



COMPUTER-AIDED DRUG DESIGN Lead Discovery

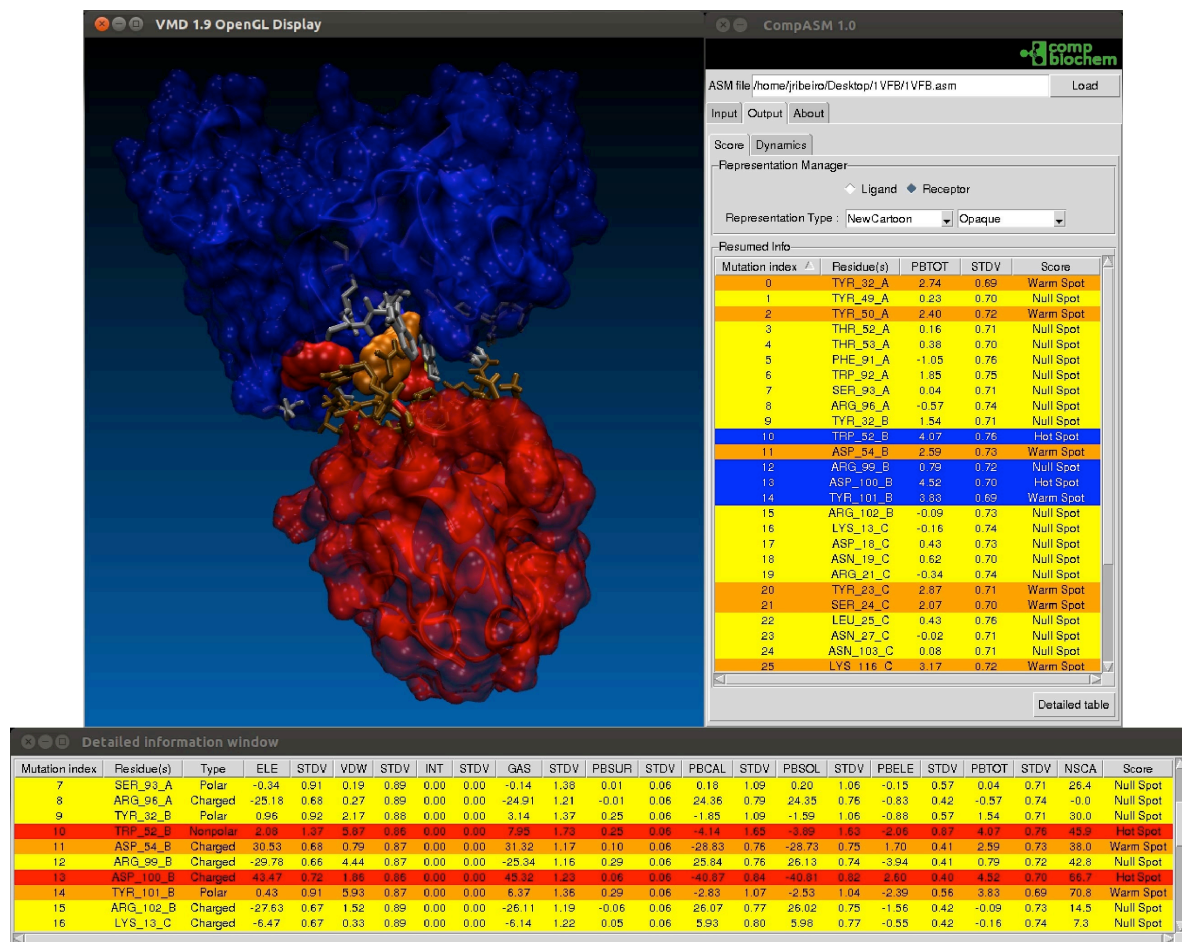
João Rui Vieira Ribeiro

Tese de Doutoramento apresentada à

Faculdade de Ciências da Universidade do Porto

Química

2014



COMPUTER-AIDED DRUG DESIGN Lead Discovery

João Rui Vieira Ribeiro
Programa Doutoral em Química Sustentável
Departamento de Química e Bioquímica
2014

Orientador

Maria João Ribeiro Nunes Ramos, Professor Catedrático,
Faculdade de Ciências, Universidade do Porto

Acknowledgments

I would like to show my deepest gratitude to my supervisor, Professor Maria João Ramos, for the opportunity to develop this project and for giving me support, knowledge and guidance through these years. In the same spirit I would like to acknowledge Professor Pedro Fernandes for the solid support and advice, making each project move forward to the proper end.

I thank to Professor Klaus Shulten and John Stone from Theoretical and Computational Biophysics Group for receiving me for the 3 months work collaboration. I had the possibility to learn and develop my programming skills inside a high tech level environment only available in these kinds of groups. The experience was rewarding in the scientific level as well as in personal level.

I also thank to Nuno Cerqueira for the challenging work relationship that lead each software to a higher level of detail and functionality. Each brainstorm retrieved significant conclusions to be applied in each program. I also thank Irina for the practical guidance in our collaboration.

I would also like to thank to all members of the Theoretical and Computational Biochemistry Research Group that made my PhD student life pleasant. I have special gratitude to Zé, “Professora” Natércia, Daniel, Xana and Silvia for the patience shown in several episodes. I also thank Sérgio for the scientific discussions and advices. In the informatics universe I would like to thank to Oscar for not have blocked the entry of his office to me (I think I deserved it some times) and for the very instructive conversations. I would like to let my appreciation for the fellowship of Diana, Diogo, Gaspar, Marta, João Coimbra, Nini and those that I am inexcusable for not remember their names.

I cannot finish my acknowledgments without thank my parents for their support through all these years. Without you I would never get so far. And off course, I thank to Carla for the unconditional love and companionship given since we met.

This PhD had the financial support of FCT through the doctoral scholarship SFRH/BD/61324/2009.

Para a minha Família

Abstract

The present work focuses on the development of new bioinformatics tools to assist the user in a Computational Chemistry/Theoretical Chemistry laboratory, improving the Computer Aided Drug Design process. Often the use of already existing software requires a slow learning process, where simple tasks can reveal tricky to the non-expert users. As a consequence, interfaces were developed here too that make simple the task of the user.

An outline on Computer Aided Drug Design is given in the first chapter, highlighting the drug development cost and listing the subjects proposed to be developed and improved. In the second chapter, the theory behind the present work is described, focusing on the theory on which the pre-existing used software are based.

The remaining chapters refer to the developed work in these last four years, both already published and unpublished, being each chapter devoted to each developed software. The subjects covered in each chapter are: virtual screening and the developed software vsLab; chemical motifs inside protein structures and the software Chem-Path-Tracker; structural surfaces and volumes calculated by the software VolArea; and last the Computational Alanine Scanning Mutagenesis technique to study protein-protein interactions performed by the software CompASM.

All developed software are plug-ins of the world wide used molecular visualizer, Visual Molecular Dynamics (VMD). This association revealed very interesting and useful because it was possible to provide the software with the visual dimension, thus complementing the numerical results returned by the developed tools. In this way it is given to the user the possibility to inspect the results visually, which is crucial in most of the times to improve the quality of the conclusions to be retrieved.

Resumo

O trabalho presente nesta tese consiste no desenvolvimento de novas ferramentas bioinformáticas que visam auxiliar o utilizador num laboratório de Química Computacional/Química Teórica melhorando, assim, o processo de *design* de fármacos baseadas em estudos computacionais. A necessidade deste auxílio revela-se importante aquando da utilização de vários programas informáticos afetos a esta área. Por vezes a simples utilização de um programa informático exige uma curva de aprendizagem lenta, onde nem sempre a mais trivial operação é de fácil execução para os utilizadores mais inexperientes.

No primeiro capítulo é dada uma visão geral da temática do *design* de fármacos por meios computacionais, realçando a problemática dos custos associados a este tema, apresentando os pontos propostos a serem desenvolvidos e melhorados. No segundo capítulo é descrita a teoria que serve de base às ferramentas informáticas já existentes que foram utilizadas para o desenvolvimento deste trabalho.

Os restantes capítulos são referentes ao trabalho desenvolvido no decorrer destes últimos quatro anos, publicados ou por publicar, sendo cada capítulo afeto a uma ferramenta informática. Os temas referentes a cada capítulo são: *virtual screening* e a ferramenta desenvolvida *vsLab*; padrões químicos contidos em estruturas proteicas e o programa *Chem-Path-Tracker*; cálculo de superfícies e volumes de estruturas realizado pelo programa *VolArea*; e por último a técnica computacional de mutagénese por alaninas para o estudo de interações proteína-proteína realizada pelo programa *CompASM*.

Todos os programas foram embutidos num programa de visualização molecular mundialmente utilizado denominado *Visual Molecular Visualizer* (VMD). Esta associação revelou-se bastante interessante e útil, pois foi possível dotar os programas da dimensão visual, complementando assim os resultados numéricos originados pelas ferramentas desenvolvidas. Assim é dada a possibilidade ao utilizador de inspecionar visualmente os resultados, muitas vezes crucial para melhorar a qualidade das conclusões a serem retiradas.

Résumé

Le présent travail se concentre sur le développement de nouveaux outils bioinformatiques pour aider l'utilisateur dans un laboratoire informatique Chimie / Chimie théorique, l'amélioration de la Assistée par Ordinateur processus Drug Design. Souvent, l'utilisation d'un logiciel déjà existant nécessite un processus d'apprentissage lent, où les tâches simples peuvent révéler difficile pour les utilisateurs non - experts. En conséquence, les interfaces ont été développés qui font ici aussi simple, la tâche de l'utilisateur.

Un aperçu sur Conception Assistée par Ordinateur drogue est donnée dans le premier chapitre, en soulignant le coût de développement de médicaments et la liste des sujets proposés pour être développé et amélioré. Dans le deuxième chapitre, la théorie derrière ce travail est décrit, en se concentrant sur la théorie sur laquelle sont basées les logiciels utilisés pré-existant.

Les chapitres se réfèrent au travail développé dans ces quatre dernières années, tous deux déjà publiés et non publiés, étant chaque chapitre consacré à chaque logiciel développé. Les sujets abordés dans chaque chapitre sont : le criblage virtuel et le vsLab logiciel développé ; motifs chimiques à l'intérieur des structures de protéines et le logiciel Chem -Path -Tracker , les surfaces et les volumes calculés par le VolArea logiciel structurelles et durent la technique informatique Alanine Scanning mutagenèse à étudier interactions protéine-protéine exécutées par le logiciel CompASM .

Tous les logiciels développés sont les plug-ins du monde de visualiseur large utilisé moléculaire, Visual Molecular Dynamics (VMD). Cette association a révélé très intéressant et utile, car il était possible de fournir le logiciel avec la dimension visuelle, complétant ainsi les résultats numériques retournées par les outils développés. De cette manière, il est donné à l'utilisateur la possibilité d'inspecter visuellement les résultats, ce qui est essentiel dans la plupart des fois d'améliorer la qualité des conclusions à récupérer.

Keywords

▪ 2C	Two processing cores
▪ 4C	Four processing cores
▪ ACD	Available Chemical Directory
▪ ADT	AutoDockTools
▪ AMDE/Tox	Absorption, Metabolism, Excretion and Toxicity
▪ ASM	Alanine Scanning Mutagenesis
▪ AVG	Average
▪ BDT	Automatic Protein-Ligand Docking for Everyone
▪ CADD	Computer-Aided Drug Design
▪ CCC	CUDA Compute Capability
▪ CMB	Carbohydrate-Binding Modules
▪ CompASM	Computational Alanine Scanning Mutagenesis
▪ CPT	Chem-Path-Tracker
▪ CUDA	Compute Unified Device Architecture
▪ DFT	Density Functional Theory
▪ FEP	Free-Energy Perturbation
▪ FN	False Negative
▪ FP	False Positive
▪ GA	Genetic Algorithm
▪ GPU	Graphics Processing Units
▪ GUI	Graphical user Interface
▪ HBonds	Hydrogen Bonds
▪ HSPP	Homology-derived Secondary Structure of Proteins
▪ LGA	Lamarckian Genetic Algorithm
▪ logP	Partition Coefficient
▪ LS	Local Search
▪ MADAMM	Multi Staged Docking with an Automated Molecular Modeling
▪ MD	Molecular Dynamics
▪ MM	Molecular mechanics
▪ MMPBSA	Molecular Mechanics/Poisson-Boltzmann Surface Area
▪ NCI	National Cancer Institute
▪ NMR	Nuclear Magnetic Resonance

▪ NSCA	Non-Solvent Contact Area
▪ OSC	Oxidosqualene Cyclase
▪ PDB	Protein Data Bank
▪ PME	Particle Mesh Ewald
▪ PT	Points
▪ RESP	Restrained ElectroStatic Potential
▪ RMSD	Root Mean Square Deviation
▪ RNR	Ribonucleotide Reductase
▪ SASA	Solvent Accessible Surface Area
▪ SM	Streaming Multiprocessor
▪ SP	Streaming Processors
▪ TCL	Tool Command Language
▪ TI	Thermodynamics Integration
▪ TN	True Negative
▪ TP	True Positive
▪ vdW	van der Walls
▪ VMD	Visual Molecular Dynamics
▪ vsLab	Virtual Screening Lab

Index

1. Introduction	22
2. Methods	25
2.1. Molecular Mechanics	25
2.1.1. Introduction	25
2.1.2. The Force Field Energy	26
2.1.2.1. The Stretching Energy	27
2.1.2.2. The Bending Energy	28
2.1.2.3. The Torsional Energy	29
2.1.2.4. The Van Der Waals Energy	30
2.1.2.5. The Electrostatic Energy	32
2.1.2.6. The Force Field Parameterization	32
2.1.3. Molecular Dynamics	33
2.1.3.1. Simulation Time Step	34
2.1.3.2. Simulation Time Scale	34
2.1.3.3. Boundary Conditions	35
2.1.4. Predicting the Association Energy	35
2.1.4.1. Molecular Dynamics/Poisson-Boltzmann Surface Area	36
2.1.5. Molecular Docking	38
2.1.5.1. Search Algorithm	38
2.1.5.2. Scoring Function	39
3. Results and Discussion	41
3.1. Virtual Screening Lab (vsLab)	43
3.1.1. VsLab - An implementation for virtual high-throughput screening using AutoDock and VMD	45
3.1.1.1. Abstract	45
3.1.1.2. Introduction	45
3.1.1.3. AutoDock	47
3.1.1.4. Graphical User Interface	48
3.1.1.5. Conclusions	51
3.2. Chem-Path-Tracker	52
3.2.1. Chem-Path-Tracker - An automated tool to analyze chemical motifs in molecular structures	54
3.2.1.1. Abstract	54
3.2.1.2. Introduction	54
3.2.1.3. Software Description	56
3.2.1.3.1. Graphical Interface	56

3.2.1.3.2.	A modified Dijkstra's algorithm.....	57
3.2.1.3.3.	The output of Chem-Path-Tracker	60
3.2.1.4.	Software Validation	62
3.2.1.4.1.	Chemical motifs surrounding active sites.....	62
3.2.1.4.2.	Guanine deaminase.....	63
3.2.1.4.3.	Oxidosqualene cyclase	65
3.2.1.4.4.	Identification of Cation- π interactions	67
3.2.1.4.5.	Exploring Water Channels	68
3.2.1.4.6.	Proton and electron transfers.....	71
3.2.1.5.	Conclusions	74
3.3.	VolArea	76
3.3.1.	VolArea - A Bioinformatics tool to calculate the surface area and the volume of molecular systems.....	78
3.3.1.1.	Abstract.....	78
3.3.1.2.	Introduction	78
3.3.1.3.	Methods	80
3.3.1.3.1.	The Surface Area Algorithm.....	81
3.3.1.3.2.	The Volume Algorithm	82
3.3.1.4.	RESULTS AND DISCUSSION	86
3.3.1.4.1.	Validation	86
3.3.1.4.1.1.	Graphical Interface.....	89
3.3.1.4.2.	Applications.....	91
3.3.1.4.2.1.	Static structures assay.....	92
3.3.1.4.2.2.	Multi-framed structures assay.....	94
3.3.1.5.	Conclusion	96
3.4.	VolArea and CUDA.....	98
3.4.1.1.	Data Structure.....	100
3.4.1.2.	Results	102
3.4.1.3.	Conclusion	103
3.5.	CompASM	104
3.6.	CompASM - an Amber-VMD Alanine Scanning Mutagenesis plug-in.	106
3.6.1.	Abstract.....	106
3.6.2.	Introduction	106
3.6.3.	Methodology	107
3.6.4.	Software Description.....	109
3.6.5.	Results	111
3.6.6.	Conclusions	115
4.	General Conclusions	116

Index of figures

Figure 1- Terms of the Force Field Energy function.	26
Figure 2- The stretching energy of the C-H bond of the CH ₄ molecule..	28
Figure 3- The bending energy of the H-C-H angle of the CH ₄ molecule.....	29
Figure 4- Illustration of the torsional angle definition.	30
Figure 5: Thermodynamic cycle used to calculate the binding free energy.....	36
Figure 6: Graphical user interfaces available in vsLab	49
Figure 7: Sample VMD sessions displaying the results obtained from the vsLab plug-in.	50
Figure 8: Chem-Path-Tracker's Graphical User Interface (GUI) input tab.....	57
Figure 9: General algorithm implemented in Chem-Path-Tracker software.	60
Figure 10: Chem-Path-Tracker's Graphical User Interface (GUI) output tab..	61
Figure 11: Chemical interaction between the residues of the active site of the enzyme guanine deaminase that were highlighted by the Chem-Path-Tracker software.....	64
Figure 12: Most important chemical interactions present in the active site of OSC.	66
Figure 13: Graphical pathway that was highlighted by Chem-Path-Tracker revealing the (cation- π)- π stack chain found on the PDB structure with the code 3HHR.	68
Figure 14: Graphical pathway that was highlighted by Chem-Path-Tracker on the PDB structure with the code 2zz9.....	70
Figure 15: Interaction network between the residues of monomer R2 of RNR provided by Chem-Path-Tracker (for the clarity of the image, Tryptophan 48 and 211 were not represented).....	72
Figure 16: Illustration of the usage of the probe radius to calculate the surface of molecules.	81
Figure 17: Schematic illustration of the searching radius superposition in order to illustrate the volume algorithm.	83
Figure 18: Example of the searching radius superposition, here applied to the cavity volume calculation.....	84
Figure 19: Diagram of the algorithm that is used to calculate the volume in VolArea.	85
Figure 20: a) The relative error in the protein volume calculation (defined as the difference between the experimental and calculated volumes divided by the experimental volume) vs. the scale value and b) the average and standard deviation of the relative error for the 11 structures calculated..	86
Figure 21: Variation of deviation as a function of the protein volume with a scale of 0.7 Å.	87
Figure 22: a) Average of the times of the protein volume calculation (using scale values from 1.0 to 0.7 Å) using four processing cores (4C) as a function of experimental volume, b)	

ratio of the times in the volume calculation using two (2C) and four processing (4C) cores. The speed up is linear.	88
Figure 23: VolArea Graphical interface:	90
Figure 24: Surface values obtained with VolArea while analyzing the interface region of two proteins (retrieved from the PDB structure 1VFB):	93
Figure 25: A) Exposed surface area of the residues located at the protein interface that interact more closely with the ligand..	95
Figure 26: Simple representation of the NVIDIA Fermi GPU architecture. “A simplified hardware block diagram for the NVIDIA representing the arithmetic units “streaming processors” (SP) and “special function units” (SFU) for computing especial algebraic functions	99
Figure 27: Schematization of the organization of the atoms in bins	101
Figure 28: Chart displaying the gains retrieved from GPU	102
Figure 29: General algorithm of the CompASM procedure 27.	108
Figure 30: Scheme representing the Non-Solvent Contact Area (NSCA)	110
Figure 31: Results of the protein-protein interface study of immunoglobulin complexed with an egg lysozyme (detailed results in supporting information).	111

Index of Tables

Table I - Summary of the software's available for calculating the volume and the surface of structures.....	80
Table II - Results originated by CompASM. The Non-Solvent Contacting Area (NSCA) is calculated by the CompASM program.....	113
Table III - Comparison of the values obtained using CompASM with those that result from the software/servers available.....	114

1. Introduction

The discovery of new therapeutic compounds and the improvement of existing drugs are crucial aspects for every modern society, especially for rational drug design where the time and the costs of the process must be controlled and environmental harmless procedures must be taken. Despite some divergences in terms of the real cost of drug design¹⁻⁴, some authors suggested that it is necessary more than 8 years and millions of dollars (between 800 and 2000 million US dollars) to develop a new therapeutic compound. These predictions are based in the probability of the new drug successfully pass each clinical trial phase. Here, the primordial steps of the drug design are extremely important to improve these probabilities, mainly when computational means are efficiently used.

