6}	0.26358
low-activity class	3.29242×10^{-6}	0.79829
	relative number of O atoms	
all data	6.35861×10^{-11}	
high-activity class	7.84348×10^{-11}	0.65476
low-activity class	5.42829×10^{-11}	0.45061
	relative number of double bonds	
all data	4.29394×10^{-11}	
high-activity class	4.76203×10^{-11}	0.59009
low-activity class	4.17239×10^{-11}	0.52315
	minimum partial charge	
all data	2.08129×10^{-13}	
high-activity class	1.38454×10^{-14}	0.00031
low-activity class	4.46041×10^{-13}	0.89167
	xy shadow/xy rectangle	
all data	2.28424×10^{-10}	
high-activity class	2.40350×10^{-10}	0.55730
low-activity class	1.16261×10^{-10}	0.20763
	number of O atoms	
all data	1.64377×10^{-7}	
high-activity class	1.59522×10^{-7}	0.50650
low-activity class	1.96183×10^{-7}	0.63843

Table 5.6: Results of the significance tests for the 15-descriptor model

class	sample var	*p* value
gravitation index (all bonds)		
all data	0.006598345	
high-activity class	0.009597693	0.74696
low-activity class	0.003282790	0.19943
relative number of rings		
all data	1.35371×10^{-12}	
high-activity class	6.73992×10^{-13}	0.16061
low-activity class	2.28121×10^{-12}	0.80728
avg information content (first order)		
all data	2.52100×10^{-9}	
high-activity class	3.26083×10^{-9}	0.68250
low-activity class	1.57660×10^{-0}	0.29219
RPCS		
all data	2.16152×10^{-8}	
high-activity class	3.30726×10^{-8}	0.77258
low-activity class	7.61959×10^{-9}	0.10254
Kier and Hall index (third order)		
all data	6.58340×10^{-9}	
high-activity class	6.12045×10^{-9}	0.47961
low-activity class	8.25509×10^{-9}	0.66517
avg structural information content (first order)		
all data	1.41814×10^{-10}	
high-activity class	1.37590×10^{-10}	0.50634
low-activity class	1.44629×10^{-10}	0.55053
topographic electronic index (first order)		
all data	6.23883×10^{-9}	
high-activity class	4.81514×10^{-9}	0.36704
low-activity class	9.10702×10^{-9}	0.74275

Table 5.6 (cntd.): Results of the significance tests for the 15-descriptor model

The insignificant descriptors of the 7-descriptor model were replaced by the significant descriptors of the 15-descriptor model, since more of the $-\log(IC_{50})$ values are described in this model. For the 7 attribute model the p-value of the descriptor "zx shadow / zx

rectangle" remains over 0.5 for both classes. Moreover, the p-value of "relative number of double bonds" is very high for the low-activity class and the remains above 0.4 for the high-activity class. Therefore, these two descriptors are to be removed and replaced by other two descriptors that are found to be significant from the 15-attribute model. The descriptor having the minimum p-values for this model was "minimum partial charge", but it was already in the 7-attribute model. "zx shadow" and "RPCS" received significantly low p-values: the former with a p-value of 0.08732 for the high-activity class and the latter with a p-value of 0.10254 for the low-activity class. (See Table 5.4) These were placed in the new 7-descriptor model. The descriptors selected after the significance test can be seen in Table 5.7.

selected descriptors
moment of inertia c
zx shadow
yz shadow
moment of inertia b
RPCS
min partial charge (Qmin)
xy shadow/xy rectangle

Table 5.7: Selected descriptors for the 7-descriptor model after significance tests

With these new descriptors, classification of the 45 molecules was repeated by the hyper-box model using the same training and test set rule. The result was a success: an accuracy of 100% was reached as can be observed from Table 5.8, where the accuracy and classifications are presented. Therefore, it can be said that the procedure offered based on significance testing and replacement of insignificant descriptors with significant ones paid off.

	high	low
high	9	0
low	0	7
accuracy	100%	

Table 5.8: Confusion matrix for the final 7-descriptor model obtained by the Hyper-Box classifier

The results were compared with the available classification methods in WEKA [81], a widely used data-mining tool. Some of the most frequently used classification methods from this tool are briefly studied below.