Independently of the chosen methodology to develop a new drug, crucial steps are required, such as the identification of hit compounds and improvement of the lead compounds; the correct study and description of the drug target, and the prediction of the effects of the drug in terms of absorption, metabolism, excretion and toxicity (AMDE/Tox). Despite the possibility of performing these steps *in vitro*, it is using computational tools where major gains are obtained.

Computer-Aided Drug Design (CADD) is a field of research that comprehends a vast collection of computational solutions to store, manage, analyze and model chemical compounds. Here, the main purpose is to simulate ligand-receptor complexes in biological conditions in order to predict chemical properties to anticipate and manipulate drugs functionality and behavior. One of the greatest advantages of a CADD campaign is the possibility to select or model the potential compounds; calculate the drug-receptor binding properties and optimize compounds *in silico*, avoiding the usage and production of harmful substances. Only then, the most promising compounds are experimentally synthesized and tested improving the success and speed, i.e. reducing the costs of drug discovery.

The bases of the new potential drug commonly derive from one or more databases containing a massive number of molecules, which is the case of the ZINC database⁵ (despite the name it is not a Zinc based compound database). This free repository in particular contains about 21 million compounds prepared for virtual screening in their biologically relevant forms. Besides the atoms' 3D-coordinates, this database contains the values that describe the residues protonation states, molecular weights, calculated LogP and the rotatable bonds, making ZINC database an important source for CADD campaigns. Beside this repository, there are many other repositories containing a large number of structures as well, which is the case of Available Chemical Directory (ACD, over 7 million compounds)⁶, National Cancer Institute compound database (NCI, over 260 000

compounds)⁷ among others⁸⁻¹⁰, in which some of them unfortunately are not so completed in terms of biological significant values.

One keystone of the drug design process is the full characterization of the target structure, even before the selection of the possible candidates from the previous mentioned databases. Generally, these structures are obtained from experimental means and stored in the Protein Data Bank (PDB)^{11,12}. This database is composed by the structures of large biological molecules, such as proteins and nucleic acids, mainly resolved by X-Ray Crystallography/Diffraction and Nuclear Magnetic Resonance (NMR). However, when the experimental 3D-structure is not available, the most reliable technique to use is Homology Modelling¹³⁻¹⁵ or to extract the structure from the homology based, Homology-derived Secondary Structure of Proteins(HSPP) database^{16,17}.

The availability of such amount of information is a vital aspect for computational chemists allowing the application of CADD techniques in a wide variety of biological processes, using appropriate and successfully proven methodologies such as molecular docking¹⁸ and virtual screening¹⁹. CADD methodologies have been supported by the appearance of numerous scientific software focused on drug design or molecular simulations, serving different purposes, presenting more or less accurate results, freely available or purchasable²⁰⁻²⁴. The development of efficient parallel algorithms has boosted CADD even further, and lately the usage of Graphics Processing Units (GPU) in massive parallel operations²⁵ has resulted in significant gains in terms of speed and allowed size of the simulated systems. On the other hand, large collections of structures and software specialization can also bring difficulties and challenges to the researcher in his/her projects. Here, bioinformatics play an important role creating tools to complement existing software, turning all this information usable and manageable. Using these tools is possible to extract significant and relevant values, improving in this way the efficiency of CADD campaigns.

Examples of such demand for process optimization in CADD methods is felt, for instance, when it is necessary to evaluate the binding of a large set of compounds extracted from one of the previously mentioned databases. It is thus necessary to analyze the poses of those structures against the target's binding site and rank the ligands regarding the strength of binding interactions or other parameters. The information required to perform this process would be impossible to handle manually, and even computationally users would have to be familiar with programming languages such as Python or Tool Command Language (TCL). For instance, at the end of a protein-ligand docking calculation, the total number of files originated by the software can easily double or quadruple the number of structures files (e.g. using Autodock²⁶), making the extraction of the information of interest laborious and hardworking. The same exponential growth of generated information can occur when another CADD technique is applied, namely Computational Alanine Scanning Mutagenesis

(CompASM)²⁷. This method analyzes protein-protein interfaces, identifying the residues that contribute more for the binding of the intervening systems. CompASM is based on the premise that residues responsible for the binding of the proteins (hotspots) cause significant variations in the binding energy of the complex when mutated by an alanine residue²⁸. Moreira *et al.* proposed a successful protocol for a computational application of ASM which comprehends Molecular Dynamics (MD) simulations and Molecular Mechanics/Poisson-Boltzmann Surface Area (MMPBSA)²⁹ calculations.

Despite the major improvements in protein-ligand docking, virtual screening and ASM methods, several developments are required in order to bring this analysis to general use and to non-expert users. These new tools must have the ability to assist the user in the generation and parameterization of the input data and assemble the final results, displaying them in a user-friendly manner.

The work presented in this thesis provides solutions to assist computational chemists in their daily work, developing new tools to simplify the application of Molecular Docking, Virtual Screening, and Computational Alanine Scanning Mutagenesis, applying world-wide used software such as AutoDock²⁶, and the Amber molecular simulation package³⁰. In this chapter new algorithms are described to calculate protein structural features, namely surface residues in contact with solvent or other compounds, as well as to calculate the volume of any structure and empty spaces such as cavities and clefts. It is also presented a bioinformatics tool to identify and track chemical motifs (hydrogen bonds, cation- π -interactions, proton-electron transfer pathways and water tunnels) throughout a protein/molecular system. In order to provide a visual aspect to the user, allowing the inspection of the results, the developed tools are compatible with the widely used molecular visualizer, Visual Molecular Dynamics (VMD)³¹.

2. Methods

This section intends to give an overview of the theoretical methods applied in the developed work, in order to provide the foundations of the software used in the presenting new bioinformatics tools. The covered subjects are: Molecular Mechanics, Molecular Dynamics highlighting the specific binding energy calculation method Molecular Mechanics/Poisson-Boltzmann Surface Area (MMPBSA); and Molecular Docking, in particular Protein-Ligand Docking and Virtual Screening. In each topic, the theoretical contents associated to each subject is described numbering the most used software. A full method description and deeper understanding of computational chemistry is presented in ³².

2.1. Molecular Mechanics

2.1.1. Introduction

Molecular mechanics (MM) is a simplification of the molecular structure model to an aggregated complex of “balls and springs”, where the unitary particle is the atom, neglecting electrons and protons as individual particles. In MM methods, the information assigned to each particle (atom) is the atomic mass, charge, van der Waals radii and instead of calculating the bonding information as a product of the Schrödinger equation, the values for bond lengths, bond angles and dihedral angles are provided explicitly as parameters of the force field. Since the first published paper applying molecular mechanics calculations³³, several works have been published using this method with very interesting and promising results^{27,34-38}. The application of this method to biomolecules has got great acceptance mainly because of the relative good performances in terms of speed and final results. Other important factor that contributed to the generalization usage of these methods were the increase of the community users that have applied these tools through these last decades, making somewhat easy to obtain the parameters for a wide range of structures type. At the same time, the requirement for these parameters is one of the drawbacks of this methodology, which are not always available, requiring its calculation. This method also fails when a deeper understanding of the system is intended, for instance, the study of formation and breaking of the chemical bonds.

In spite of the negative aspects, MM methods are still a very powerful strategy for molecular simulations and drug design. For instance, it is now possible to simulate 1 million atoms model of the satellite tobacco mosaic virus with a simulated time of 50 nanoseconds (ns) ³⁹. The application of these methodologies is quite straightforward while the type of

atoms in the system remains common or is possible to obtain the parameters from other user calculations. In both cases, MM packages such as AMBER⁴⁰, CHARMM⁴¹, GROMACS⁴² incorporate a large source of parameters for a wide range of biological molecules, such as amino acids and nucleotides, and have a large community of users. This last factor increases the probabilities of finding some missing parameters, allowing the simulation of a wide range types of systems.

2.1.2. The Force Field Energy

Molecular Mechanics describes the force field energy (E_{FF}) as the sum of the terms describing the energy required to distort a molecule in a specific way:

$$E_{FF} = E_{str} + E_{bend} + E_{tor} + E_{vdw} + E_{ele} + E_{cross} \quad (1)$$

This equation is composed by energy functions that deal with bond interactions, non-bound interactions and cross-terms, being the first three terms the *stretching* energy (E_{str}), the *bending* energy (E_{bend}) and the *torsional* energy (E_{tor}) functions to describe the interactions between bonded atoms; the following two are the van der Waals (E_{vdw}) and electrostatic energy (E_{ele}) functions to describe the unbound interactions, and the last term of the equation is the combination of the any of the previous terms.

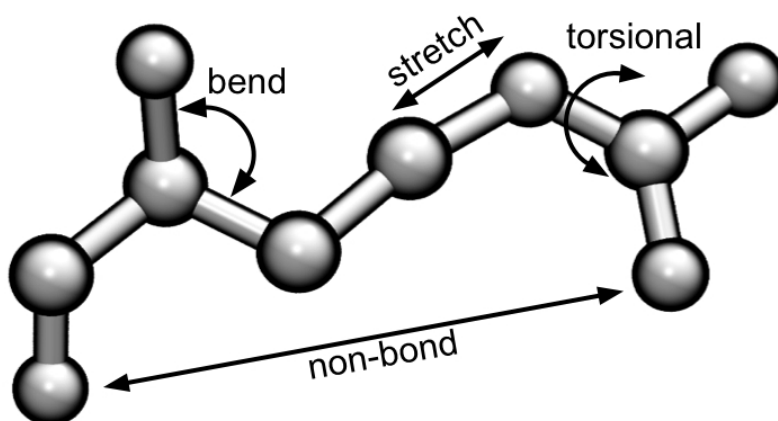


Figure 1- Terms of the Force Field Energy function.

The differences in the composition and complexity of the equations adopted to calculate these energies are mainly determined by the problem to be solved by the different force fields. For instance, the AMBER force field aims at dealing with biological systems, which can easily turn very computationally demanding due to the size of the complexes. In this

case simplified functions are required, describing only the bond and non-bond terms, dismissing the description of the out-of-the-plane bending energy and the cross-term. All molecular simulations presented in this work were carried out using the AMBER force field, and only the terms contained in this force field will be described in the following sections.

2.1.2.1. The Stretching Energy

E_{str} is the function of the energy associated to the elongation of the bond between two atoms (A and B) around the equilibrium bond length. A simple way to calculate this value is by the harmonic potential presented in equation (2).

$$E_{str}(R^{AB}) = K^{AB}(R^{AB} - R_0^{AB})^2 = K^{AB}(\Delta R^{AB})^2 \quad (2)$$

In equation (2) K^{AB} defines the bond force constant and R_0^{AB} defines the equilibrium bond length between atoms A and B. Despite the good results obtained by this potential in equilibrium geometries, a more accurate solution is required to analyze long-range bond distance or when it is necessary to include additional energy terms such as vibration energies.

The Taylor expansion of the harmonic potential up to the fourth term in equation (3) is a better energy descriptor when it is necessary to analyze these energies beyond the equilibrium bond length.

$$E_{str}(R^{AB}) = K_2^{AB}(\Delta R^{AB})^2 + K_3^{AB}(\Delta R^{AB})^3 + K_4^{AB}(\Delta R^{AB})^4 \quad (3)$$

In spite of the better result, this expansion still presents two major drawbacks: the first one is the requirement of more parameters to be assigned, and the second one is the tendency to infinity ($+\infty$) in long bond lengths, instead of a tendency to a constant energy, the dissociation energy.

The most accurate method to calculate the stretching energy is through the Morse potential (4), unfortunately this method is also the most slow/computationally demanding method.

$$E_{Morse}(\Delta R) = D(1 - e^{-\alpha\Delta R})^2$$
$$\alpha = \sqrt{\frac{K}{2D}} \quad (4)$$

Like the previous potential, this method also requires some parameters such as the dissociation energy D and the force constant at the equilibrium distance K .

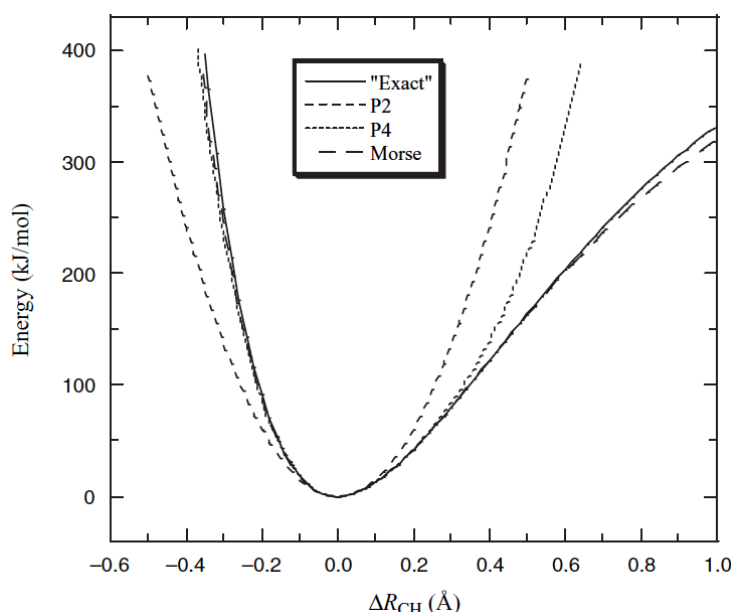


Figure 2- The stretching energy of the C-H bond of the CH₄ molecule. The exact curve was based on an electronic structure calculation with CASSCF/6-311++G(2df,2dp), the P2 and P4 curve stands for the simple harmonic potential and the Taylor Expansion of the harmonic potential, respectively ³².

At ambient or biological temperature, the bond length usually presents a variation of 0.03Å and the curve region of interest for simulation purposes is the bottom energy ~40 kJ/mol. In this region, all methods present close results, differing only in the computational and parameters requirements. In this case, the simple harmonic potential does not introduce significant errors, being profitable mainly by the calculation speed. The force field used in this work, AMBER force field, implemented the harmonic potential as the elected method to calculate stretching energies.

2.1.2.2. The Bending Energy

The E_{bend} energy function describes the energy necessary to change the angle formed by three atoms A-B-C, centered on atom B. Just like in the previous term, the bending energy is efficiently described by the simple harmonic potential (5). This potential formulation also requires the parameterization of the force constant K^{ABC} and instead of the equilibrium bond length R_0^{AB} , here it is required the value of the equilibrium bond angle θ_0^{ABC} .

$$E_{str}(\theta^{ABC}) = K^{ABC}(\theta^{ABC} - \theta_0^{ABC})^2 = K^{ABC}(\Delta\theta^{ABC})^2 \quad (5)$$

If a more accurate solution is required, the third term of the Taylor expansion returns a reasonable value, which is not normally necessary as it is possible to see in Figure 3.

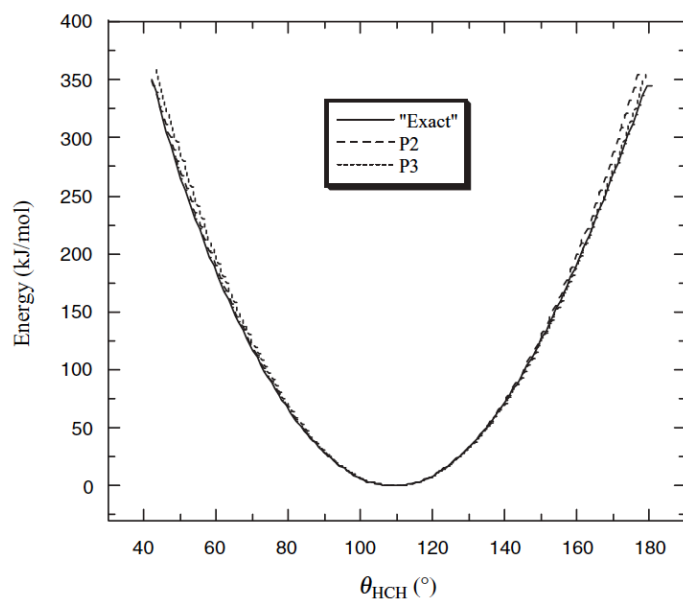


Figure 3- The bending energy of the H-C-H angle of the CH₄ molecule. The exact curve was based on an electronic structure calculation with (MP2/aug-cc-pVTZ), the P2 and P3 curve stands for the simple harmonic potential and the Taylor Expansion of the harmonic potential, respectively³².

As mentioned previously, at the bottom 40 kJ/mol energy of the curve (equivalent to most hydrogen bond energies), both methods fit reasonably well to the energy curve, being the simple harmonic potential more efficiently computed mainly due to the requirement of only two parameters.

2.1.2.3. The Torsional Energy

E_{tor} (torsional energy) is the energy necessary to rotate atoms A or D around bond B-C in a sequence of bonded atoms A-B-C-D (Figure 4).

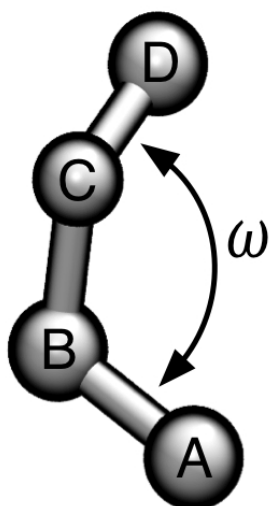


Figure 4- Illustration of the torsional angle definition.

Unlike the E_{str} and E_{bend} functions, the harmonic potential is not the perfect solution to calculate the torsional energy mainly by the absence of descriptors for two observed behaviors in this type of distortion: one is the periodic movements allowed when the atoms are rotated in turn of the central bond, and the other is the low energy requirement to alternate between different minima.

To incorporate the missing aspects, a Fourier series (6) is used instead of the harmonic potential.

$$E_{tor}(\omega^{ABCD}) = \sum_{n=1} \frac{V_n^{ABCD}}{2} [1 + \cos(n\omega^{ABCD} - \gamma)] \quad (6)$$

The periodicity of this function is acquired by the n term, which provides to each period n^{th} rotation angles: $n = 1$ term describes a 360° periodicity, $n = 2$ describes a period in each 180° , $n = 3$ term determines periodicity in each 120° , and so on. The V_n constant is the size of the barrier of the rotation around the B-C bond. The γ factor, the division by 2 and the summation of 1 are operations necessary to produce standard outputs so the values returned by this equation may have a user defined minimum at 180° , and the E_{tor}/V_n values vary between one and zero.

2.1.2.4. The Van Der Waals Energy

The van der Waals (E_{vdw}) energy function term is the first descriptor for the non-bonded atoms interactions. This term is correlated with the non-polar interaction between the atoms, describing the attraction and repulsion provoked by the electron clouds surrounding the nuclei. This physical property explains the energy variation, in which at very small distances,

the E_{vdw} energy becomes positive, reflecting the repulsion forced by the overlap of the electron clouds. At median distances, about 3.5 Å, this energy becomes slightly negative (attractive) due to the induced dipole-dipole forces, tending to zero at very long distances, a R^{-6} dependency, being R the distances between the two atoms.

The Lennard-Jones potential (7) reflects the behavior described above, presenting two parameters, the van der Waals radius R_0 and the softness factor (ϵ).

$$E_{LJ}(R^{AB}) = 4\epsilon \left[\left(\frac{R_0^{AB}}{R^{AB}} \right)^{12} - \left(\frac{R_0^{AB}}{R^{AB}} \right)^6 \right] \quad (7)$$

The 12 exponential factor of the repulsive fraction of the equation is not related with physical evidences or theoretical basis, in which the values 9 or 10 returned better results. The usage of this factor is only associated with computational intrinsic features, which are calculated by squaring the attractive fraction with the exponential factor 6.

If a better descriptor is required, it is possible to calculate this energy using two different potentials: on is the “Buckingham” or “Exponential R^{-6} ” potential (8), and the other one is the Morse potential. In the first potential, A B and C factors are constants and the R^{AB} is the distance between the atom A and B.

$$E_{vdw}(R^{AB}) = A e^{-BR} - \frac{C}{(R^{AB})^6} \quad (8)$$

Despite the better accuracy of this potential, its usage can also constitute a source of error, especially at very small distances, where the $(R^{AB})^{-6}$ tends to $-\infty$ creating a “nuclear fusion”, instead of becoming a positive repulsive energy. The Morse potential increases the accuracy of the results in all circumstances, but it also increases the computational time necessary to calculate these interactions like explained before.

The non-bonded atom interactions require special attention in terms of the computational time requirements. This problem becomes even more important in biological systems, since they contain a large number of atoms. While the number of bonded interactions increase linearly with the number of the atoms, the non-bonded interactions grow as the *square* of the system's size, turning the calculations of these energies the predominant procedure in the force field energy calculation.

To reduce the calculation time of the non-bonded energies, one or more cut-off distances can be applied. In the case of one cut-off distance, in which the van der Waals energy becomes zero at great distances, the potential becomes a discontinuous descriptor. To solve this problem, a two cut-off distance approach can be adopted, in which a smothering function is applied between the two distances values, where the energy slowly tends to zero between these two energies.

2.1.2.5. The Electrostatic Energy

The electrostatic energy (E_{ele}) term of the force field energy equation is the second non-bond interaction descriptor. This term describes the electrostatic interactions between the average point charges of the atoms in the system and it is calculated based on the Coulomb potential (9).

$$E_{ele}(R^{AB}) = \frac{Q^A Q^B}{\epsilon R^{AB}} \quad (9)$$

In this potential, Q^A and Q^B are the charges of the correspondent atoms A and B, R^{AB} is the distance between them and ϵ is the dielectric constant of the medium. The dielectric ϵ can be parameterized to simulate the electronic environment of the system, such as $\epsilon = 1$ for vacuum, or higher values; for instance $\epsilon = 4$ for proteins and $\epsilon = 80$ for water, simulating the water as the solvent in the absence of explicit water molecules.

In a physical sense, the atom point charge is not fully correct, being adopted as the point that describes better the arrangement of the electric field around the atom. This concept is applied in the AMBER force field through the Restrained ElectroStatic Potential fit (RESP).

The E_{ele} energy describes not only the influence of the formal charged atoms in the system, just like the oxygen atom in a hydroxide molecule, but also the partial charged atoms. Here, dipole-dipole interactions and hydrogen bonds are also implicitly included. For instance, an alcohol oxygen atom partially negatively charged would have the tendency to interact favorably with a partially positively charged hydrogen atom from another alcohol group. One of the most severe limitations of molecular mechanics is related to this function, in which the surrounding atom environment has no influence in the point charges of the atoms, keeping their charges static through the simulation calculation.

2.1.2.6. The Force Field Parameterization

Besides the importance of the equations choice to describe the different terms of the force field, the efficient and accurate parameterization of the constants in the equation also plays a central role in the molecular mechanics methods. In spite of no force field parameterization having been performed in this work, this subject has a relevant importance in molecular simulations.

For an ordinary structure such as proteins, most of the parameters are included in the molecular simulations packages. However, this is not the case for most substrates or inhibitors, in which the calculation of parameters like the stretching energy constant force K^{AB} , or the torsional barrier size, is required. This last term is problematic due to the number of possibilities.

There are two ways for a computational chemist to obtain values for these parameters: one is deriving them from experimental values, but we are always conditioned by their availability, and the other option is by computational means, which are relatively easier, cheaper and straightforward to apply. Here, sophisticated computational methods such as Density Functional Theory (DFT) or post Hartree-Fock methods can be applied. In the case of the stretching energy, the parameters can be obtained by structure optimization and posterior scan, where the bond length is stretched and shrunken until the equilibrium value is reached. Later, this data is fitted to the harmonic potential or other potential.

The van der Waals parameters are normally assigned to single atoms directly from the experimental data, and later combined for the diatomic value. The simplest approach is to calculate the diatomic van der Waals distance as the sum of the two van der Waals radii, and the softness factor as the geometrical mean of the atomic softness constants.

As mentioned above, the charge parameter should describe the electrostatic field surrounding the atom, instead of a group of single point charges. These charges are calculated by the electrostatic potential of a series of points in a grid around the desired molecule, and adapted later to single points that reproduce better those values.

2.1.3. Molecular Dynamics

One of the most relevant applications of the molecular mechanics principles is in the description of the trajectories of the atomic structures throughout time. Here, Molecular Dynamics (MD) simulations describe the atoms' Cartesian coordinates that have suffered the influences of the other atoms. The movements of the atoms are simulated by solving Newton's second equation, $F = ma$, in order to obtain a time dependent descriptor (10).

$$\frac{d^2x_i}{dt^2} = -\frac{F_{x_i}}{m_i} \quad (10)$$

The previous equation describes the trajectory of particle i with mass m_i along one coordinate x_i with F_{x_i} representing the force acting on the particle in that particular direction. This force is calculated based on the potential energy obtained from the terms described in the earlier sections. This mobility description is deterministic, meaning that for any value of time variable, the state of the system can be calculated once the position and the velocities are deduced.

In Molecular simulations, this function is applied by the Newton's equation as a function of 3N coordinates (i.e. the 3D spatial location). The complexity of the systems usually analyzed by this function turns the analytical solution impossible, requiring therefore the numerical

resolution of the equations. To do so is necessary to define a finite integration step denominated simulation time step (δt).

2.1.3.1. Simulation Time Step

The proper parameterization of the integration step requires special care that will determine the right description of molecular behavior. This time step must be set as a function of the quickest motion period to be described, and at the same time, the value that returns a relative economy in terms of computational demand and time consumption. The wrong parameterization of this value can lead to a mismatched description of the atoms trajectories due to the abrupt variation of the potential energy, or a software breakdown due to the numerical overflow.

The common rule applied to choose this value is the calculation of the number that is at least one order of magnitude smaller compared to the highest-frequency vibration motion in biological structures. This motion is the bond stretching movement involving hydrogen atoms, which is around 10 fs (e.g. the C-H bond stretching), originating an integration step of 1 fs. Despite of the fact that C-H bond stretching is the fastest motion period, its influence in terms of the potential energy variation is not significant, and the respective fluctuations do not imprint a significant numerical instability, giving space to restrain their variation and therefore save computational time. To perform these constrains, the SHAKE algorithm⁴³ is usually applied, in which the bonds containing hydrogen atoms are frozen to their equilibrium values, and the highest-frequency vibrations in the system are now the bonds stretching between heavy atoms such as the C-C bond. These frequencies have a motion period between 2 and 5 times slower (20 to 50 fs) than the hydrogen intervenient bonds, which is translated in a suggested time step of 2 fs.

2.1.3.2. Simulation Time Scale

Besides the integration step, there is another time variable, which is the time simulation. This time scale defines the time range in which the atoms trajectories will be simulated. The value of this variable is related to the kind of structural motions that is intended to observe. For instance, if only the atoms position fluctuations or side chains motions are intended to simulate, the time range necessary to observe these variations is in the order of picoseconds. On the other hand, loop motions can take several nanoseconds, a protein subunit motion or even the folding and unfolding of protein could be out of the scope of this method. The choice of these values can also be influenced by the size of the system. Here, the total cost in terms of computational time can be very high. The values of time scale that

establish a good commitment between the computational time (real time) and the quality of the simulated trajectory are in the range of the hundreds of nanoseconds for the enzymatic systems.

2.1.3.3. Boundary Conditions

In biological environment, it is possible to admit that a molecule is immersed in infinite universe, or has infinite space around it in all directions. In a molecular dynamics simulation containing explicit solvent water molecules (e.g. TIP3P⁴⁴) instead of an implicit description of the solvent, is necessary to define the simulated space. In this work, this option was not applied, but like the force field parameterization, boundary conditions are also very important in the molecular dynamics simulations field.

As mentioned beforehand, the size of a molecular structure, for instance, a protein, compared to the intracellular space is substantially small, considering that there is infinite space surrounding a protein in a biological environment. Being computationally impossible to simulate an infinite space, a set of cut-off distances defining the borders of the system must be applied. Here, a box containing the structure and the solvent molecules must be defined. The edges of the simulation box are described as the edges of the copies of simulation box. These copies placed around the simulation box are in contact with each other in a way that, for instance, a solvent molecule describing a trajectory that goes out of the bottom of the simulation box, will appear on the top of this same box.

The size of the simulation box must be defined carefully in order to avoid self-interaction (the interaction between a molecule located in a boundary limit and itself). As mentioned in the E_{ele} section, a set of cut-off distances must be used, and a special method to deal with these interactions is through the application of the Particle Mesh Ewald (PME)⁴⁵.

2.1.4. Predicting the Association Energy

One of the most important values to be determined in this kind of simulation is the Gibbs free energy of association. The importance of these values is revealed in several computational studies, such as the evaluation of the binding of a protein with small ligands (e.g. enzyme-ligand binding), or the binding between larger structures like two proteins⁴⁶⁻⁴⁹. It is in the former subject that this free energy prediction took relevant weight in this work. Here, the method used to calculate the Gibbs free energy (ΔG) was the Molecular Mechanics/Poisson-Boltzmann Surface Area (MMPBSA)⁵⁰ implemented in the molecular simulation package AMBER³⁰. More accurate methods could be used for the same purposes, such as the Free-Energy Perturbation(FEP) and the Thermodynamics Integration(TI)

methods, which in fact return better results. The reason for the choice of the MMPBSA method in detriment of FEP and TI is the higher computational efficiency of the MMPBSA. For protein-protein association analyses, MMPBSA presents relatively good agreement to the experimental results without compromising the computational time required. Only the MMPBSA method will be described in the following section because it was the only free energy calculation method used in this work.

2.1.4.1. Molecular Dynamics/Poisson-Boltzmann Surface Area

The values of free energies calculated through molecular mechanics methods are not accurate enough to perform, for instance, a comparison between different systems. This feature relies in intrinsic properties of the MM method where the zero point of the energies descriptors must be parameterized, which in most cases is not the same for all force fields and the approximations necessary to solve the equations mentioned beforehand also induce a certain level of errors. This error accumulation leads to significant differences in the final value, making it devoid of scientific meaning. Instead of considering only one final free energy value, the binding free energy values can be compared based on error cancelation, retrieving a variation on the binding free energy, which is provided of scientific accuracy. In this work, these values are used for the analysis of the variation of the binding free energy of the system when a residue on a protein-protein surface is mutated by an alanine. The resulting Alanine Scanning Mutagenesis Method is described in the results section.

The MMPBSA methodology conjugates a molecular mechanics energy calculation with a molecular dynamics simulation in implicit solvent. The binding free energy can be calculated using the thermodynamic cycle shown in Figure 5, in which ΔG_{gas} is the interaction free energy between the two binding partners in the gas phase, and ΔG_{solv}^{lig} , ΔG_{solv}^{recep} and $\Delta G_{solv}^{complex}$ are the solvation free energies of the two binding partners and the complex, respectively.

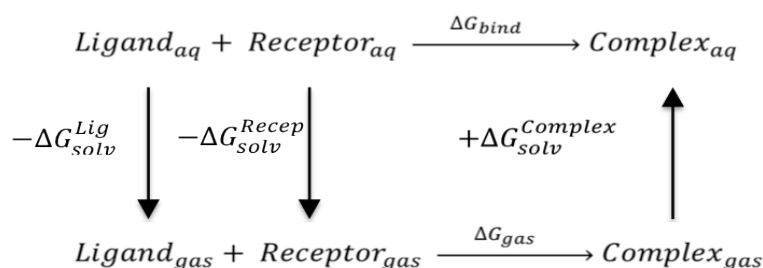


Figure 5: Thermodynamic cycle used to calculate the binding free energy

The binding free energy of two molecules in a complex is defined as the difference between the free energy of the complex and the respective monomers (11).

$$\Delta G_{\text{binding-molecule}} = G_{\text{Complex}} - G_{(\text{Ligand-Receptor})} \quad (11)$$

The free energy of the complex and the respective monomers can be calculated by summing the internal energy (bond stretching, angle bending, and torsional energy), E_{internal} ; the electrostatic and the van der Waals interactions, $E_{\text{electrostatic}}$ and E_{vdW} ; the free energy of polar solvation, $G_{\text{polar solvation}}$; the free energy of nonpolar solvation, $G_{\text{nonpolar solvation}}$; and the entropic TS contribution for the molecule's free energy as is written in equation (12).

$$G_{\text{molecule}} = E_{\text{internal}} + E_{\text{electrostatic}} + E_{\text{vdW}} + G_{\text{polar solvation}} + G_{\text{nonpolar solvation}} - TS \quad (12)$$

The first three terms are retrieved from the force field applied in the molecular dynamics simulation. The electrostatic solvation free energy can be calculated by solving the Poisson-Boltzmann equation with the Delphi software⁵¹⁻⁵³, which has been shown to constitute a good compromise between accuracy and computing time. During the Poisson-Boltzmann equation solving process, two different dielectric constants are assigned in order to simulate two different environments: one refers to the external/solvent medium which is normally assigned with the value 80 ($\epsilon = 80$) for water solvent medium or $\epsilon = 1$ in case of vacuum, and $2 < \epsilon < 4$ in the case of the solute/protein environment.

The nonpolar contribution to solvation free energy by the van der Waals interactions between the solute and the solvent, and cavity formation, is modeled as a term dependent on the solvent-accessible surface area of the molecule. It is estimated using the empirical relation,

$$\Delta G_{\text{nonpolar}} = \alpha A + \beta \quad (13)$$

where A is the solvent-accessible surface area calculated by the molsurf program, which is based on the idea primarily developed by Michael Connolly⁵⁴ and it is implemented in the AMBER molecular package. α and β are empirical constants, with values $0.00542 \text{ kcal } \text{\AA}^{-2} \text{ mol}^{-1}$ and $0.92 \text{ kcal mol}^{-1}$, respectively. The entropy term, obtained as the sum of the translational, rotational, and vibrational components, will not be calculated because it is assumed, on the basis of previous work, that its contribution to the $\Delta\Delta G_{\text{binding}}$ is negligible^{55,56}.

2.1.5. Molecular Docking

Molecular Docking is one of the most used molecular modeling methods in computer-aided drug design. This method is usually the first approach in the drug discovery process when the receptor structure is known, often extracted from the Protein Data Bank (PDB)^{11,12} or similar database. Instead of just characterizing the position and orientation of one ligand inside the receptor binding site (docking campaign), this method reflects its full benefits in the characterization and sorting of several ligands (virtual screening). The main goal of virtual screening campaigns is to select the compounds that have higher probabilities to develop a strong binding (e.g. inhibitor or substrate) and ranking them by their docking score and/or predicted binding energy.

Molecular docking software are characterized by their search algorithm, sometimes named optimization algorithm, and their scoring function. The differences in the search algorithms are related to the model and the allowed degrees of freedom in the ligand and receptor structures. Looking at the scoring functions, some software adopts molecular mechanics model approximations like in the force field energy calculation, or empirical or knowledge-based scoring functions. The combination and tuning of the search algorithm and scoring function is always dependent of the accuracy/computational demand relationship that dictates the general usage of the program, especially in databases containing a vast number of compounds.

There are good reviews in the literature concerning molecular docking, e.g. the reviews from Sousa *et. al*^{18,57}.

2.1.5.1. Search Algorithm

The simplest search algorithm to be considered is an algorithm that treats all the complex units as rigid bodies and only the rotation and translation of the units around each other are considered. This type of algorithms can be very useful in virtual screening campaigns, in which a large amount of ligands are evaluated. The speed of this kind of methods allows the analyses of large databases of small ligands, and suits even better in the docking of large structures as is the case of protein-protein docking⁵⁸. On the other hand, the molecular description is not enough in the cases where some conformational changes have to take place in order to allow a proper interaction between the ligand and the receptor⁵⁹. In these cases some flexibility (or even the full flexibility) of the complex is required and is implemented in the search algorithms of the docking software through three different methods: systematic, random or stochastic and simulations methods.

In the systematic methods category, the conformational search, fragment-based and database-based algorithms are included. The conformational search algorithm is based on the analysis of all possible conformations originated by the rotation of the dihedral angles of the ligand. In case of the fragment-based, the algorithm tries to bind small parts of the ligand, adding groups to the initial structure until the full structure is completely docked in the receptor binding site. Regarding the database-based method, this algorithm is based on the docking of rigid structures. The ligand structures are generated computationally or experimentally and only the structures that present higher probability to exist are considered.

Random or stochastic methods are the most applied search algorithms in docking software. An example of a random algorithm is the Genetic Algorithm (GA) implemented in AutoDock or Gold^{60,61}, where the search routine is based in the Darwin survival theory and the Mendel genetics processes like mutations and genes transmission. Another algorithm often implemented in docking programs is the Monte Carlo algorithm, which is very useful when it is intended to evaluate a wide range of conformations, such as the Tabu algorithm. In this algorithm several structures are generated avoiding the creation of equal structures, giving also a wide variety of structures.

The simulation search methods apply molecular dynamics simulations and/or energy minimization calculations. These methods are suitable for the optimization of the structures resulted from a previous virtual screening or docking campaign. The convergence rate of these algorithms is low and very computational demanding.

Published works like MADAMM (multi staged docking with an automated molecular modeling protocol), from Cerqueira *et al.*³⁸, apply a combination of systematic algorithms and simulation methods, with a rigid body docking in order to improve the docking results. The major drawback of these protocols is the time required to evaluate all the conformations and the molecular simulations, which despite of improvements, are still considerably slow for a general application in a large database.

2.1.5.2. Scoring Function

Searching for the best conformation of a ligand-receptor binding complex could not be complete if the evaluation of these conformations took place. Actually, the evaluation (scoring) and the search routines are intercalated in the docking process. For instance, in the Genetic Algorithm, the elements of the genes can only be transmitted to the next generations if they are fit to survive.

As with the search algorithms, different software apply different methods to evaluate the binding of the ligand into the receptor binding site, namely: force field, empirical, knowledge-based, and consensus scoring functions. The force field scoring functions are based on the

numerical calculation of the energy applied in the molecular energy descriptors described in the 2.1.2 section, just like internal energy calculation (bond stretching, angle bending and torsional energy). As mentioned beforehand, as the molecular descriptor becomes more sophisticated, the computational demand becomes an issue, turning the application of this kind of scoring functions not practical for fast docking processes.

The empirical scoring functions are based on experimental observations, in which it is intended to recreate these values by approximated and parameterized functions. Despite the speed of calculation of these functions, these parameters are not transferable to all structures, demanding constant parameterization, function tuning and constant search for experimental values (which are not availability for all cases). Knowledge-based functions employ the same parameters approximation, but instead of an energy calculation, these functions are focused on geometry optimization.

Instead of being a different method to score docked conformations, the consensus functions are the combination of the previously mentioned scoring functions. This approach is based on error cancelation while maintaining the advantages of the features of each method.