A multilayer perceptron [84] is basically a network built by processing elements called perceptrons that compute an output represented by a nonlinear activation function of a linear combination of multiple inputs. Another classification method is the logit boost, which is also called the additive logistic regression [85], which, as the name indicates, uses the logistic regression model for the learning phase. The bayesian network [86] is essentially a directed cyclic graph, whose nodes correspond to the variables and the edges between the nodes have probability values. Such a graph and the probabilities are first learned by the training run and this graph is then used to carry out inference by maximizing the likelihood. Naive Bayes simple and naive Bayes updatable [87] are methods that are also based on Bayesian networks. They both have the naive assumption that the variables are independent from each other, and build the graph and probabilities accordingly. K-star is the upgraded version of the k nearest neighborhoods method [88]. Both K-star and LWR (locally weighted regression) [89] are instance based, i.e. they build query specific local models. This is contrary to what neural networks or decision trees make, which is building global representations of the target functions. LWR assigns instance based weights to variables to build the regression model. Another statistical model is the logistic classifier [90] that constructs a logistic regression model having two classes. This method is a

generalized version of the least squares regression method. It assumes that the logarithm of the likelihood ratio of the distributions of the two classes is linear in the observations. SMO (sequential minimal optimization) [91] trains a support vector classifier by breaking a large quadratic programming (QP) optimization problem into smaller QP problems utilizing polynomial kernels. OneR (one rule) [87] learns a rule from each attribute of a one-level decision tree and then picks one rule that has the smallest error rate. A nice classifier that can handle multiple classes is the multiclass classifier [92], which handles the classes by utilizing other two-class distribution classifiers. A threshold-based classifier [87] uses a distribution classifier and aims to minimize the misclassification error by putting an upper threshold on the probability output. The last but not the least is the decision stump [93], which classifies data also by using a threshold that is captured by the maximum likelihood function.

For classification with the methods available in WEKA, 66% of the data set was separated into a training set and the remaining into a test set. The following table shows the results obtained by the most frequently used classifiers of WEKA and also the best results obtained from other methods. The outputs from the methods not presented here were very poor and was not comparable. Tuning parameters of the classifiers were WEKA's default values.

It can be seen from Table 5.9, the proposed method received better results than all of the classification methods available in WEKA did. Among the WEKA algorithms, the ones based on instances and distribution classifiers performed better than others. For the 7-attribute classification, the best result from WEKA turned out to be an accuracy of 68.75% whereas the hyperbox method received and accuracy rate of 100%. For the 10-attribute case, the best result achieved by WEKA algorithms was 75% whereas the hyperbox method again achieved 100%.

	accuracy (%)	
classification method	**7-attribute**	**10-attribute**
Bayes network	62.50	56.25
naive Bayes	43.75	50.00
naive Bayes simple	56.25	56.25
naive Bayes updatable	43.75	50.00
logistic	68.75	68.75
multilayer perceptron	50.00	62.50
SMO	62.50	62.50
K-star	62.50	75.00
LWR	56.25	68.75
logit boost	56.25	68.75
multiclass classifier	68.75	68.75
threshold selector	37.50	37.50
decision stump	62.50	75.00
oneR	43.75	75.00

Table 5.9: Accuracies obtained by the algorithms in WEKA

The methodology was also compared with the QSAR methods proposed in the literature and applied on the same data set of 45 DHP derivatives, and it was seen that it outperforms all these methods. The regression results can be seen in Table 5.10 and the classification results are presented in Table 5.11.

method	**number of variables**	R^2
MLR [83]	8	0.550
PCANN [83]	8	0.730
CoMFA [56]	8	0.872
CoMSIA [56]	8	0.908
GRID/GOLPE [56]	8	0.821
HM [57]	7	0.870
LSSVM [57]	7	0.870
PLS	7	0.854
PLS	10	0.899
PLS	15	0.932

Table 5.10: Regression results of the reference methods for the same data set

method	accuracy (%)
PCA [53]	82.2
BPNN [83]	88.9
LDA [57]	86.7
LSSVM [57]	91.1
hyperbox, 7-attribute	100.0
hyperbox, 10-attribute	100.0