3. Results and Discussion

This chapter intends to report the accomplished work, either published or to be published, describing the resultant software. The descriptions of the new bioinformatics tools are based on the correspondent scientific papers, complemented with relevant information regarding the software used to perform molecular calculations, theoretical information or developments carried out since their release. The visual inspection of the structures and the results is obtained by the inclusion of all developed tools in the world wide web molecular visualizer, Visual Molecular Dynamics (VMD).

The first section of this chapter describes the software vsLab, which was developed to assist the user in virtual screening campaigns and is based on one of the most used molecular docking software, AutoDock. This program allows the user to perform a virtual screening campaign in a very easy-to-use fashion. All the parameters required to carry a molecular docking prediction are set in a Graphical User Interface (GUI) in which the resultant ranking of the docked ligands is displayed, sorted by the docking score returned by the AutoDock software. The code under the GUI manages all the files necessary to launch the AutoDock calculations and also deals with all intermediate steps of the molecular docking protocol, including the structure files preparation and the generation of the affinity atoms grid maps required by the AutoDock software.

The second part of this chapter focus on the description of the Chem-Path-Tracker (CPT) software. This tool aims to find and highlight chemical motifs present in structures. These motifs can be formed by different type of structures and purposes, such as the coordination of a metal-enzyme binding site, the alignment of different residues in order to form a cation- π interactions chain, or the hydrogen bonds formed inside a water tunnel. The algorithm implemented in the CPT software is a modified version of the Dijkstra's algorithm, which main goal is to find the closest path between two points containing several nodes between them. Defining the nodes as atoms or residues of different types is possible to trace different paths between the nodes to form cluster/motifs of nodes. In this way is possible to highlight structural motifs and thus reveal the surrounding environment.

The third section of this result and discussion chapter stands for the description and validation of the molecular structure analyzer VolArea. This software calculates and analyzes different structural features like the surface area accessible to different compounds such as the solvent, calculating in this way the solvent accessible surface area (SASA). VolArea is also constituted by an algorithm to calculate volume, both atoms and empty spaces volume such as cavities and clefts. This program has proven itself to be very useful in molecular dynamics simulations analysis, as it was possible to highlight and quantify events observed in the visual analyses, such as the closure of the binding site. It was also possible to analyse

contacting surfaces, such as protein-protein interacting surfaces. The volume algorithm was later improved and adapted to the usage of the Graphical Processing Units (GPU), providing this tool with a high parallelized algorithm, originating a huge gain in computing time.

The forth and final section of this chapter is devoted to the Computational Alanine Scanning Mutagenesis (CompASM) tool. This software was developed in order to bring the alanine scanning mutagenesis (ASM) method to the general usage, allowing the usage of this methodology by non-expert users. The ASM method, either experimental or computational, maps the importance of residues in a protein:protein interface surface. The method is based on the variation in the binding free energy of a complex triggered by the mutation of a residue by an alanine residue. The computational variant of this method has shown good balance between computational time and accuracy, when compared with more accurate technics such as Thermodynamic Integration (TI) or Free-Energy Perturbation(FEP) calculations. The CompASM protocol uses a molecular trajectory performed in the AMBER molecular simulation package in implicit solvent, using the Generalized Born solvation method, and the calculation of the binding energy applying the Molecular Mechanics/Poisson-Boltzmann Surface Area (MMPBSA) method, also implemented in AMBER.

3.1. Virtual Screening Lab (vsLab)

Molecular docking is nowadays an important tool in the search for promising compounds as drug candidates, having become a daily used tool in the drug discovery process. Despite the relative success of some docking software in the prediction of ligand-receptor binding conformations, the complexity of these programs are often high, requiring a slow learning process, which leads to difficulties in the usage of these software by non-expert users. This is an obstacle and a cornerstone issue for the research teams in the fields of Chemistry and Biochemistry, whom are interested in conducting this kind of calculations but do not have enough programming skills. To overcome these limitations, we have designed vsLab (virtual screening Lab), an easy-to-use graphical interface for one of the most used molecular docking softwares AutoGrid/AutoDock; vsLab has been included into VMD as a plug-in. This program allows almost anyone to use AutoDock and AutoGrid for simple docking or for virtual screening campaigns without requiring any deep knowledge about these techniques. The potential associated to this plug-in makes it an attractive choice not only for educational purposes, but also for more advanced users whom are able to use vsLab to increase workflow and productivity of everyday tasks.

Article I

“VsLab - An implementation for virtual high-throughput screening using AutoDock and VMD”

Adapted from:

Cerqueira N.M.F.S.A., Ribeiro J., Fernandes P.A, Ramos M. J.

Int J Quantum Chem 111: 1208–1212, 2011.

3.1.1.VsLab - An implementation for virtual high-throughput screening using AutoDock and VMD

N.M.F.S.A. Cerqueira, J. Ribeiro, P.A Fernandes, M. J. Ramos

REQUIMTE , Departamento de Química e Bioquímica, Faculdade de Ciências, Universidade do Porto, Rua Campo Alegre s/n , 4169-007 Porto, Portugal.

3.1.1.1. Abstract

Molecular docking is a key tool in structural molecular biology and computer-assisted drug design. The goal of ligand-protein docking is to predict the predominant binding mode(s) of a ligand with a protein of known three-dimensional structure. There is much docking software that has been successfully used in a myriad of keystone problems, however, as commonly happens with most of the scientific software, the programs are often complex and a deep knowledge is required for the common user to carry out standard steps. This is an obstacle and a cornerstone issue for the research teams in the fields of Chemistry and the Life Sciences, who are interested in conducting this kind of calculations but do not have enough programming skills. To overcome these limitations, we have designed vsLab (virtual screening lab), an easy-to-use graphical interface for the well-known molecular docking software AutoGrid/AutoDock that has been included into VMD as a plug-in. This program allows almost anyone to use AutoDock and AutoGrid for simple docking or for virtual screening campaigns without requiring any deep knowledge about these techniques. The potential associated to this software makes it an attractive choice for more advanced users that can use vsLab to increase workflow and productivity of everyday tasks. The program can be freely obtained at <http://www.fc.up.pt/pessoas/nscerque/vsLab/vLab/HomePage.html>

Keywords: Molecular Docking, Virtual Screening, vsLab, software, AutoDock.

3.1.1.2. Introduction

In the field of drug discovery, molecular docking software is playing a relevant role, as identifying molecules that might fulfill the stringent criteria necessary to become a drug has often been compared to searching for a needle in a haystack. These methodologies seek to simulate the structure of protein/ligand complexes, in which the receptor is usually a protein or a protein oligomer and the ligand is either a small molecule or another protein. The

solutions are then sorted out using a scoring function, which allows estimating the conformation of the ligand that tends to interact more strongly with the studied receptor⁶².

Initially, molecular docking was only used to predict and reproduce protein-ligand complexes, but the successes of these early studies led to the exploration of molecular docking as a promising tool in drug discovery in order to find hits and optimize lead compounds. In the quest of predicting this type of interactions, several docking algorithms have been developed in the last two decades with commercial, freeware and open source licences⁶³⁻⁷⁰. The efficiency and accuracy of these programs is well established in the literature, where a source of successful examples and software comparisons can be easily found^{71,72}. However, as commonly happens with much scientific software, the programs are complex to use and a deep knowledge is required for the common user to carry out standard procedures. In the particular case of molecular docking software, the learning step increases when the user intends to repeat the same task several times in order to screen thousands or millions of compounds.

As these methods are becoming standard tools for everyday use, in a wide range of experimental sciences in the fields of Chemistry and the Life Sciences, there is a constant challenge to turn these struggling programs more user-friendly and with more accessible front-ends to allow their use by non-expert users. In the literature, there are many graphical user interfaces (GUI) that help the user to prepare the input files and the execution of common tasks in these types of software. However, the number of them that are free, user friendly and allow preparing virtual screening campaigns with a suitable graphical backend to analyse the results is very limited. Taking this into account we developed a plug-in for the VMD (Visual Molecular Dynamics) program⁷³, called vsLab, which allows any user to dock one or several ligands into the binding pocket of a protein in one step fashion, without the frustrating preparation of several input files and the execution of multi sequential and complex steps.

The main idea behind vsLab is therefore to turn molecular docking as easy as it can get, allowing its use in everyday work situations in a couple of clicks, without disregarding the reliability and reproducibility of the results. To ensure this key asset, vsLab uses in its core system, one of the most popular molecular docking software, i.e., AutoDock⁷⁴. The developed GUI has several modules that ensure a smooth execution of the program, requiring only minimal user intervention to start a simple docking job or more complex high-throughput virtual screening campaigns.

3.1.1.3. AutoDock

VsLab allows to predict the docking of a single ligand to the binding pocket of a protein, or simulate virtual screening campaigns searching for hit or lead compounds from a large compound database. In order to improve the attractiveness and utility of the program, it has been encapsulated into the VMD program ⁷³, and uses AutoDock ^{75 74} to perform molecular docking simulations to ensure the reliability and confidence of the results.

We have chosen AutoDock because it is one of the most popular molecular docking software available in the market and is freely available to download, under the GPL license, by any user ⁵⁷. This software can efficiently predict, with an elevated degree of confidence, the correct binding pose of several ligands in a wide range of receptors ^{76 68 77-80}. This program is now in version 4.2 and with the AutoDockTools facilities allows the application of this technique to a very large range of biological systems. The search algorithms available in AutoDock are the Genetic Algorithm (GA), the Local Search (LS) algorithm and the GA/LS hybrid algorithm named Lamarckian Genetic Algorithm (LGA). This last algorithm was introduced in the third version, being the most promising search algorithm offered by this program. The Genetic algorithm implements genetic notions, applying them to the data in study. For instances, in case of molecular structures, the 3D coordinates of the atoms are the phenotype and all information regarding the angle rotation angles, torsions etc. is printed in the genotype. The gains of the Lamarckian algorithm against the GA and LS algorithms is in the local minimum search between generations, and the inclusion of this information in the descendants and the efficiency in the convergence obtained by this inclusion. This feature is the key character of the Lamarckian theory assuming the transmission of the phenotype characteristics to the next generations. The scoring function implemented in the AutoDock is a semi-empirical force-field based scoring function, where the internal, torsional and electrostatic energies are evaluated applying the parameterization already implemented in the software. Different parameters are also allowed by the inclusion of a parameterization file, which like in most cases, is not advised due to the introduction of a new source of error.

In spite AutoDock usage advantages, the use and handling of AutoDock is not straightforward for non-expert users, since it requires several separate pre-docking steps, e.g., ligand preparation, receptor preparation, and grid map calculations, before the actual docking process can take place. Existing tools, such as AutoDockTools (ADT) ⁷⁴ and BDT (Automatic protein-ligand docking for everyone) ⁸¹, integrate individual AutoDock steps within a graphical user interface, and provide automated features for simple docking runs. Still, neither its use is trivial nor it contains the capability to effectively process and analyze millions of compounds in a single execution. The former issue can be overcome using a set of UNIX scripts, but its execution is rather difficult and complex, which turns its use almost impossible for non-expert UNIX users.

3.1.1.4. Graphical User Interface

vsLab intends to solve most of difficulties encountered in the docking process and considerably reduces the learning step required to start using this type of software, avoiding the creation of the several input files and complex high-level automated scripts to execute the program. To accomplish this objective, the program includes a set of graphical user interfaces that turns molecular docking process as easy as a couple of clicks.

vsLab is installed as a VMD plug-in and only few dependencies, concerning the localization of AutoDock and AutoGrid executable files and AutoDockTools path, have to be set up. Once installed the program can be found in the extension menu of VMD, under the sub-menu called Modelling.

When the program is launched the main interface is composed by a set of menus that can be used to prepare a new run, open previous executed files, and start new calculations, among other options. In the same interface we can also find two tabs that are used to display and sort out the results of single or multiple docking jobs, as well as review the parameters used to run those calculations (Figure 6A).

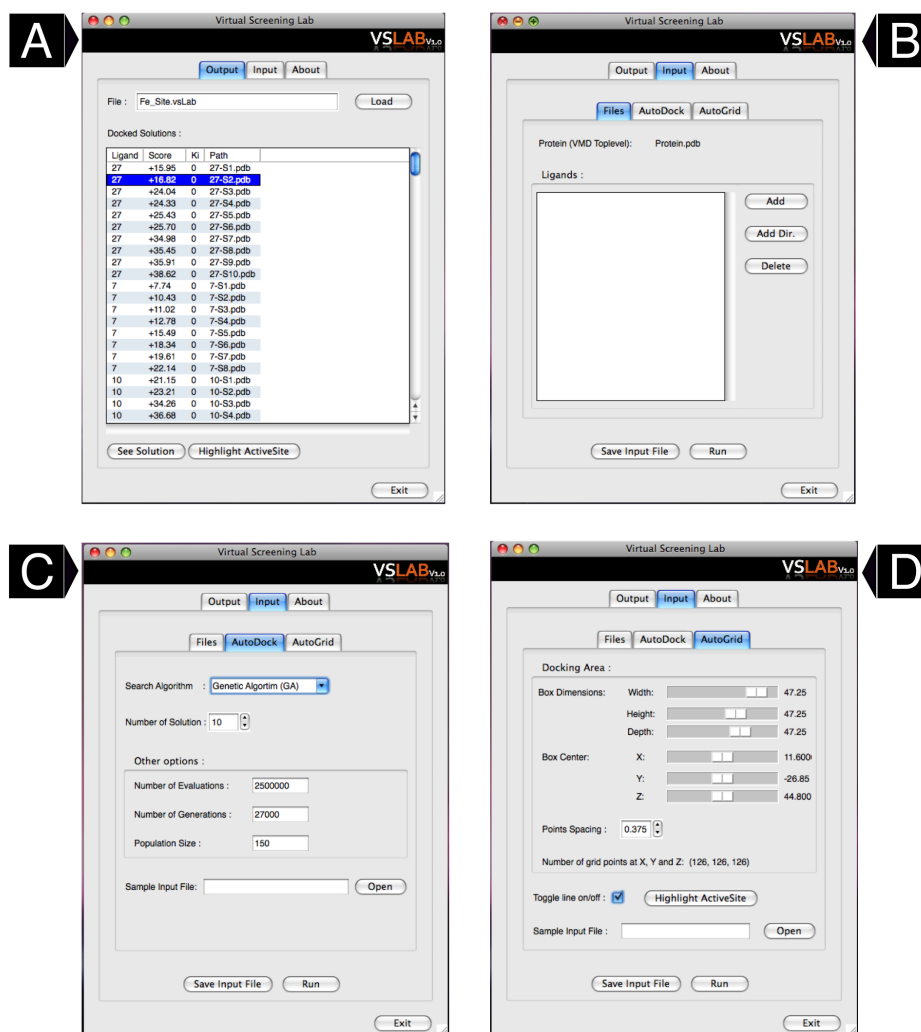


Figure 6: Graphical user interfaces available in vsLab. A: Main interface. B: Input file interface. C: AutoDock options interface. D: AutoGrid options interface. E: Visualizing the results using VMD front-end.

To create a simple docking or a virtual screening campaign job, the user just has to select the receptor and the ligand, or group of ligands, which are going to be screened during the docking procedure (Figure 6B). The receptor must be previously uploaded to the VMD interface and selected on the interface. The ligands can have the pdb or the mol2 file format and they can be independently uploaded into the interface, one at a time or simultaneously through the selection of the directory where they can be found. The latter option is useful when the user wants to dock several thousands or millions of compounds into the same receptor and undergo virtual screening campaigns.

The same window also allows the user to set up the AutoGrid and AutoDock parameters that are required to run the docking simulation. The AutoDock Tab allows the user to change a set of parameters that control the execution of the AutoDock program. For instance, it allows to adjust the type of algorithm that will be used during the docking process, and set up the parameters that control its execution (Figure 6C). All entries in this interface are pre filled with standard values, but the user can modify them for his/her needs, namely, the number of

solutions that will be generated in each run, the number of evaluations, as well as the number of generations and the population size that control the termination of the genetic algorithm.

The AutoGrid tab includes all parameters that are necessary to run the AutoGrid executable. This program maps the binding site of the target protein, and calculates the atomic affinity potentials for each atom type in the macromolecule, such as aliphatic carbons, aromatic carbons, hydrogen bonding, oxygens, etc. To select such region vsLab offers a set of buttons that allows the user to draw, resize and move a box around the protein easily (Figure 6D). The region inside of that box is the one that will be analyzed by the AutoGrid executable, and should therefore include all atoms/residues of the receptor that the ligands should interact with (in this part it is advisable that the user uses the selection tools of VMD to highlight the active site). The height, width and depth units of the square box present in the interface are in angstrom, but these values are normally converted to a number of grid points in AutoGrid, i.e., the number of points that will be used within the grid map (in each axis) to calculate the potential energy of a specific 'probe' atom or functional group over a 3D space around the receptor surface. The number of grid points in each axis depends therefore on the size of the square and on the space that is used to distance each point on the grid (which can also be easily changed in the vsLab interface, in the *grid spacing entry*).

Since sometimes it is not clear to evaluate which area of the receptor is being analyzed, the interface offers one option that calculates the ligand accessible surface area for that selection. The resolution of the surface can be modified and the color changed by the type of amino acids, exposed area, etc, which can help the user to choose or examine the binding region with more precision.

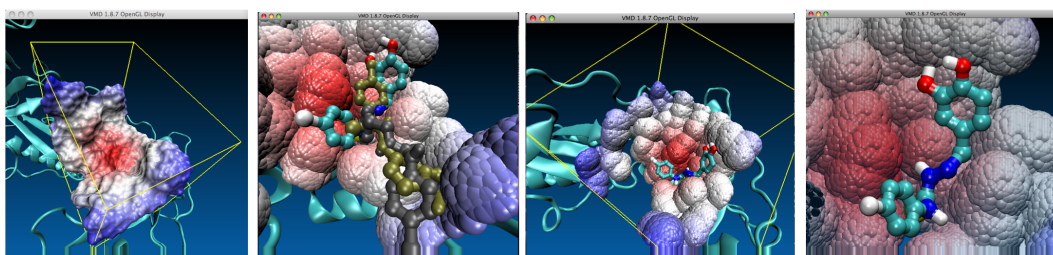


Figure 7: Sample VMD sessions displaying the results obtained from the vsLab plug-in.

Once these variables have been set up and saved, the user can submit the input file for calculation. The process can take several minutes or several days depending on the number of ligands that are analyzed and the variables imposed to the system. During the execution of the input file, vsLab stores all data in a database that can be subsequently uploaded in the same interface and analysed using the powerful graphical front-end of VMD (Figure 7). The interface allows the user to sort out the results taking into account the score attributed by AutoDock to each docked ligand/solution, or by the calculated KI (Inhibition constant) that

estimates the inhibitory strength of that ligand for the analyzed protein. These values can then be used to select the best binding pose of a putative ligand inside the active site of the receptor (if only one ligand was analyzed), or search for possible hit or lead compounds, if more than one ligand has been investigated.

VsLab can also be used when the location of the binding site is unknown. This is commonly referred to as “blind docking”, when all that is known is the structure of the ligand and the macromolecule. In such cases, the user has to set up the program to search for the entire surface of the protein (or other macromolecule) of interest. In those situations, it is advisable to divide the macromolecule in different and adjacent sections in order to attain a good balance between the analyzed area and the resolution of the grid map, which should contain the maximum number of points in each dimension⁸².

3.1.1.5. Conclusions

vsLab removes most of the complexities and organizational problems associated with the use of molecular docking software. It provides a convenient and efficient solution to dock a ligand to a specific receptor as well as to implement more complex high-throughput virtual screening jobs. The docking session is fully integrated and automated and all inputs files are specified via a graphical user-friendly interface that allows any user to use it without requiring any previous knowledge on the molecular docking area.

The inclusion of AutoDock in vsLab guarantees the reproducibility and reliability of the results and the VMD symbiosis allows the user to explore its powerful graphical front-end to analyze the final results. The simplicity of the software makes it also an attractive tool not only for education purposes, but also for more advanced users that can use vsLab to improve the workflow and production of everyday tasks. We believe that vsLab contributes significantly to the progress of the research teams in the fields of Chemistry and the Life Sciences.

3.2. Chem-Path-Tracker

The internal organization of proteins is a crucial subject in the study of their structure. The function of an enzyme, for instance, is regulated and mediated by the action of several fragments (motifs) that together lead to the correct catalytic mechanism. However, the detection of these chemical motifs is rather difficult because they often consist of a set of amino acid residues separated by long, variable regions, and they only come together to form a functional group when the protein is folded into its three dimensional structure. In order to simplify the analysis of these chemical motifs and give access to a generalized tool for all users, we developed Chem-Path-Tracker. This software is a VMD plug-in that allows the user to highlight and reveal potential chemical motifs in a very easy and intuitive fashion. The analysis is based on atoms/residues pair distances applying a modified version of Dijkstra's algorithm, and it makes possible to monitor the distances of a large pathway, even during a molecular dynamics simulation.

Article II

“Chem-Path-Tracker: An automated tool to analyze chemical motifs in molecular structures “

Adapted from:

Ribeiro J., Cerqueira N.M.F.S.A., Fernandes P.A., Ramos M. J.
2013 (submitted).

3.2.1. Chem-Path-Tracker - An automated tool to analyze chemical motifs in molecular structures.

J. V. Ribeiro, N.M.F.S.A. Cerqueira, P.A. Fernandes, and M.J Ramos

REQUIMTE , Departamento de Química e Bioquímica, Faculdade de Ciências, Universidade do Porto, Rua Campo Alegre s/n , 4169-007 Porto, Portugal.

3.2.1.1. Abstract

In this article we propose a method for locating functional relevant chemical motifs in protein structures. The chemical motifs can be a small group of residues or structure protein fragments with highly conserved properties that have important biological functions. However, the detection of chemical motifs is rather difficult because they often consist of a set of amino acid residues separated by long, variable regions, and they only come together to form a functional group when the protein is folded into its three dimensional structure. Furthermore, the assemblage of these residues is often dependent on non-covalent interactions among the constituent amino acids that are difficult to detect or visualize. To simplify the analysis of these chemical motifs and give access to a generalized use for all users, we developed Chem-Path-Tracker. This software is a VMD plug-in that allows the user to highlight and reveal potential chemical motifs requiring only a few selections. The analysis is based on atoms/residues pair distances applying a modified version of Dijkstra's algorithm, and it makes possible to monitor the distances of a large pathway, even during a molecular dynamics simulation. This tool turned out to be very useful, fast and user-friendly in the performed tests. The Chem-Path-Tracker package is distributed as an independent platform, and can be found at <http://www2.fc.up.pt/PortoBioComp/database/doku.php?id=chem-path-tracker>.

Keywords: Protein networks, Dijkstra's algorithm, motifs, VMD

3.2.1.2. Introduction

The development of new and effective drugs is strongly dependent on the identification of new drug targets with reduced side effects and better efficiency. Resolving this issue depends partially on a thorough understanding of the biological function of proteins. Unfortunately, the experimental determination of protein function, when possible, is expensive and time consuming. To support and accelerate this

endeavor, one of the major challenges in bioinformatics is to develop robust methods capable of interpreting and extracting important data concerning molecular structures that enclose specific and conserved features, such as chemical motifs, from which the chemical/biological function depends on.

Generally speaking, chemical motifs are just like electronic circuits that are built from simple boolean gates and are required to perform specific functions. In a chemical environment, chemical motifs can be a small set of residues or structure protein fragments with highly conserved properties that perform some important biological function. For instance, these conserved motifs may be a cluster of amino acid residues that are required to maintain a metal ion bound to a specific region of a protein or even to allow protein-protein interactions. They can also be a large group of residues involved in long range proton/electron transfers or even determinant to the three dimensional structure of the protein⁸³.

The detection of chemical motifs is however rather difficult because they often consist of single conserved amino acid residues separated by long, variable regions, and they only come together to form a functional group when the protein is folded into its three dimensional structure. Furthermore, the assemblage of these residues is often dependent on non-covalent interactions among the constituent amino acids that are difficult to detect or visualize. Additionally, it is becoming increasingly clear that the interactions between amino acids at the global level are the deterministic factor of the chemical motifs, and an investigation involving pairwise interactions alone is not sufficient to understand the basis of the uniqueness of these chemical motifs⁸⁴.

Chemical motifs are generally found deeply buried in the protein structure. This allows them to maintain their proper spatial configuration that is fundamental for their specific function and to be protected from the solvent or other molecules that might interfere with their function. In this regard, many authors suggest that chemical motifs can be seen as independent regions in the protein structure that are surrounded by shielding residues that protect their chemical identity and at the same time ensure their correct location and orientation⁸⁵.

Based on these assumptions we developed a new bioinformatics tool, called Chem-Path-Tracker, for locating functionally relevant motifs in protein structures by means of protein structured networks. These networks are constructed at a coarse-grain level, considering atoms or residues as node points and are generated based on

a starting point that is pre-selected by the user and a cut-off distance that will determine the maximum distance between two consecutive nodes. The final results will generate several pathways (networks) that interconnect the nodes that fulfill the requirement of the selected conditions. For instance, if these requirements are set to search for specific interactions, such as hydrogen bonds, then all the possible hydrogen bonds between the selected nodes will be visualized.

Chem-Path-Tracker includes a graphical interface that has been embedded in VMD and allows an easy and simple preparation of the input files as well as the analysis of the output files. The software works with any structure that can be identified by the VMD software. This means that it can analyze a wide range of static structures such as a PDB file retrieved from the protein databank or even multi-frame structures such as molecular dynamics (MD) simulations performed with different MD software (amber, namd, gromacs, etc). In the latter case, the software provides a statistical treatment of the values and sorts the best pathways based on the data that was collected. The software can be found at: <http://www2.fc.up.pt/PortoBioComp/database/doku.php?id=chem-path-tracker>.

In the next sections it is described the algorithm, the graphical interface, as well as several examples that demonstrate the potential of Chem-Path-Tracker.

3.2.1.3. Software Description

3.2.1.3.1. Graphical Interface

Chem-Path-Tracker was developed in Tcl/Tk and was included in VMD as a plug-in. This allows the program to explore all the powerful graphical backend of VMD as well as all the facilities provided by this software, such as reading several file formats including several types of multi-frame structures like the ones generated by molecular dynamics. The program can be directly integrated in the extensions/analysis menu of VMD from where it opens as a graphical user interface (GUI) (Figure 8).

Once the input files are set up, all calculations are performed by VMD, without the use of external programs. Therefore, our plug-in just requires a structure or a trajectory file to be loaded into VMD. Furthermore, all calculations that are produced by Chem-Path-Tracker can be analyzed by the plug-in itself or can be written in an

appropriate txt file format, which allows the exchange of data with other applications, such as spreadsheets.

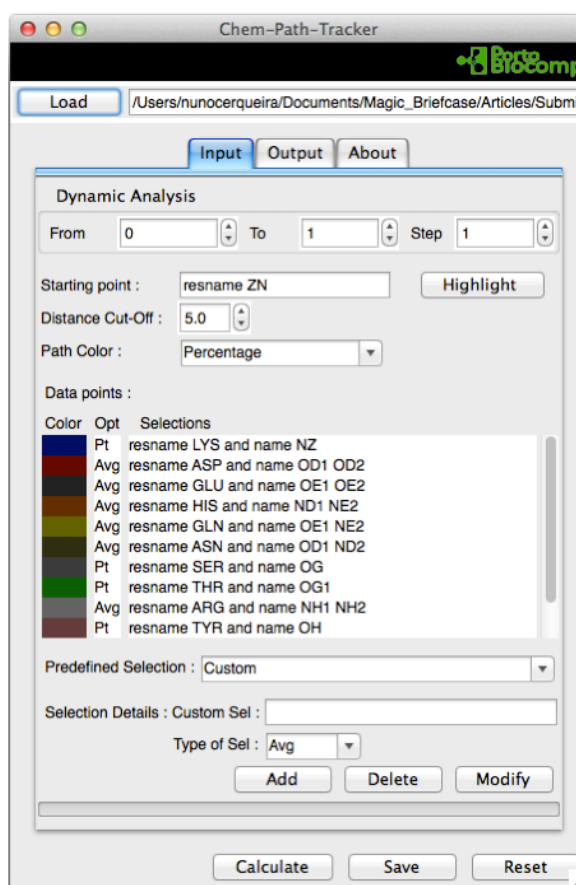


Figure 8: Chem-Path-Tracker's Graphical User Interface (GUI) input tab. In this tab the user can set the parameters and the selection to be applied in the pathways' search..

3.2.1.3.2. A modified Dijkstra's algorithm

The main algorithm behind Chem-Path-Tracker is the Dijkstra's algorithm^{8,9}. This algorithm starts at a pre-determined point (here called the starting point) and tries to connect all the other points (here called nodes) in such a way that each point is visited just one time. The distance between all the node points is then evaluated in such a way that the shortest path between two consecutive points is always chosen. The algorithm requires therefore the selection of a starting point and a set of nodes that the user wants to link to the starting point.

In Chem-Path-Tracker, the starting point of Dijkstra's algorithm can consist of any point of the three-dimensional space of the biological system under study, available

on the VMD graphical interface, which the user can choose using the selection tools available. *E.g.* the VMD command “index 100” will select the point in space that is occupied by the atom with that index. When more than one atom is selected, the program always calculates the geometric center of that selection and uses it as the starting point. For example, if the user selects residue 120 as a starting point, the program will take into account the position of the atoms that are part of residue 120, and will calculate their geometric center. This point will then be used as the starting point. When the starting point is selected it will be displayed in the graphical interface as a yellow sphere.

The nodes can be defined in Chem-Path-Tracker using the same protocol that was described before. These points should identify key features in the structure of the object that the user wants to analyze and link them with the starting point. *E.g.* if the user wants to search for all possible hydrogen bonds in a protein structure, then he needs to select the atom types that are good electron donors and those that are good proton donors. However, this type of selection can become time consuming and therefore less efficient. In order to overcome this issue, we added two options for the selection process: PT (points) and AVG (average). When the user selects the PT option, the atoms included in the selection will be treated as individual points in the search algorithm. This option is very useful if the user wants to select a certain atom type. For instance in the example given above, if the user wants to select all atoms that are electron donors, he/she could activate option PT and select all the atoms of the protein whose atom type possesses the relevant characteristics.

When the AVG option is used, the software will calculate the geometric center of each residue present in the selection, and only those points will be included in the algorithm. For instance, if the user selects the AVG option and the atom selection includes all tyrosine residues present in the protein, only the geometric center of all tyrosine residues in the protein will be included in the algorithm. However, in this example, if the user chooses to select the PT option instead, all atoms of the tyrosine residues available in the protein will be explicitly included in the algorithm.

Once the starting point and the nodes are defined, the software is ready to run. In Chem-Path-Tracker, we have changed the original Dijkstra’s algorithm in order to make it more useful in a molecular framework (Figure 9). The software will initiate at the starting point and will begin the search for the closest “node point”. There is a cut-

off value that limits the distance that is used to search for new node points from the starting point. This means that this variable should be used to control the maximum distance that separates two consecutive nodes, which can be correlated with the type of interaction that we are seeking. *E.g.* if the user is searching for hydrogen bonds in a protein structure, the cut-off can be defined as 3Å between heavy atoms. This will ensure that the distance between each node in the final pathway will be always below 3Å. The many nodes that do not fulfill such requirement are discarded and the final pathway will not include them. However, in some cases, more than one node will fulfill the requirement imposed by the cut-off distance. Here, the nodes with the closest distance from the starting point will be chosen and the process will be repeated until all possible nodes are linked to each other. This will create the main pathway. All points that fulfill the cut-off distance in relation to the same starting point, are used to create ramifications of the main pathway. This will lead to the formation of secondary pathways that connect a starting point to different nodes. When a ramification occurs, the software ensures that the new pathways are unique and do not share any of the nodes from other pathways, except those that are common until the ramification starts. The number of pathways and their end points are dependent on the cut-off distance that is chosen. The original Dijkstra's algorithm can be activated when the cut-off distance is increased to infinity.

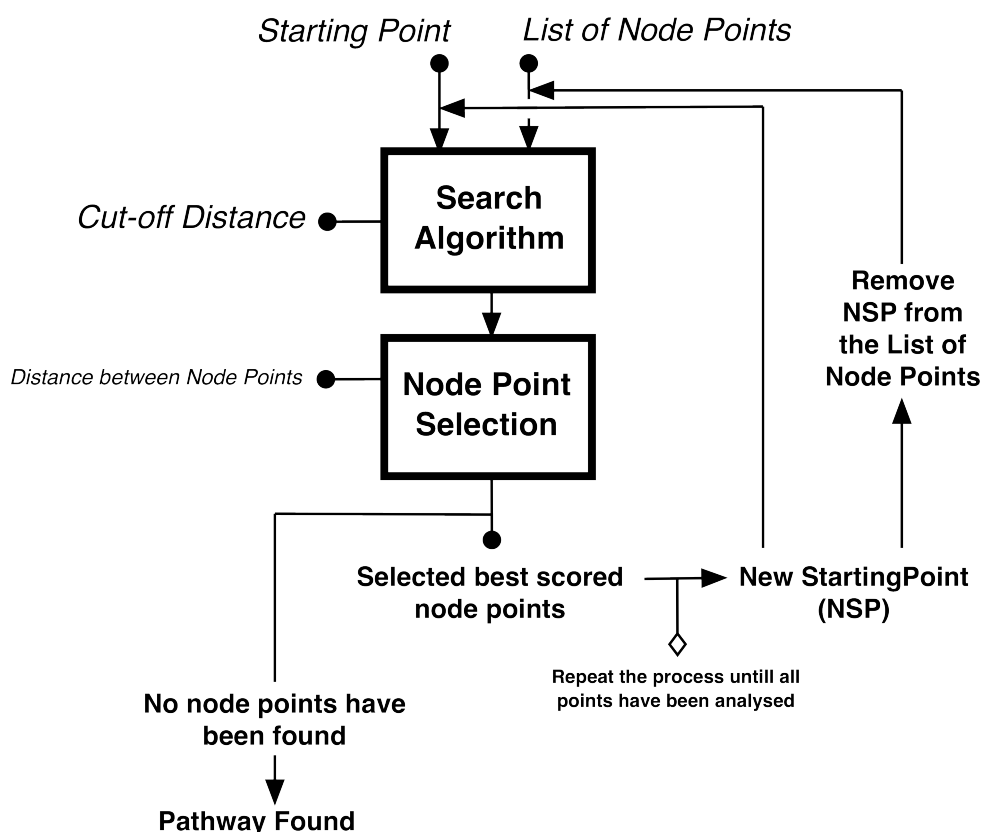


Figure 9: General algorithm implemented in Chem-Path-Tracker software.

3.2.1.3.3. The output of Chem-Path-Tracker

Once the calculation is complete, Chem-Path-Tracker stores all the results in a text file with a bpf extension. This file stores all the information generated in the form of a table, and each line of this table contains the pair of nodes that are part of the pathways generated. If the system that is studied is a protein, each node will be presented in this table by the residue name and the respective identification. This table will provide also the score of each pair of nodes given by the scoring function. In the case of multi-frame structures, such as the trajectories provided by molecular dynamics simulations, Chem-Path-Tracker will provide additionally a statistical overview of the score of each pair of nodes during the selected frames, *i.e.* the average value and the standard deviation.

Point1	Res1	Point2	Res2	DistMean(Å)	STDV(Å)
52	ASP455A	187	SER454A	4.69	0.00
1	LAN1740A	351	CYS533A	4.15	0.00
334	TYR587A	362	HOH2814A	2.59	0.00
92	GLU532A	362	HOH2814A	4.71	0.00
92	GLU532A	191	SER580A	4.35	0.00
337	TYR704A	338	TYR707A	4.80	0.00
334	TYR587A	338	TYR707A	4.19	0.00
1	LAN1740A	361	HOH2679A	3.83	0.00
1	LAN1740A	362	HOH2814A	2.83	0.00
312	TYR98A	337	TYR704A	2.92	0.00
1	LAN1740A	52	ASP455A	2.91	0.00
52	ASP455A	348	CYS456A	4.77	0.00

Figure 10: Chem-Path-Tracker's Graphical User Interface (GUI) output tab. This tab presents the final results of the PCET pathways' search.

In order to turn the analysis of the results more clear, all this information can be easily controlled in the output tab of Chem-Path-Tracker. Here, each pair of nodes is represented as balls and the connection between them as sticks. The color of the nodes is the same of that in the selection done for the input, in order to facilitate their identification. The color of the lines linking two nodes can be chosen by the user. If a multi-frame structure is selected, the color of the path is a function of their percentage of occurrence, in a scale from blue to red as the percentage rises from 0 to 100 %.

As it was outlined before, Chem-Path-Tracker can generate more than one pathway starting from the initially selected starting point. Some times it is useful to analyze the shortest scored pathway that connects two nodes. Taking this into account we have implemented such facility in the output table of Chem-Path-Tracker in order to carry out such type of analysis. Here, the user just has to select two points

that are present on all the paths that were analyzed by Chem-Path-Tracker. Once this is complete, the programs will automatically give the best-scored pathway that links the two points based on the information collected previously. If the user wants to analyze more than one pathway, he can color each pathway with different colors, facilitating thus each selection (see Figure 10).

3.2.1.4. Software Validation

In order to evaluate the applicability of the Chem-Path-Tracker software, we looked for special chemical motifs in several protein structures that are difficult to analyze even by experienced users. For this purpose we selected five protein structures: guanine deaminase, oxidosqualene cyclase, human growth hormone/receptor complex, aquaporin, and ribonucleotide reductase.

Proteins guanine deaminase and oxidosqualene cyclase were used to demonstrate the capability of Chem-Path-Tracker to identify a network of residues around the active site. In the human growth hormone/receptor complex we tried to locate an unusual chemical motif, which involves a cluster of 9 residues that interact via cation- π interactions. The aquaporin structure was used to identify the residues that interact more closely with a tunnel of water molecules located inside the protein structure and that have an important biological role. The enzyme ribonucleotide reductase (RNR) was used to demonstrate the capabilities of Chem-Path-Tracker to pursue residues that are involved in proton or/and electron transfer across the protein structure. In this case Chem-Path-Tracker was used in a multi-framed structure involving 20 ns of a molecular dynamics simulation.

3.2.1.4.1. Chemical motifs surrounding active sites

Some of the most interesting chemical motifs that have been conserved throughout the evolution of proteins are the active sites. These regions determine the function of proteins and are generally found in clefts or pockets within the protein structure. Structurally, an active site is lined up by a set of amino acid residues that participate directly or indirectly in a chemical reaction. These residues are specifically aligned and oriented in the protein structure and are modeled by a set of

molecular interactions that turn that site unique both from structural and chemical points of view.

Generally, only a small set of residues of the active site is analyzed either because those are the ones that are directly involved in the catalytic process or because they have somehow been shown to be important for the reaction. The surrounding residues are most of the time disregarded, although several studies have shown that these residues are equally important for the function of the enzyme, as they maintain the reactive residues close to each other and they can also change the chemistry of neighbor residues⁸⁶⁻⁹¹.

In this sense, Chem-Path-Tracker can be an extremely good tool to explore the networks of molecular interactions in the active site region, and highlight the neighboring residues that might interact with it through medium and long-range interactions.

To exemplify the potential associated with Chem-Path-Tracker we have analyzed the active sites of 2 enzymes: guanine deaminase and oxidosqualene cyclase.