Table 5.11: Accuracies of the reference methods for the same data set

The R^2 values indicate how successful is the linear model built by the regression model in describing the IC_{50} values of the molecules. However, it should be noted that this study used regression for selecting the most descriptive attributes from many, and then these selected ones were subjected to a significance test and were changed if seen necessary. Therefore, the R^2 values are not enough to measure the success of the methodology, but only the success of the regression method used. Although the R2 value of the PLS method with the 15-attribute case is better than the R2 values of the other methods, such a comparison would not be fair since these methods use less attributes to achieve their R2 values. Nevertheless, the PLS results for the 7-descriptor case still performs better than three of the other methods and seems comparable with the rest.

To compare the whole methodology that is composed of the regression, significance and classification studies, the classification accuracy is a better measure. The classifications results of the reference methods are presented in Table 5.11. It can be seen that the hyperbox method again achieved the best results among these methods. With this, the success of the methodology proposed in this thesis in classifying this data set into low and high active classes was established. It can be said that the 10 and 15-attribute models are not necessary for this data set, since a 100% accuracy was already achieved with the 7-attribute model.

Chapter 6

CONCLUSION and FUTURE WORK

The motivation of this thesis was to develop tools to be used for a fragment-based structure based drug design based on the idea that opmization itself can be used to design ligands. This idea is built on the prerequisite that the active site is represented by the union of a set of convex hulls, and a fragment library is available.

Proteins have massively many different types of shapes and an algorithm that can represent the active site of any protein is a hard problem. This study realized the first prerequisite partly achieving close representations for protein pockets that are completely buried, thus are completely surrounded with atoms. However, the algorithm has hard time describing the active sites that have large openings to the outer space. For such proteins, the constraint that there should be no atoms within the represented active site is offered to be relaxed. In this way, a larger volume of the active site can be represented and also the atoms within the site are close to the walls of the convex hulls, allowing the assumption that the mathematical programming model to be developed for this purpose can handle these by putting a constraint on the van der Waals regions of all atoms prohibiting placing a fragment atom within these regions.

The algorithm may be run by starting from different regions of the active site that is taking the initial surface from a different part of the site each time. The outputs may be taken into consideration together to be used as geometric constraints of the drug design

model as future work. Moreover, some dummy points that are thought to be helpful for the algorithm to extract better outputs may be introduced to the active site by the user.

Another task that may be useful is to further automatization of the SLICE algorithm. In the current edition, the user introduces the starting surface, but a method that will automatically detect the pocket mouth may be developed. The "unite" algorithm may also be studied so that it can optimize the surfaces to be united between the j^{th} and the $j+1^{st}$ convex hulls. However, this task is not trivial, since there is no rule that will optimize the united surfaces without including a point within the structure.

The second prerequisite was successfully implemented. A fragment library composed of feasible dual combinations of 57 fragments is built, and their bond distances, bond angles and torsional angles were calculated. These will hopefully be used in future studies as parameters for geometrical constraints of the optimization model.

The main part of the project, i.e. the mathematical programming algorithm that energetically optimizes the structure comprising the protein active site and fragments by binding fragments together and building non-covalent bonds with the site surface was attempted. A conceptual model was obtained, which needs further development before it can be used for drug design purposes. The constraints for the bond angles and torsional angles have to be developed further for the cases when atoms whose bond or torsional angles are to be calculated belong to more than one fragment type. Besides these constraints, there may be a number of new constraints to be posed. However, many presumably valid constraints, variables and parameters were introduced in this study. Therefore, it is believed to be having achieved a starting point for future developers.

Lastly, an evaluation strategy for the activities of generated ligands was developed. This strategy was essentially a QSAR study utilizing a home-made classification algorithm developed by Üney and Türkay. The methodology offered in the study was proved to be

achieving better accuracy results than all of the reference methods for a dataset of DHP derivatives.

As a conclusion, this thesis initiated a fragment-based drug design project by proposing an algorithm for mathematical representation of the active site of proteins, building a basis for the optimization model for drug design purposes and presenting a methodology for early prediction of the activities of the designed ligands.

BIBLIOGRAPHY

[1] G. Economou, J. N. Ward-McQuaid, A Cross-over Comparison of the Effect of Morphine, Pethidine, Pentazocine, and Phenazocine on Biliary Pressure, J. Gut, 12 (1971), 218-221.

[2] Richard B. Silverman, The Organic Chemistry of Drug Design and Drug Action, Academic Press Ltd, London, (2004).

[3] Regine S. Bohacek, Colin McMartin, Wayne C. Guida, The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective, Medicinal Research Reviews, 16, 1, (1996), 3 – 50.

[4] Hans J. Wolters, Geometric Modeling Applications in Rational Drug Design: A Survey, J. Computer Aided Geometric Design, 23, 6 (2006), 482-494.

[5] Hugo Kubinyi, Combinatorial and Computational Approaches in Structure-Based Drug Design, J. Current Opinion in Drug Discovery and Development, 1, (1998), 16-27.

[6] Mati Karelson, Molecular Engineering and Drug Design, in Changing the Way Research is Done, European Commission, Brussels, (2003).

[7] Amy C. Anderson, The Process of Structure-Based Drug Design, J. Chemistry & Biology, 10, (2003), 787–797.

[8] A. T. Laurie, R. M. Jackson, Q-SiteFinder: An Energy-Based Method for the Prediction of Protein-Ligand Binding Sites, J. Bioinformatics, 21, (2005), 1908–1916.

[9] M. Dixon, E. C. Webb, C. J. R. Thorne, K. F. Tipton, in Enzymes (3rd edition), Longman, London, (1979), 220-243.

[10] Murad Nayal and Barry Honig, On the Nature of Cavities on Protein Surfaces: Application to the Identification of Drug-Binding Sites, J. Proteins: Structure, Function, and Bioinformatics, 63, (2006), 892–906.

[11] A. S. Aytuna, A. Gursoy, O Keskin, Prediction of Protein–Protein Interactions by Combining Structure and Sequence Conservation in Protein Interfaces, J. Bioinformatics, 21(12), (2005), 2850-2855.

[12] Paul J. Gane, Philip M. Dean, Recent Advances in Structure-Based Rational Drug Design, J. Current Opinion in Structural Biology, 10, (2000), 401–404.

[13] R.D. Taylor, P.J. Jewsbury and J.W. Essex, A Review of Protein-Small Molecule Docking Methods, Journal of Computer-Aided Molecular Design, 16(3), (2002), 151-166.

[14] I. M. Kapetanovic, Computer-Aided Drug Discovery and Development (CADDD): In Silico-Chemico-Biological Approach, J. Chemico-Biological Interactions, 171, (2008), 165–176.

[15] P. Kahraman, M. Turkay, Classification of 1,4-Dihydropyridine Calcium Channel Antagonists Using the Hyperbox Approach, J. Ind. Eng. Chem. Res., 46, (2007), 4921-4929.

[16] T. Solmajer, J. Zupan, Optimization Algorithms and Natural Computing in Drug Discovery, J. Drug Discovery Today: Technologies, 1 (3), (2004), 247-252.

[17] F. Uney, M. Turkay, A Mixed-Integer Programming Approach to Multiclass Data Classification Problem, Eur. J. Oper. Res., 173 (3), (2006), 910-920.

[18] J. Tooze, C. Branden, Introduction to Protein Structure. USA: Garland Publishing Inc., (1999), 251-281.

[19] C. Sotriffer, G. Klebe, Identification and Mapping of Small-Molecule Binding Sites in Proteins: Computational Tools for Structure-Based Drug Design, J. Il Farmaco, 57, (2002), 243–251.

[20] Renxiao Wang, Ying Gao, and Luhua Lai, LigBuilder: A Multi-Purpose Program for Structure-Based Drug Design, J. Mol. Model, 6, (2000), 498 – 516.

[21] J. H. M. Berman, Z. Westbrook, G. Feng, T. N. Gilliland, H. Bhat, I. N. Weissig, P. E. Bourne Shindyalov, The Protein Data Bank, J. Nucleic Acids Research, 28, (2000), 235-242.