3.2.1.4.2. Guanine deaminase

The enzymes containing metal ions in their active sites are very interesting to analyze in this regard, because the protein scaffold has to maintain the ion bound to the protein structure and simultaneously assist the substrate binding and the catalytic process. This often leads to the formation of specific chemical motifs that are unique and determinant for the protein function⁹². The enzyme guanine deaminase is one of these cases with a zinc ion surrounded by a cluster of histidine residues in its active site⁹³. The X-ray structure with the bound product (pdb code: 2UZ9) also reveals that the metal ion interacts very closely with the substrate and therefore it participates directly in the catalytic process.

In order to exemplify some of the capabilities of Chem-Path-Tracker we have explored the structure of the active site of this enzyme. To achieve this, we started by selecting the zinc ion as the starting point (index 3546). The node points were generated based on one of the preconfigured option in Chem-Path-Tracker GUI, *i.e.* node points for hydrogen bonds (HBonds) and the substrate (resname XAN). The chosen cut-off distance of the search algorithm was set to 5.0 Å.

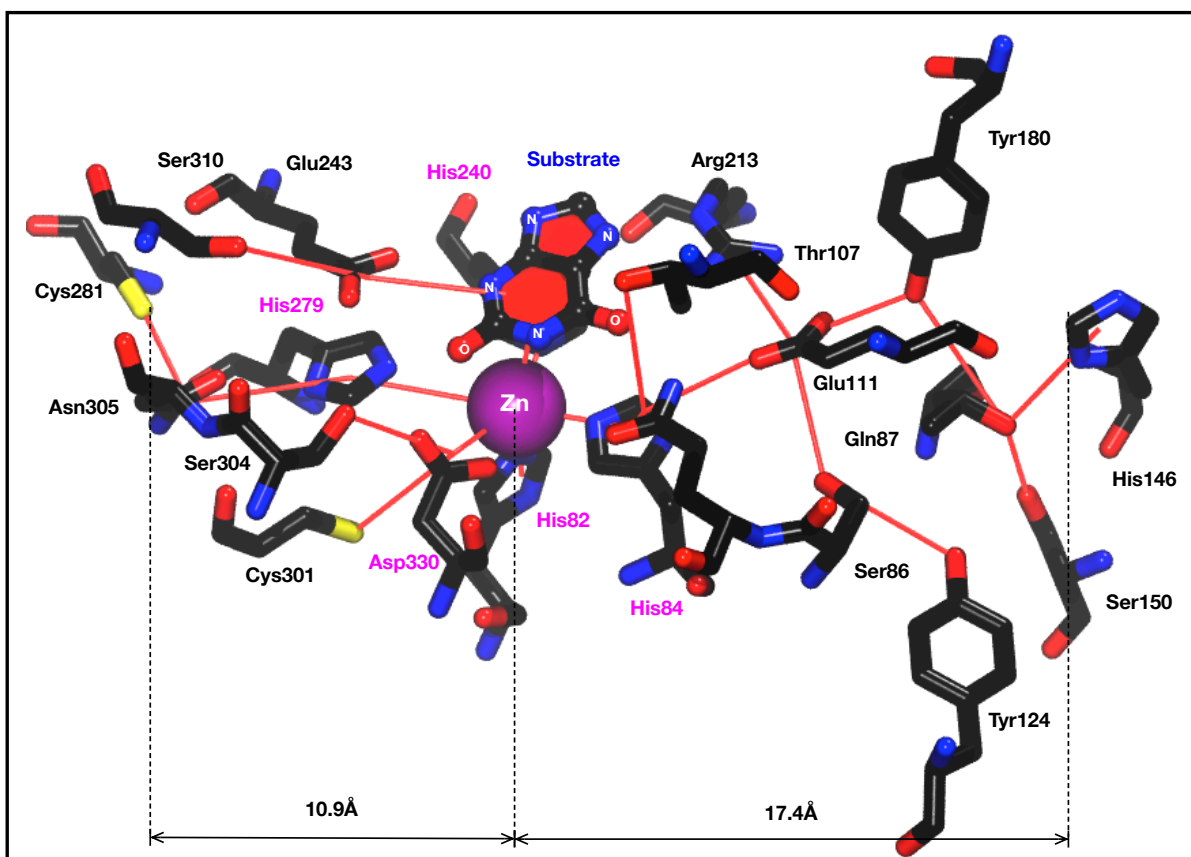


Figure 11: Chemical interaction between the residues of the active site of the enzyme guanine deaminase that were highlighted by the Chem-Path-Tracker software. In pink are represented the amino acid residues that are located at the first coordination sphere of the zinc ion.

The graphical representation generated by Chem-Path-Tracker reveals a network of interactions in which the zinc ion plays a central role. The amino acid residues that interact more closely with the zinc ion involve an aspartate (Asp330) and a cluster of histidine residues (His82, His84, His240 and His279). These residues are located at the bottom region of the active site and are thought to be crucial for the stabilization of the metal in the protein structure as well as conferring a certain chemical reactivity. Chem-Path-Tracker was also capable of identifying several amino acid residues that do not interact directly with the zinc ion but stabilize the residues that are located in the first coordination shell. For example, Asp330 interacts directly with the zinc ion and its position is strongly stabilized by two short hydrogen bonds provided by neighbors Cys301 and Ser304. The same is also true for His84 that is stabilized by Glu111 and the latter by Gln87, Tyr180 and Thr107. These results suggest that the structure and perhaps the chemistry of the active site of

guanine deaminase is supported by a network of interactions that are not confined to the residues that are located in the first coordination sphere of the zinc ion. Some residues in this network of interactions can be found at 17 Å and 11 Å away from the zinc ion as it is the case of His146 and Cys281, respectively. These results suggest that these residues can also be equally important for the protein function together with some of the residues that are located in the first coordination shell of the zinc ion. Of course that this requires further calculations and such analysis is out of the scope of Chem-Path-Tracker.

3.2.1.4.3. Oxidosqualene cyclase

Another interesting example involves oxidosqualene cyclase (OSC), an enzyme involved in the biosynthesis of cholesterol. Cholesterol is a bulky lipid molecule, composed of four linked rings of carbon decorated with hydrogen atoms and a single oxygen atom. A collection of two dozen enzymes is needed to build cholesterol from simple starting compounds. Enzyme OSC performs the most complicated step in this process. It takes a long thin carbon chain, oxidosqualene, and folds it up, creating the four linked rings. In order to explore the active site of this enzyme we analyzed the X-ray structure that contains the product of the reaction (PDB entry 1w6k)⁹⁴.

Since the reaction starts with the protonation of an epoxide, we selected the oxygen that is involved in that reaction, as a starting point (index 5909). We then selected all the residues that can interact with the starting point by hydrogen bonds (automatic pre-selection HBonds). Since the active site is located near the protein surface and it is highly exposed to the solvent, we have also selected the water molecules that populate the active site and may be involved in the catalytic process. However, due to the high number of water molecules that surround the active site we have only selected those that are in the range of 5.0 Å from the substrate (we use the following command in VMD to get this selection: “water within 5 of index 5909”). The chosen cut-off distance was set to 4.8 Å. The results provided by Chem-Path-Tracker are illustrated in Figure 12.

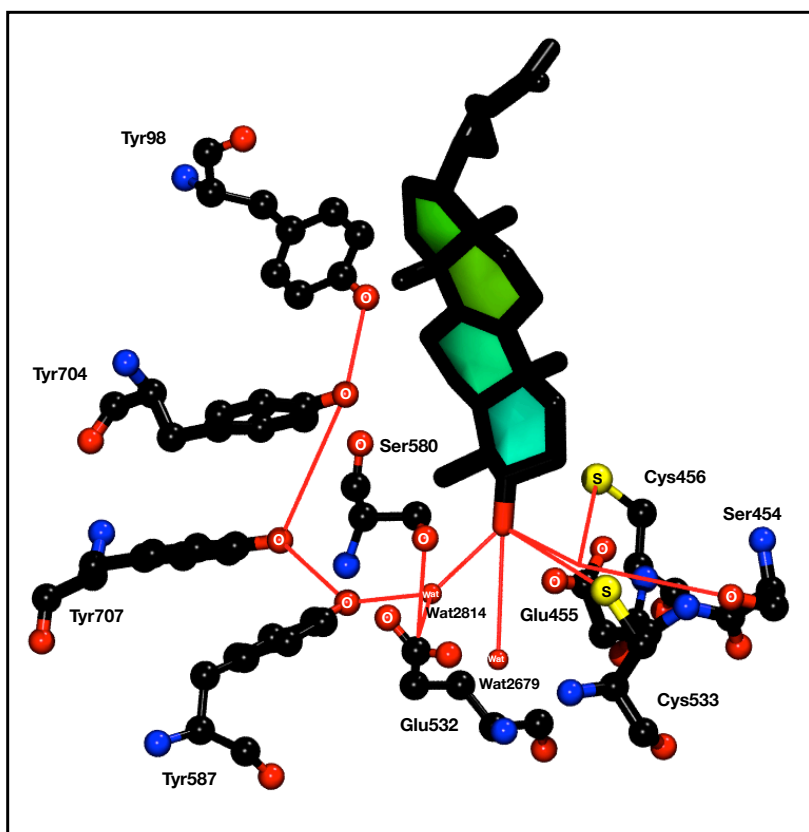


Figure 12: Most important chemical interactions present in the active site of OSC.

Chem-Path-Tracker was able to show that the oxygen atom that is involved in the epoxide protonation is in close contact with Glu455 - a residue that has been shown to be essential for the enzyme function^{95,96}. It has also shown that Glu455 is stabilized by two hydrogen bonds provided by one cysteine residue (Cys456) and one serine residue (Ser454). Chem-Path-Tracker has also revealed that near the oxygen are located two water molecules (Wat2814 and Wat2679). The position of Wat2814 is strongly stabilized by Tyr587 and by Glu532. Glu532 is further stabilized by Ser580. Interestingly, Tyr587 is part of a sequence of 4 tyrosine residues (including Tyr707, Tyr704 and Tyr704) that are aligned across the region of the active site and interacts very closely with the substrate. These residues are believed to be important for the stabilization of the carbocations that are generated through the formation of the four rings of lanosterol^{97,98}.

The result provided by Chem-Path-Tracker demonstrates that the active site of OSC is composed by several residues, which have been shown in the literature to be important in different steps of the catalytic process. These residues interact with each other through a network of hydrogen bonds and this maybe the key that allows the

concomitant formation of the four rings that is catalyzed by this enzyme and involves an uncommon carbocation chemistry.

3.2.1.4.4. Identification of Cation- π interactions

Chem-Path-Tracker can also be extremely useful when seeking non-covalent interactions, such as cation- π interaction. This type of interactions has been relatively unappreciated when compared to the more conventional interactions such as hydrogen bonds, ion pairs and the dispersive interactions, but recent studies have shown that they play an important role in molecular biology. In proteins, cation- π interactions occur between π electron-rich residues such as tyrosine, phenylalanine or tryptophan and an adjacent cationic atom or residue, such as lysine or arginine. These interactions are known to contribute to the stability of protein native states, as well as protein ligand complexes, or even DNA/RNA complexes but their structure and function remains poorly understood⁹⁹.

An interesting example of a continuous (cation- π) $_n$ stack composed of a large set of residues is observed in the X-ray structure of human growth hormone/receptor complex (pdb code 3HHR)¹⁰⁰. The identification and visualization of these structures is however difficult to make because the protein contains thousands of atoms and this type of structure only occurs in a specific region of the protein. In order to turn this process easier we have used Chem-Path-Tracker to visualize them as well as to evaluate the nature of the surrounding residues; the user just has to go through a couple of steps. The first step involves the selection of a starting point. In this case the user has to know one residue that is involved in the cation- π interaction network. In this case we selected one atom of Phe225 (index 4524). The next step involves the selection of all the potential amino acids around Phe225 that might be involved in the (cation- π) $_n$ stack chain - we selected all the amino acids that contain rings in their structure and all the cationic residues. For this purpose we have inserted two custom selections in the Chem-Path-Tracker Data points list: for the first selection we used "(resname HIS HIE HID PRO PHE TYR TRP) and not backbone)" and for the second selection "(resname ARG LYS GLU GLN) and not backbone)". We have chosen a cutoff distance of 4.5Å. The final result provided by Chem-Path-Tracker is shown in Figure 13.

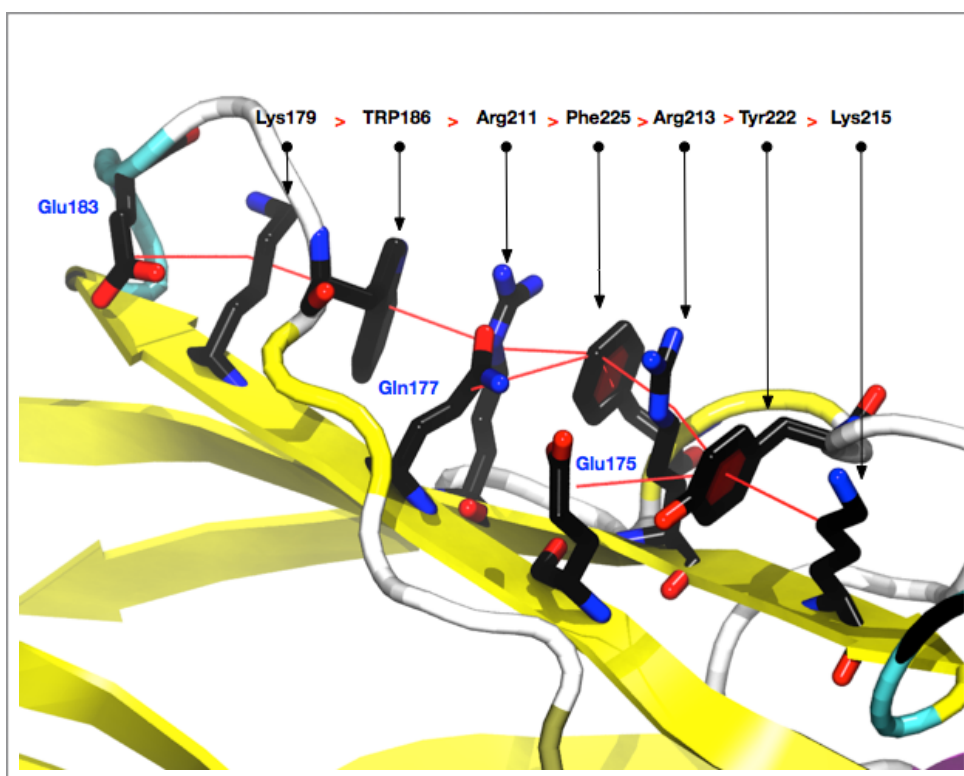


Figure 13: Graphical pathway that was highlighted by Chem-Path-Tracker revealing the (cation- π) $_n$ stack chain found on the PDB structure with the code 3HHR.

Chem-Path-Tracker has sketched a straight line across the protein structure that connected several node points. These node points are part of an interesting (cation- π) $_n$ stack chain formed by Lys179-Tyr186-Arg211-Phe225-Arg213-Tyr222-Lys215, in which the positive charged residues are intercalated with aromatic containing residues.

3.2.1.4.5. Exploring Water Channels

Aquaporins are proteins embedded in the cell membrane of biological cells that form pores and regulate the flow of water. These proteins are very important because they have specific characteristics that allow them to be impermeable to proton permeation despite their very fast water conduction. The available X-ray structures of aquaporins reveal that in the middle of these proteins there are channels that are populated with water molecules. These channels are very narrow and constrain the access of water molecules, only allowing one water molecule to pass through the

channel. Interestingly, this could lead to the idea that this net of water molecules could easily act as a proton wire to conduct protons but this cannot occur, as it would damage the membrane¹⁰¹. The amino acid residues lining the constriction of these water channels are mainly hydrophobic with the exception of some polar amino acids. Recent results have shown that the role of these residues is very important because they help to line up and orient the water molecules in such a way that prevents the proton transfer.

In this context Chem-Path-Tracker can be extremely useful not only to highlight the region that is occupied by the water molecules and therefore to highlight the water channel, but also to identify which residues interact more closely with these molecules and preclude the proton transfer. In this quest we analyzed the pdb structure 2zz9¹⁰².

In order to run Chem-Path-Tracker, the user has to start by selecting a starting point. In this case we selected the index of any water molecule located in the water channel (for instance index 1858). Subsequently, the user has to select which type of residues he wants to search for and that remains in close contact with that starting point. In this case we start by selecting the water molecules that are present in the X-ray structure in order to visualize the form of the water channel (this can be done inserting the custom selection: `resname HOH`). Additionally, we will select all the amino acids of the protein in order to identify which residues are in close contact with these water molecules (this can be done with the `HBonds` automatic selection). These selections will create a list of points (nodes) in the graphical interface. The last step that is required to start Chem-Path-Tracker is a cut-off distance that will be used by the software to constrain the search distance between the node points (in this case we used a cut-off distance of 4Å.). The final result can be seen in Figure 14.

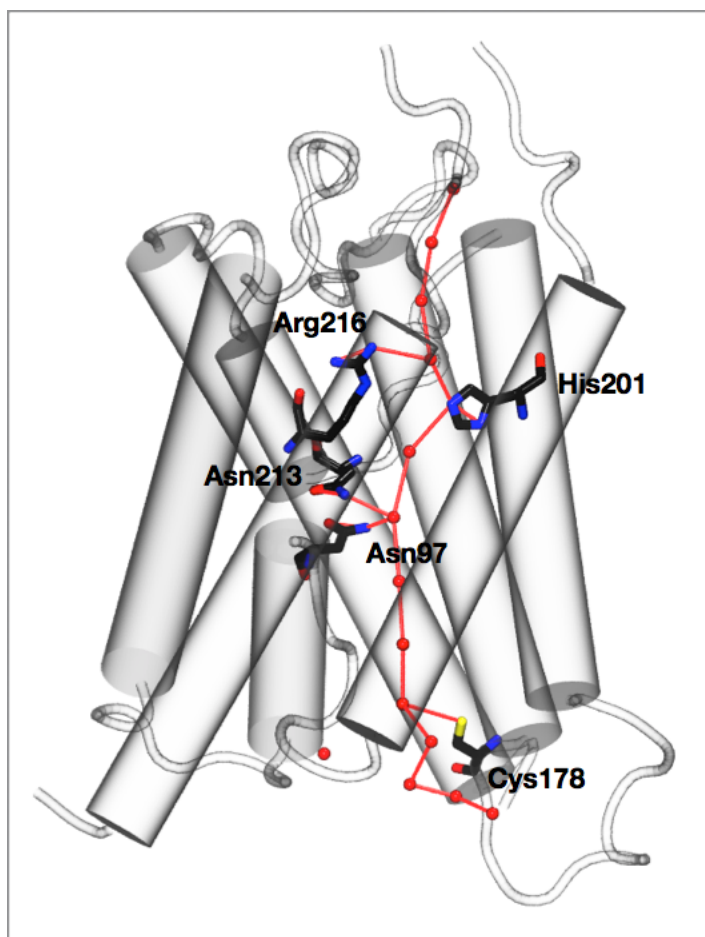


Figure 14: Graphical pathway that was highlighted by Chem-Path-Tracker on the PDB structure with the code 2zz9.

In Figure 14, we can see that Chem-Path-Tracker was able to draw a pathway in the middle of the protein that corresponds to its channel. Analyzing the output file of Chem-Path-Tracker, we found that the majority of the residues that compose this pathway are water molecules, but a small set of them are polar residues, namely two asparagine residues (Asn213 and Asn97), an arginine (Arg216) and a histidine (His201) residue.

From the literature it is proposed that the residues that were highlighted by Chem-Path-Tracker are highly conserved in the aquaporins and very important to prevent the proton conduction¹⁰². Indeed, it is proposed that the position and orientation of these residues allows them to interact through short hydrogen bonds with the nearby water molecules¹⁰³. This effect is proposed to have two consequences: i) the transient hydrogen bonds reduce the energy barrier for water molecules coming into the middle of the channel, and ii) it forces the reorientation of the water molecules. In the former case, it forces the hydrogen atoms to become oriented perpendicular to

the channel axis, thus excluding any contact with the adjacent water molecules. Consequently, the central water molecule becomes isolated from the other water molecules in the channel. As a result, the water molecules no longer act as a proton wire, and the channel no longer conducts protons.