[22] J. An, M. Totrov, R. Abagyan, Comprehensive Identification of "Druggable" Protein Ligand Binding Sites, J. Genome Informatics, 15(2), (2004), 31–41.

[23] M. S. Johnson, N. Srinivasan, R. Sowdhamini, T. L. Blundell, Knowledge-Based Protein Modeling, J. Crit. Rev. Biochem. Mol. Biol., 29, (1994), 1-68.

[24] Roman A. Laskowski, SURFNET: A Program for Visualizing Molecular Surfaces, Cavities, and Intermolecular Interactions, Journal of Molecular Graphics, 13 (5), (1995), 323-330.

[25] David G. Levitt and Leonard J. Banaszak, POCKET: A Computer Graphics Method for Identifying and Displaying Protein Cavities and Their Surrounding Amino Acids, J Mol Graph., 10 (4), (1992), 229-34.

[26] C. K. Knox, J. C. Wenstrom, 3D Visualization of Neural Structures, in Proceedings, Society for Computer Simulation, Eastern Multi-Conference, Nashville, (1990) 12-17.

[27] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel, LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-Binding Sites in Proteins, Journal of Molecular Graphics and Modelling, 15, (1997), 359–363.

[28] G. Patrick Brady Jr., Pieter F. W. Stouten, Fast Prediction and Visualization of Protein Binding Pockets with PASS, Journal of Computer-Aided Molecular Design, 14 (4), (2000), 383-401.

[29] J. Liang, H. Edelsbrunner, C. Woodward, Anatomy of Protein Pockets and Cavities: Measurement of Binding Site Geometry and Implications for Ligand Design

[30] D. T. Lee and B. J. Schachter, Two Algorithms for Constructing a Delaunay Triangulation, International Journal of Parallel Programming, 9 (3), (1980), 219-242.

[31] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, S. Subramaniam, Analytical Shape Computation of Macromolecules: I. Molecular Area and Volume Through Alpha Shape, J. Proteins: Structure, Function, and Bioinformatics, 33 (1), (1998), 1-17.

[32] Ryan G. Coleman, and Kim A. Sharp, Travel Depth, a New Shape Descriptor for Macromolecules: Application to Ligand Binding, Journal of Molecular Biology, 362 (3), (2006), 441-458.

[33] C.B. Barber, D. P. Dopkin, H. T. Huhdanpaa, The Quickhull Algorithm for Convex Hulls, ACM Trans. on Mathematical Software, 22(4), (1996), 469-483.

[34] R. K. Ahuja, K.Mehlhorn, J. Orlin, R. E. Tarjan, Faster Algorithms for the Shortest Path Problem, Journal of the ACM (JACM), 37 (2), (1990), 213-223.

[35] J. An, M. Totrov, R. Abagyan, Pocketome via Comprehensive Identification and Classification of Ligand Binding Envelopes**,** J. Molecular and Cellular Proteomics, 4, (2005), 752-761.

[36] M. Weisel, E. Proshak, G. Schneider, PocketPicker: Analysis of Ligand Binding-Sites with Shape Descriptors, Chemistry Central Journal, 1, (2007), 1-17.

[37] Ernesto Freire, Thermodynamics Guide to Affinity Optimization of Drug Candidates, Protein Reviews vol 3, ed. J. E. Ladbury, New York: Kluwer/Plenum, (2005).

[38] C. A. Lipinski, Drug-Like Properties and the Causes of Poor Solubility and Poor Permeability, J. of Pharm. and Tox. Methods, 44, (2000), 235-249.

[39] Lipinski, C., Lombardo, F., Dominy, B., and Feeney, P, Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings, Adv. Drug Deliv. Rev, 46 (1-3), (2001) 3-26.

[40] D. F. Veber, S. R. Johnson, H. Cheng, B. R. Smith, K. W. Ward, K. D. Kopple, Molecular Properties that Influence the Oral Bioavailability of Drug Candidates, J. Med. Chem., 45 (12), (2002), 2615-2623.

[41] George A. Jeffrey, An Introduction to Hydrogen Bonding (Topics in Physical Chemistry), Oxford University Press, USA, (1997).