3.2.1.4.6. Proton and electron transfers

Reactions involving proton and/or electron transfers have always attracted considerable attention, mostly because of their wide distribution in many areas of biology and chemistry.

One of these interesting and challenging cases involves the enzyme ribonucleotide reductase (RNR). RNR is an extraordinary enzyme that catalyzes the conversion of the nucleotides into their corresponding 2'-deoxynucleotides. This enzyme is composed by two homodimeric subunits, named R1₂ and R2₂ that have to dimerize in order to turn the enzyme active. Each R1 monomer lodges an active site that is responsible for the enzymatic activity, whereas each R2 monomer contains a diiron-tyrosyl radical cofactor that is deeply buried in the protein surface. Each cofactor is responsible for the generation of a radical that has to migrate to the active site that is located almost 35Å away in monomer R1. This subject has been under intense research in the last three decades but the atomistic insight of how it happens remains unsolved because the X-ray structure of the active dimer is still unknown¹⁰⁴.

In the last 2 decades several site mutagenic experiments on monomer R2 have shed some light on this issue and identified several residues that participate in the proton and/or electron transfer on subunit R2, namely Y122, H118, W48 and Y356^{105,106}. This pathway has been shown to be dependent on a network of residues that interact with each other very closely starting at the metal cluster and ending at the protein surface of monomer R2. Taking this into account we have tested Chem-Path-Tracker to see if it could disclose the network of residues that are involved in such proton and/or electron transfer of protein R2 and then correlated the final result with the data that is available in the literature.

In this case we opted to analyze a multi-framed structure of subunit R2 of RNR, instead of static X-Ray structures as in the previous examples. Therefore, we first performed a classical atomistic molecular dynamics simulation of the full R2₂ subunit,

using the AMBER9 molecular dynamics package¹⁰⁷. The protein residues were described using the parm99 force field and the iron ions of the metal cluster were described as Lennard-Jones spheres with a charge of +2. In all simulations, SHAKE constraints were added to bonds involving hydrogen atoms, allowing for a 2 fs timestep. The temperature was kept constant using a Langevin thermostat with coupling parameter of 1 ps⁻¹.

In order to disclose the network of residues that are involved in the proton and/or electron transfer at protein R2, we selected in the Chem-Path-Tracker GUI one of the iron ions present in the metal cluster as the starting point. We then selected all the residues that could be involved in a network of hydrogen bonds as well as the residues that possess rings structures. In this case we chose a cut-off distance of 5.2Å and set Chem-Path-Tracker to analyse all the snapshots of the molecular dynamics run in steps of 10 frames.

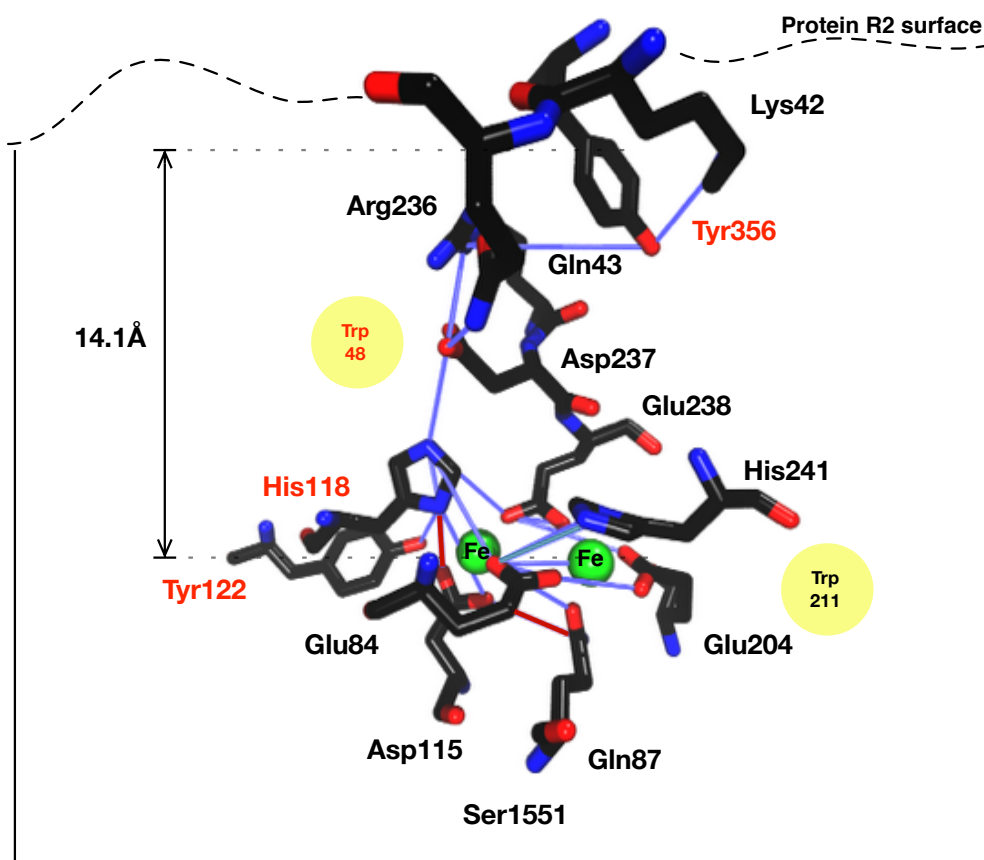


Figure 15: Interaction network between the residues of monomer R2 of RNR provided by Chem-Path-Tracker (for the clarity of the image, Tryptophan 48 and 211 were not represented). The residues labeled in red were experimentally identified by site-directed mutagenesis to be crucial for enzymatic activity (the nomenclature of the residues was based on the X-ray structure 1XIK).

The results provided by Chem-Path-Tracker are shown in Figure 15. In this case the pathway drawn by the software has several colors. These colors indicate the tendency of a pathway to occur during the molecular dynamics simulation. This is calculated based on the average of the residues pair distances and the respective standard deviations, which were calculated in each frame that was analyzed in the molecular dynamics simulation.

Despite the great simplicity and speed of Chem-Path-Tracker, the algorithm was able to identify in the protein structure a network of interactions that start at the metal cluster and end at the protein surface. These networks reveal that the di-iron metal cluster is surrounded by a group of negatively charged residues, namely aspartate Asp155 and three glutamate residues Glu84, Glu204 and Glu238, as well as two histidine residues (His118 and His241) a glutamine residue (Gln85) and a tyrosine residue (Tyr122). These residues should be important to maintain the metals bound to the protein scaffold and at the same time provide the chemical environment required for the formation of the radical that is required for the catalytic process. Additionally, Chem-Path-Tracker has also highlighted a group of residues that interconnect this region with the protein surface. This pathway is composed by Asp237, Arg238, Tyr358 and Lys42. This group of residues should therefore be implicated in the migration of the radical from the metal complex to the surface of the protein structure. Gln43 and Trp48 were also identified by Chem-Path-Tracker but their function seems to be more involved in stabilizing the position of Asp237, Arg238 or perhaps in stabilizing the migration of the radical through these residues.

Comparing the residues that were marked by the Chem-Path-Tracker algorithm and the residues that are known experimentally¹⁰⁵ to be involved in the proton electron transfer (residues label in red in Figure 8), we can see that there is a very good match. Indeed, Chem-Path-Tracker was able to highlight all the residues that were experimentally observed by site directed mutagenesis to be involved in the proton electron transfer (residues marked with an asterisk in Figure 8). This example demonstrates that Chem-Path-Tracker can be a very handy tool to explore the protein structure and can be used to help experimentalists to foresee possible residues that might be important for protein function.

3.2.1.5. Conclusions

Chem-Path-Tracker was developed to provide a fast approach to identify possible chemical motifs in any type of system containing thousands to millions of atoms. These structures are often difficult to reveal by the non-expert user or hard to characterized to its full extent because they are a small part of the protein structure that are not sequential as general alfa-helices or beta-sheets and they only come together when the protein is folded in its tri-dimensional structure.

The algorithm used in Chem-Path-Tracker relies on a modified version of Dijkstra's algorithm that searches for the closest path between a pre-selected set of node points that are pin-pointed by the user. These node points can be amino acids, atoms types, or any type of selection that can be identified by VMD. The algorithm starts from a pre-selected point chosen by the user and uses a cut-off distance to joint that point with all the other node points. In the course of the process this will generate a pathway of points. The node points are included or not in the pathway depending on the cut-off distance, which means that not all the node points will be part of the pathway. If more than one node point fulfills the condition imposed by the cut-off distance, the Chem-Path-Tracker algorithm will generate n-pathways and guarantees that each pathway is unique.

The final result provided by Chem-Path-Tracker will always be dependent on the conditions that are imposed by the user and therefore they must have a chemical meaning. To help the user, the program comes with a pre-defined condition that can be used to search for molecular motifs involving hydrogen bonds, ionic bonds and cation- π interactions.

The Chem-Path-Tracker software can be used to analyze static structures, as it is the case of the X-ray structures deposited in the protein databank (pdb structure), or any other type of structure that can be read by VMD. It can also be used to analyze multi-framed structures, such as molecular dynamics simulations whose format can be analyzed by VMD.

The application of Chem-Path-Tracker to the structure of several proteins has revealed to be extremely useful. It allows the identification and characterization of the structure of active sites, pathways of residues involved in proton and/or electrons transfers, or even to look for specific residue interactions, as it is the case of cation- π

interactions. Some of these results are capable of highlighting key amino acids that were identified experimentally in the past to be crucial for the protein function. Based on these results, we believe that Chem-Path-Tracker can be a very handy tool to explore the structure of proteins and speed up the deciphering of chemical motifs in biological systems.

3.3. VolArea

Protein structures can determine the type and strength of their interactions with other compounds, either small molecules or big proteins. An example that can strongly influence the interaction between proteins is the accessibility of certain residues from one of the proteins to the other/s protein/s, increasing the probabilities of strong interactions between them. Besides the different surfaces that VolArea can evaluate, such as the solvent accessible surface or surface accessible to other compounds, the volume of the structures or empty volumes such as cavities and clefts can also reveal important information, for instance, the space available to the interaction and the behavior of the surrounding environment of the binding site when a trajectory is analyzed.

In order to provide the user with the tool necessary to calculate molecular surface areas and volumes, we have developed a computer program named VolArea, a VMD plug-in that presents a very intuitive Graphical User Interface, supporting the user in the preparation of the values necessary to the calculation, displaying the results in a very simple way and evaluating standard deviations and averages in the cases of a multi-frame analysis.

Article III

“VolArea - A Bioinformatics tool to calculate the surface area and the volume of molecular systems“

Adapted from:

Ribeiro J., Cerqueira N.M.F.S.A., Fernandes P.A., Ramos M. J.
Chem Biol Drug Des 82:743–755, 2013.

3.3.1. VolArea - A Bioinformatics tool to calculate the surface area and the volume of molecular systems

*J. V. Ribeiro, J. A.C. Tamames, N. M.F.S.A. Cerqueira, P. A. Fernandes and M. J. Ramos*REQUIMTE ,
Departamento de Química e Bioquímica, Faculdade de Ciências, Universidade do Porto, Rua Campo
Alegre s/n , 4169-007 Porto, Portugal.

3.3.1.1. Abstract

We have developed a computer program named VolArea that allows for a rapid and fully automated analysis of molecular structures. The software calculates the surface area and the volume of molecular structures, as well as the volume of molecular cavities. The surface area facility can be used to calculate the solvent-exposed surface area of a molecule or the contact area between two molecules. The volume algorithm can be used to predict the space occupied not only by any molecular structure but also the volume of cavities, such as tunnels or clefts. The software finds wide application in the characterization of systems such as protein:ligand complexes, enzyme active sites, protein:protein interfaces, enzyme channels, membrane pores, solvent tunnels, among others. Some examples are given in order to illustrate its potential. VolArea is as a plugin of the widely distributed software Visual Molecular Dynamics (VMD) and is freely available at <http://www.fc.up.pt/PortoBioComp/Software/Volarea/Home.html>

Keywords: Surface Area; Volume; Molecular Dynamics Simulation; AMBER; Visual Molecular Dynamics

3.3.1.2. Introduction

From the simplest bacteria to the complex human being, proteins play essential roles, even viruses depend on them. Like other biological macromolecules such as polysaccharides and nucleic acids, proteins are virtually involved in all physiological functions, and understanding how they work is a fundamental problem in biology. In the last 40 years the atomic three-dimensional structures of thousands of proteins have been characterized by experimental techniques such as X-ray crystallography and nuclear magnetic resonance, revealing some of the key features that allow them to play unique roles in living organisms. One of these key features is their ability to specifically bind other molecules. These molecules can be either small compounds, such as low molecular weight substrates or inhibitors, or larger biopolymers like DNA, RNA or other proteins.

Understanding how large chemical structures bind each other and how small molecules bind to larger chemical structures with high affinity and specificity, has become a subject of paramount importance for the understanding of structure/function relationships in biology.

For a possible interaction between a ligand and a receptor to take place, it is necessary that there is enough space for the binding to occur, i.e. it is necessary that the ligand fits inside the binding site; to predict if the binding will occur with high affinity it is necessary to know which set of residues will interact more strongly with the ligand. These two characteristics can be determined, at least partially, by the volume of molecules and their cavities and the contact area that they share. The molecular surface area calculation can be applied also to evaluate different molecular properties, especially when the calculation is targeted to the property intended to evaluate. For instance, surface area calculation adapted to the atoms type can be used to calculate ADME (absorption, distribution, metabolism, and excretion) parameters such as the partition coefficient ($\log P$)¹⁰⁸, the contribution of this surface to the entropy term in the protein-protein interaction¹⁰⁹ and the solvation energy¹¹⁰. The advantage of these methods when compared with others such as Thermodynamic Integration (TI) or the Poisson-Boltzmann/Generalized Born Surface Area is the computational efficiency achieved where only surfaces areas are calculated.

In the last 5 years, several computational algorithms have emerged to calculate the volume and the surface area of molecular structures. However, despite the developments and the improvements carried out by these softwares¹¹¹⁻¹¹⁶, there is still much room for improvement, mostly in terms of quantifying the volume and surface, establishing multi-frame structure analysis or even through the development of molecular visualization interfaces. Table 1 summarizes the main characteristics of the most used softwares that perform this kind of calculations.

Table I - Summary of the software's available for calculating the volume and the surface of structures. a) Finds and assigns the routes of tunnels; b) Finds binding sites using protein sequence; c) Uses the surface algorithm to calculate the volume using Monte Carlo; d) Grid-based method used to find and characterize binding sites and pockets (cavities), using both the surface area and volume as parameters; e) Visualizes channels and tunnels f) Uses the same algorithm to calculate the surface and the volume.

Characteristics	Software					
	Caver ¹¹¹	PocketMatch ¹¹⁶	McVol ¹¹⁴	MDpocket ¹¹⁵	HOLLOW ¹¹²	3V ¹¹⁷
Volume	-	-	a	a	-	a
Surface	-	-	a	a	-	a
Multi-frame Analysis	-	-	a	a	-	-
Main features	a)	b)	c)	d)	e)	f)
Molecular Viewer	PyMol ¹¹⁸	-	-	-	-	-

In this work we describe VolArea, a new bioinformatics tool developed not only for analyzing molecular structures, such as proteins and nucleic acids, but also molecules of any size or nature. The program has been included into the widely distributed Visual Molecular Dynamics software (VMD) ³¹ as a stand-alone plug-in that provides a rapid, accurate and fully automated process to calculate the surface area ¹¹⁹ and the volume of any molecular structure. VolArea overcomes most of the limitations of the pre-existing software and provides a very intuitive and useful interface to study the structure of protein complexes, such as protein:ligand or protein:protein complexes, or even membrane pores or nucleic acids.

3.3.1.3. Methods

VolArea is a multiplatform application to calculate the surface area and the volume of any molecular structure. The full procedure is divided into two main algorithms, the surface area and volume algorithms described in the following sections. The VolArea graphical user interface (GUI) manages and displays all the data required and generated by VolArea. This GUI will also be described in the following sections.

One important characteristic of both surface area and volume algorithms is their dependence on the selection of atoms. This selection defines the universe in which the algorithms will be applied. Throughout the Application Section this dependency will be explained and clarified.

3.3.1.3.1. The Surface Area Algorithm

The surface area is calculated making usage of the algorithm developed by Futamura *et al.*¹²⁰, which has been already implemented in the VMD program³¹ as a command named *measure sasa*. VolArea intends to present here a new possibility for large-scale applicability and non-expert use of this command. Until now, the algorithm is only available as a command line, which becomes difficult to use for non-experts and difficult to apply in large sets of residues or structures even for experienced users. Using this plug-in, the user can calculate the surface area of any structure (not only residues), exposed to the solvent or in contact with other structures, such as small ligands, or even exposed to protein sub-units. All these calculations only require simple small atoms selection and a few clicks.

The surface area calculation depends on the selected area, on the probe radius, and on the number of points that are used to compute it. The probe radius can be defined as an *in silico* rolling spherical probe that is screened around the selected area. The surface formed by the rolling sphere above the molecule is the exposed surface area (Figure 16).

The default radius value of the probe sphere is 1.4 Å (the radius of a water molecule) and 50 points are used to estimate the solvent accessible area, but other values can be used to investigate other type of interactions.

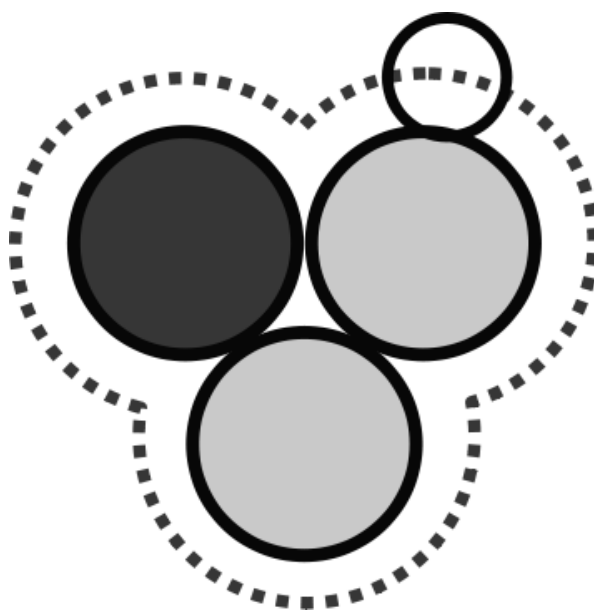


Figure 16: Illustration of the usage of the probe radius to calculate the surface of molecules.

In the end, the surface values are displayed in the GUI, as the total area of the selected region and the area of each residue defined by VMD. A statistical analysis of the surface values is also performed when two or more frames are available in the structure file (e.g.

molecular dynamics simulation). In this case the GUI presents the average value of the surface area, the *per* residue surface area, and the respective standard deviations (total and *per* residue). In multi-framed structures the GUI also creates a surface area vs frame number graphic.

3.3.1.3.2. The Volume Algorithm

To begin with, the user defines a “selection region”, which is the region of the molecular system, which he wants to calculate the volume of. It can be the entire molecular system or just a smaller fraction of it. The “selection region” always corresponds to a rectangular 3D box. All atoms that are inside the box are selected.

Additionally, all atoms outside the selection region, closer than 4Å from the edges of the selection box, will get selected too. The reason for including these atoms is that the tools for selecting atoms in VMD (and in most software) use the nuclei coordinates, and not the van der Waals (vdW) volumes. This means that if a given atom is partially inside the selection region, but its nucleus is not, it will not be selected, which is undesirable.

Subsequently, the selected region is divided in a grid of small cubes, and their side corresponds to the *scale* parameter, which is chosen by the user. Therefore, the volume of the cube is given by *scale*³.