[42] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, Essential Cell Biology, Garland, New York, (1998).

[43] William L. Jorgensen, The Many Roles of Computation in Drug Discovery, Science 303, (2004), 1813-1817.

[44] Gisbert Schneider and Uli Fechner, Computer Based de Novo Design of Drug-Like Molecules, Nature Reviews, 4, (2005), 649-663.

[45] Hans-Joachim Bohm, The Computer Program LUDI: a New Method for the de Novo Design of Enzyme Inhibitors, J. Comp. Aid. Mol. Des., 6, (1992), 61-78.

[46] G. N. Ramachandran, C. Ramakrishnan, V. Sasisekharan, Stereochemistry of Polypeptide Chain Configurations, J. Mol. Biol, 7, (1963), 95.

[47] M. B. Eisen, D. C. Wiley, M. Karplus, R. E. Hubbard, HOOK: a Program for Finding Novel Molecular Architectures that Satisfy the Chemical and Steric Requirements of a Macromolecule Binding Site, J. Proteins, 3, (1994), 199-221.

[48] A. D. MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wio´rkiewicz-Kuczera, D. Yin, and M. Karplus, All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Protein, J. Phys. Chem., 102, (1998), 3586-3616.

[49] D. A. Pearlman, M. A. Murcko, CONCERTS: Dynamic Connection of Fragments as an Approach to de Novo Ligand Design, J. Med. Chem., 39, (1996), 1651-1663.

[50] A. Bugrim, T. Nikolskaya, Y. Nikolsky, Early Prediction of Drug Metabolism and Toxicity: Systems Biology Approach and Modeling. Drug Discov. Today, *9* (3), (2004), 127-135.

[51] C. Helma, In Silico Predictive Toxicology: The State of the Art and Strategies to Predict Human Health Effects, Curr. Opin. Drug Discov. Des., 8, (2005), 27-31.

[52] B. Hemmateenejad, R. Miri, M. Akhond, M. Shamsipur, QSAR Study of the Calcium Channel Antagonist Activity of Some Recently synthesized Dihydropyridine Derivatives: An Application of Genetic Algorithm for Variable Selection in MLR and PLS Methods, Chemom. Intell. Lab. Syst., 64, (2002), 91-99.

[53] Y. Takahata, M. C. A. Costa, A. C. Gaudio, Comparison Between Neural Networks (NN) and Principle Component Analysis (PCA): Structureactivity Relationships of 1,4-Dihydropyridine Calcium Channel Antagonists (Nifedipine Analogues, J. Chem. Inf. Comput. Sci., 43, (2003), 540-544.

[54] A. C. Gaudio, A. Korolkovas, Y. Takahata, Conformational Analysis of the 1,4-Dihydropyridines Linking the Structural Aspects to the Biological Binding Event: A Study of the Receptor-Site Conformation, J. Mol. Struct., 303, (1994), 255-263.

[55] Y. Takahata, M. C. A. Costa, A. C. Gaudio, A Comparative Study of Principal Component and Linear Multiple Regression Analysis in SAR and QSAR Applied to 1,4-Dihydropyridine Calcium Channel Antagonists (Nifedipine Analogues, J. Mol. Struct., 394, (1997), 291-300.

[56] K.-J. Schleifer, E. Tot, CoMFA, CoMSIA and GRID/GOLPE Studies on Calcium Entry Blocking 1,4-Dihydropyridines, Quant. Struct.-Act. Relat., 21, (2002), 239-248.

[57] X. Yao, H. Liu, R. Zhang, M. Liu, , Z. Hu, A. Panaye, J. P. Doucet, B. Fan, QSAR and Classification Study of 1,4-Dihydropyridine Calcium Channel Antagonists Based on Least Squares Support Vector Machines, Mol. Pharm., 2 (5), (2005), 348-356.

[58] H. Z. Si, T. Wang, K. J. Zhang, Z. D. Hu, B. Fan, QSAR study of 1,4-Dihydropyridine Calcium Channel Antagonists Based on Gene Expression Programming, Bioorg. Med. Chem., 14, (2006), 4834-4841.

[59] A. R. Katritzky, V. S. Lobanov, M. Karelson, Comprehensive Descriptors for Structural and Statistical Analysis, Reference Manual, Versions 2.0 and 2.13; University of Florida: Gainsville, FL, (1997).

[60] Marco Attene and Michela Spagnuolo, Automatic Surface Reconstruction from Point Sets in Space, in Eurographichs 2000, ed. M. Gross and F.R.A. Hopgood, 19 (3), (2000).

[61] D. T. Moustakas, P. T.Lang, S. Pegg, E. Pettersen, I. D. Kuntz, N. Brooijmans, R. C. Rizzo, Development and Validation of a Modular, Extensible Docking Program: DOCK 5, J. Comput. Aided Mol. Des., 20, (2006), 601–619.

[62] D. S. Goodsell, G. M. Morris, A. J. Olson, Automated Docking of Flexible Ligands: Applications of AutoDock, J. Mol. Rec., 9 (1), (1998), 1-5.

[63] Matthias Rarey, Bernd Kramer, Thomas Lengauer and Gerhard Klebe, A Fast Flexible Docking Method using an Incremental Construction Algorithm, Journal of Molecular Biology, 261 (3), (1996), 470-489.

[64] B. Chazelle, L. Palios, Decomposing the Boundary of a Nonconvex Polyhedron, J. Algorithmica, 17, (1997), 245-265.

[65] F. P. Preparata, S. J. Hong, Convex Hulls of Finite Sets of Points in Two and Three Dimensions, J. Commun. ACM, 20 (2), (1977) 87–93.

[66] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry, Algorithms and Applications, Springer, (2000).

[67] A. Armon, D. Graur and N. Ben-Tal, ConSurf: an Algorithmic Tool for the Identification of Functional Regions in Proteins by Surface Mapping of Phylogenetic Information, Journal of Molecular Biology, 307 (1), (2001), 447-463.

[68] MATLAB, The MathWorks, Inc., (1984-2008).

[69] C. D. Mol, D. R. Dougan, T. R. Schneider, R. J. Skene, M. L. Kraus, D. N. Scheibe, G. P. Snell, H. Zou, B. C. Sang, K. P. Wilson, Structural Basis for the Autoinhibition and STI-571 Inhibition of c-Kit Tyrosine Kinase, J.Biol.Chem., 279, (2004), 31655-31663.

[70] V. Cody, D. Chan, N. Galitsky, D. Rak, J. R. Luft, W. Pangborn, S. F. Queener, C. A. Laughton, M. F. Stevens, Crystal Structure and Molecular Modeling Studies on the Pneumocystis Carinii Dihydrofolate Reductase Cofactor Complex with TAB, a Highly Selective Antifolate, J. Biochemistry, 39, (2000), 3556-3564.

[71] C. M. Silva, R. Reid, Gastrointestinal Stromal Tumors (GIST): C-kit Mutations, CD117 Expression, Differential Diagnosis and Targeted Cancer Therapy with Imatinib, J. Pathol Oncol Res., 9(1), (2003), 13-19.

[72] T. Langer, G. Wolber, Virtual Combinatorial Chemistry and in Silico Screening: Efficient Tools for Lead Structure Discovery, J. Pure Appl. Chem., 76 (5), (2004), 991–996.

[73] S. Makino, T. J. A. Ewing, I. D. Kuntz, DREAM++: Flexible Docking Program for Virtual Combinatorial Libraries, J. of Comp.-Aided Mol. Design, 13, (1999), 513–532.

[74] I. E. Grossman, Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques, J. Optimization and Engineering, 3, (2002), 227–252.

[75]

[76] Y. Cheng, W. H. Prusoff, Relationship Between the Inhibition Constant (K1) and the Concentration of Inhibitor which Causes 50 Percent Inhibition (I_{50}) of an Enzymatic Reaction, J. Biochem Pharmacol., 22 (23), (1973), 3099-3108.

[77] HyperChem. 7.5, Hypercube, (2003)

[78] A. R. Katritzky, V. S. Lobanov, M. Karelson, Comprehensive Descriptors for Structural and Statistical Analysis, Reference Manual, Versions 2.0 and 2.13, University of Florida: Gainsville, FL, (1997).

[79] Garthwaite, P. H. An interpretation of partial least squares. J. Am. Stat. Assoc., 89 (425), (1994), 122-127.

[80] MINITAB Statistical Software, Release 14 for Windows; MINITAB Inc.: State College, PA, (2003).

[81] WEKA 3: Data Mining Software in JaVa; The University of Waikato: Hamilton, New Zealand, (2005).

[82] J. L. Devore, Probability and Statistics for Engineering and the Sciences, 5th Edition, Duxbury, CA, (2000).

[83] V. N. Viswanadhan, G. A.Mueller, S. C. Basak, J. N. Weinstein, Comparison of a Neural Net Based QSAR Algorithm (PCANN) with Hologram and Multiple Linear Regression-Based QSAR Approaches: Application to 1,4-Dihydropyridine-Based Calcium Channel Antagonists. J. Chem. Inf. Comput. Sci., 41, (2001), 505-511.

[84] B. Widrow, M. A. Lehr, 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation, Proc. IEEE, 78 (9), (1990), 1415-1442.

[85] J. Friedman, T. Hestie, R. Tibshirani, Additive Logistic Regression: A Statistical View of Boosting, Ann. Stat., 28 (2), (2000), 337, 407.

[86] D. Heckerman, A Tutorial on Learning with Bayesian Network; Technical Report; Microsoft Research Advanced Technology Division, Microsoft Corporation: Redmond, WA, (1996).

[87] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd Ed., Morgan Kaufmann: San Francisco, (2005).

[88] G. C. John, E. T. Leonard, K*: An Instance-Based Learner Using an Entropic Distance Measure, In Proceedings of the 12th International Conference on Machine LearninG, Morgan Kauffman: San Francisco, (2001), 108-114.

[89] W. S. Cleveland, S. J. Delvin, Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting, J. Am. Stat. Assoc., 83 (403), (1988), 596-610.

[90] A. J. Scott, M. J. Symons, Clustering Methods Based on Likelihood Ratio Criteria, Biometrics, 27, (1991), 387-397.

[91] J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization. In Advances in Kernel Methods Support Vector Learning, B. Scholkopf, C. Burges, A. Smola Eds.; MIT Press: Cambridge, MA, (1999), 185-208.

[92] D. M. J. Tax, R. P. W. Duin, Using Two-Class Classifiers for Multiclass Classification. In 16th International Conference on Pattern Recognition (ICPR'02), International Association for Pattern Recognition (IAPR): Durham, NC, 2, (2002), 20124.

[93] Y. Qu, B. L. Adam, Y. Yasui, M. D. Ward, L. H. Cazares, P. F. Schellhammer, O. J. Semmes, Boosted Decision Tree Analysis of Surface-Enhanced Laser Desorption /Ionization Mass Spectral Serum Profiles Discriminates Prostate Cancer from Noncancer Patients, J. Clin. Chem., 48, (2002), 1835-1843.

[94] V. Cody, D. Chan, N. Galitsky, D. Rak, J. R. Luft, W. Pangborn, S. F. Queener, C. A. Laughton, and M. F. G. Stevens, Structural Studies on Bioactive Compounds. 30. Crystal Structure and Molecular Modeling Studies on the Pneumocystis carinii Dihydrofolate Reductase Cofactor Complex with TAB, a Highly Selective Antifolate, J. Biochemistry, 39 (13), (2000), 3556 -3564.

[95] D. Benarroch, P. Smith, S. Shuman, Characterization of a Trifunctional Mimivirus mRNA Capping Enzyme and Crystal Structure of the RNA Triphosphatase Domain, J. Structure, 16 (4), (2008), 501-512.

[96] D. M. Haapalainen, G. Merilainen, J. E. Jalonen, P. Pirila, R. K. Wierenga, J. K. Hiltunen, T. Glumoff, Crystal Structure of the Liganded SCP-2-Like Domain of Human Peroxisomal Multifunctional Enzyme Type 2 at 1.75 A resolution, J.Mol.Biol., 313, (2001) 1127-1138 .

[97] *http://www.pumma.nl/index.php/Theory/Potentials*