The size of the cubes must be much smaller than the van der Waals radius of the atoms. Therefore, each cube may contain either a single nucleus or it can be simply empty. If the cube contains a nucleus, then the nucleus is centered in the cube, and the algorithm searches for the cubes that are within the vdW radius of the atom. All the cubes that are within the atom radius are marked as “occupied” by that same atom. Otherwise they are marked as “empty”.

If the user intends to calculate the volume of a specific molecular structure, he might not want to include any surrounding atom beyond the selection region, which e.g. may belong to another molecule close by. In that case the user can choose additional selection tools to specify more strictly, beyond the simple distance criteria, the atoms that he/she wishes to select.

The molecular volume is calculated by adding the volumes of all the small cubes marked as “occupied”.

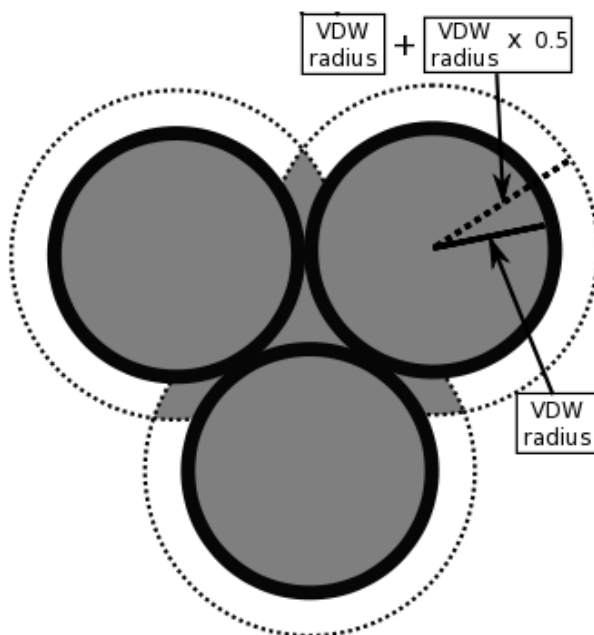


Figure 17: Schematic illustration of the searching radius superposition in order to illustrate the volume algorithm.

The total volume of a molecular structure should include small spaces between the atoms as well. Figure 17 depicts schematically those small spaces that exist between the atoms (when they are defined as spheres with vdW radius). These small spaces exist in any structure but they are smaller in crystalline materials or metals. They are particularly relevant in protein structures, where the atoms are not spatially very organized. Such small volumes are too tiny to be occupied by any atom/molecule. They are in fact “inaccessible” and this is why they should be included in the total volume of the molecular structure.

To include them, we further searched for the cubes that are located in a spherical shell, centered at the nuclei, having as lower and upper limits the vdW radius and 1.5 times the vdW radius (Figure 17). If a cube belongs to the spherical shell of, at least two atoms, it is included in the total volume of the molecular structure.

The calculation of the volume of cavities is conducted in a slightly different way. The search for cubes that belong to a cavity is performed with another probe radius, named “Cavity Probe Radius”, centered in the atomic nuclei. The radius of the cavity probe will be chosen by the user, according to the shape and size of the cleft or pocket. In fact, the user must choose the region in which he will be searching for a cavity, distinguishing between a “cleft” and “bulk solvent”, before the calculation takes place, in the same way as the user must select a region to calculate a molecular volume before the algorithm calculates its value. The cavity probe radius will be the tool defining the region to be analyzed for cavities.

Subsequently, the volume of the atoms is expanded from the vdW radius to the cavity probe radius. This is done assigning the value of “occupied” to all cubes within a sphere, centered in each nuclei, and having a radius given by the cavity probe. The cubes that are

within the cavity probe radius of more than one atom are assigned as “cavity cubes” and the final volume of the cavity corresponds to the sum of the volume of all cubes assigned as “cavity cubes”. (Figure 18).

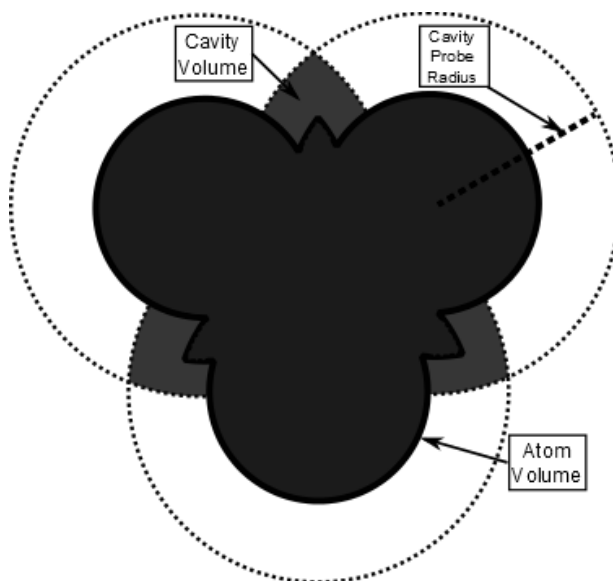


Figure 18: Example of the searching radius superposition, here applied to the cavity volume calculation.

The shape of the cavity may exhibit small irregularities in its borders, i.e. depressions/pockets so small that cannot accommodate any atom. The algorithm further searches for these very small surface pockets and eliminates all those that are smaller than the volume of a hydrogen atom (around $5 \text{ \AA}^3$). This elimination is accomplished by searching $1 \text{ \AA}$ in each direction of the current position, creating a search matrix with $8 \text{ \AA}^3$, which is large enough to include an hydrogen atom. In if the total empty space volume is small than the hydrogen atom volume, the current position is unmarked, disabling it to be part of the main cavity. The volume algorithm is resumed in Figure 19.

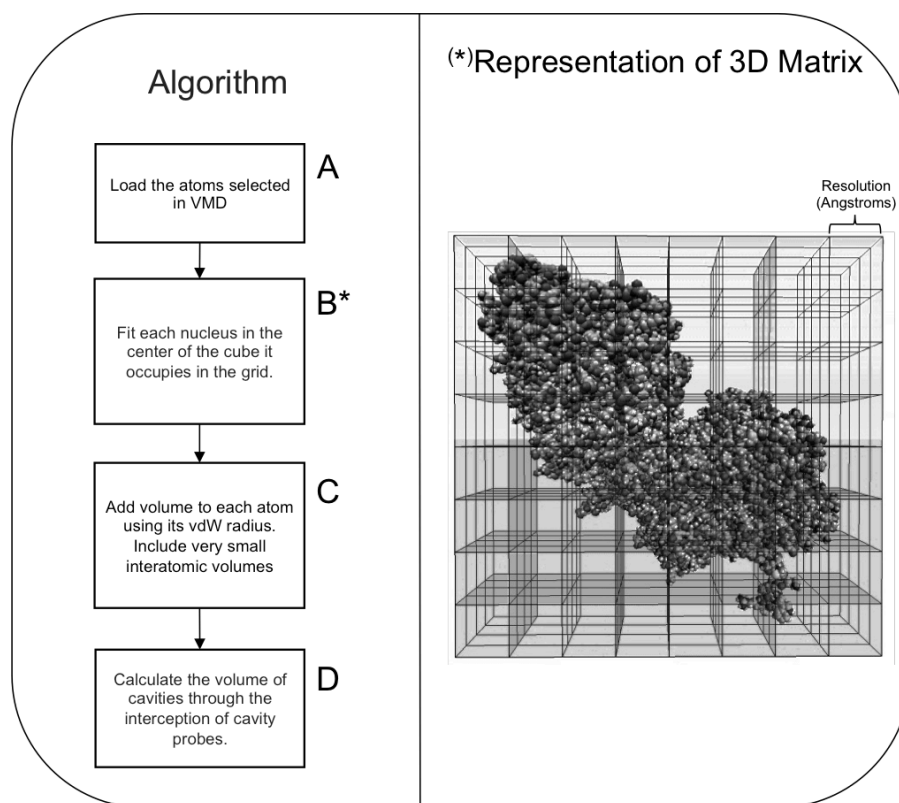


Figure 19: Diagram of the algorithm that is used to calculate the volume in VolArea.

If the input is a multi-frame structure the algorithm processes all the molecular conformations sequentially and presents the average and the standard deviation of the cavity volume or molecular volume, in the same way as it did with the surface area algorithm. These values can be represented also in a volume vs frame plot. Once this procedure has been completed, the final results are stored in a structured SQLite database (<http://www.sqlite.org/>) that can be used subsequently by the user for further analysis or data manipulation through standard SQL functions.

VolArea was developed with TCL/TK as the programming language and is available as a plug-in of VMD, version 1.8.7 or higher. This software requires a TCL shell installed in the host machine as well as the thread package. The version of this package depends on the type of operating system, i.e. for both windows and macOS operating system machines it is required the TCL shell version 8.4(32 bits) and tcl-thread package (2.6 or higher); for linux operating system machines, the TCL shell version 8.5(32 or 64 bits) and tcl-thread package (2.6 or higher) are required. All these package are freely available at <http://www.activestate.com/activetcl/downloads>.

3.3.1.4. RESULTS AND DISCUSSION

We start by validating the VolArea algorithm, by comparing the calculated molecular volume against experimental values¹²¹. Afterwards, we describe the Graphical User Interface (GUI) and exemplify some possible applications of this software, showing how it can be used to get insights into the physical properties of biochemical systems.

3.3.1.4.1. Validation

In order to validate the volume algorithm implemented in VolArea, we have calculated the volume of eleven proteins, and compared the results with experimental data¹²¹. The pdb codes of the structures used in this test were: 4pti¹²²; 3rn3¹²³; 1lzt¹²⁴; 5mbn¹²⁵; 3adk¹²⁶; 1ppn¹²⁷; 1rei¹²⁸; 2cna¹²⁹; 3est¹³⁰; 1rhd¹³¹ and 2ctb¹³².

We have used the Open Babel software¹³³ to add hydrogen atoms to their X-ray structures. Then we have calculated the protein volume with VolArea using different *scale* values (grid cubes sides). Finally, we have compared the results with the experimental values and evaluated the relative errors. All the protein structures had all residues defined in the pdb files. Figure 20 summarizes the results.

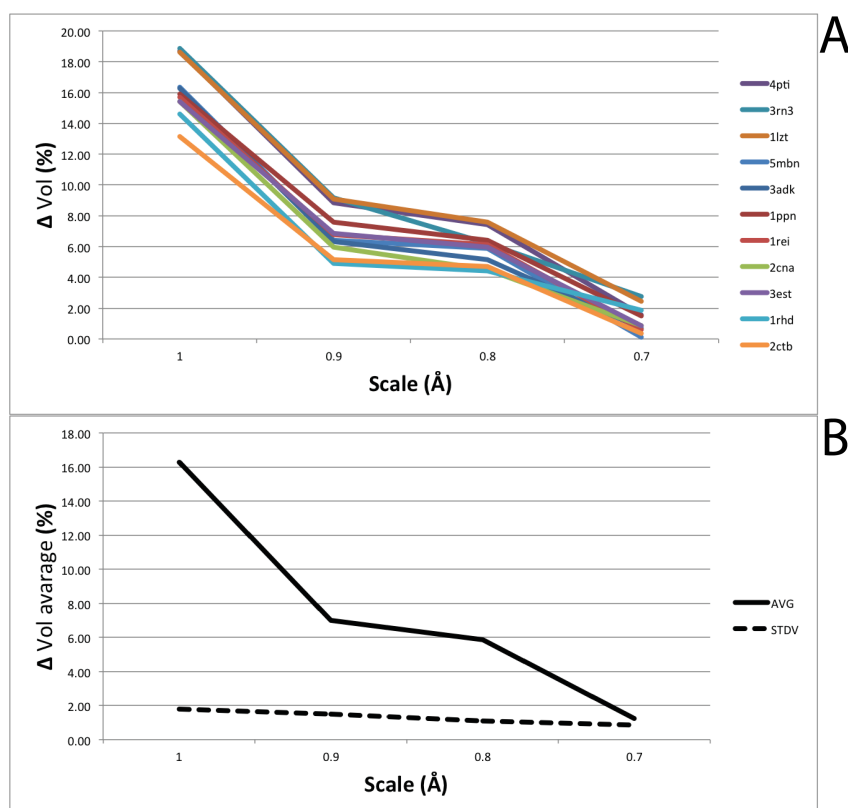


Figure 20: a) The relative error in the protein volume calculation (defined as the difference between the experimental and calculated volumes divided by the experimental volume) vs. the scale value and b) the average and standard deviation of the relative error for the 11 structures calculated. All results are presented as percentages.

Figure 20 shows that the influence of the *scale* value in the calculated volume was approximately the same in all tested proteins. The maximum deviation was 18.6 percent with a *scale* value of 1.0 Å in the 3rn3 structure and the minimum value was 0.11 % with a *scale* value of 0.7 Å in the 5mbn structure.

To understand how accuracy (i.e. the relative error in relation to the experimental value) depends on the total volume of the protein we plotted the values obtained before and presented them as a function of the experimental protein volume (Figure 21). The results do not show any direct correlation between them, meaning that the deviations are not systematic. Therefore, the algorithm accuracy is independent of the size of the molecular structure. Figure 6 indicates that the deviation is between 0 and 3.0 % at 0.7 Å *scale*.

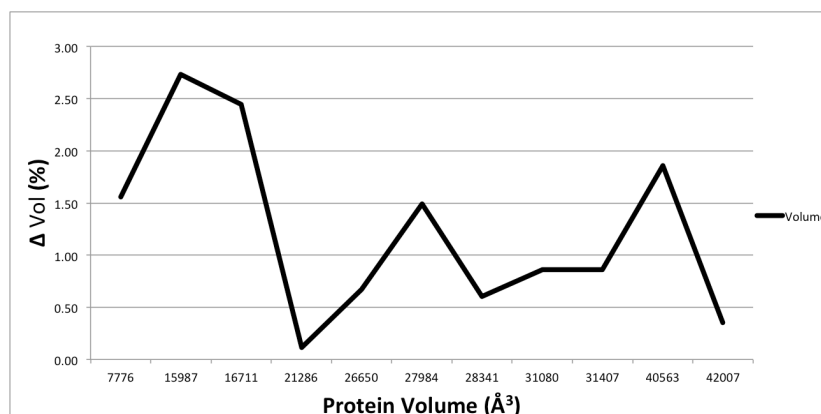


Figure 21: Variation of deviation as a function of the protein volume with a scale of 0.7 Å.

Note that the deviation of the calculated values from the experimental ones arises in part from the fact that the experimental volume reflects an average over many conformations. On the other hand, a single pdb structure was used in the calculations for each particular case. However, the x-ray structure incorporates implicitly an average folding over many molecules in the crystal, but in the end the atom positions in the pdb files must be fitted to just a single stable minimum energy conformation. Another source of deviation is the *scale* used in each test. We have limited the *scale* values to a range between 1.0 and 0.7 Å because within this range the accuracy increased almost linearly with the decrease of the *scale* parameter. For *scales* smaller than 0.7 the accuracy changes in a non-systematic way due to the set of approximations of the algorithm, which was designed to produce very accurate results with large *scale* values for improved computational performance.

In fact it is desirable to include all the scales values between 1.0 and 0.1, which is impossible due to the computational demanding that makes it intractable. The error associated to the limitation of the scale to the values between 1.0 and 0.7 are canceled by the usage of the 1.5 multiplier in the van der Waals overlap search algorithm. The results observed in the Fig 20 reveal the non-fortuitous behavior of the algorithm, reflected in the

good agreement in the tendency of the error in terms of volume percentage in function of the scale.

The code was parallelized to enhance the performance of the volume algorithm, in particular when processing multi-frame structures from computer simulations. Figure 22 shows the speed up of the process using two or four processing cores. All calculations were performed in a *quadcore* machine (Intel Core 2 duo 2.66 GHz with 4 Gbytes of RAM). Calculation times refer to the time of the volume calculation only, excluding the time spent in representing graphically the volume in the OpenGL VMD window, since this representation is performed only once independently of the number of frames analyzed. The time shown for each protein in Figure 22.a. is the average time for the calculation of the volume taking into account all *scale* values tested before (1.0; 0.9; 0.8 and 0.7 Å). In Figure 22.b, the ratio of time in the volume calculation is presented using two (2C) and four (4C) processing cores. Here, the values correspond to the time average (in seconds) of the volume calculation, vs. a given *scale* parameter. The parallelization performance was almost the same whatever the resolution of the protein used.

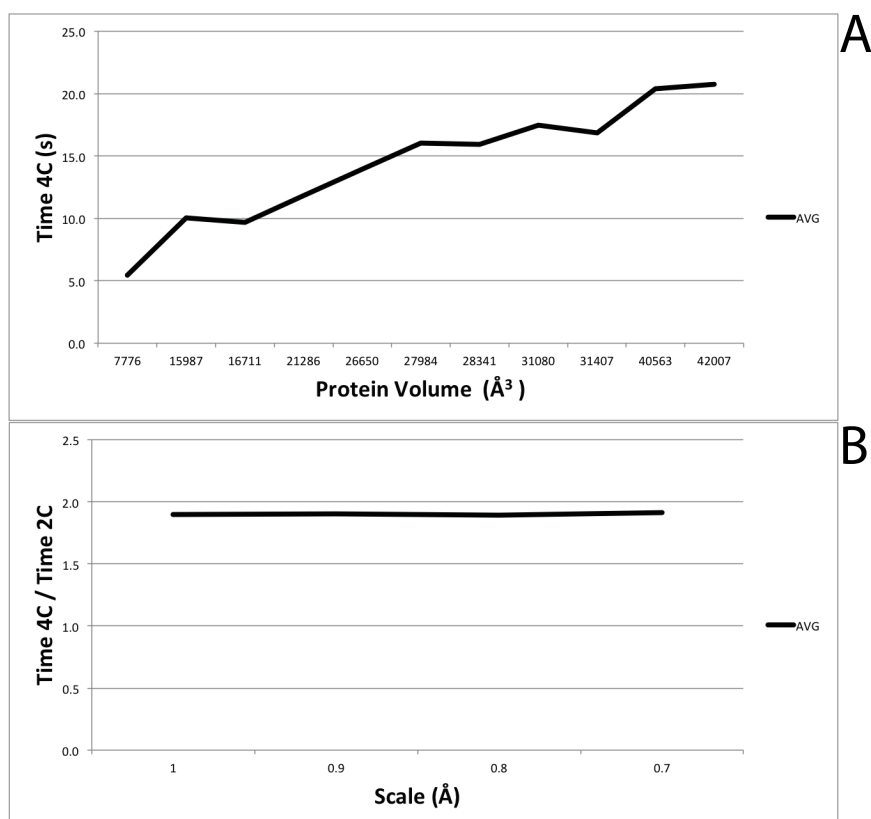


Figure 22: a) Average of the times of the protein volume calculation (using scale values from 1.0 to 0.7 Å) using four processing cores (4C) as a function of experimental volume, b) ratio of the times in the volume calculation using two (2C) and four processing (4C) cores. The speed up is linear.

The time required to calculate the protein volume grows with the protein volume, as expected (Figure 22.A). It should be noted that the difference between the smallest (7800 Å³) and the largest protein (42000 Å³) corresponds to less than 20 seconds, independently of the resolution.

In this analysis it is possible to see that the user saves 50% of computing time duplicating the number of cores (linear speed up) (Figure 22.B), and this ratio was observed in all proteins studied.

3.3.1.4.1.1. Graphical Interface

In order to generalize the application of these algorithms and to make it available to a wider audience, we developed a GUI in Tcl/Tk¹³⁴ that was included in VMD as a plug-in (Figure 23). We have chosen VMD because it is an extremely powerful molecular viewer that represents the structure of the molecules in a wide range of formats and is very handy to perform many structural analyses. In addition, VMD is a very flexible program that can be used to display all the data generated by VolArea. This is extremely important because both the surface and the volume can be difficult to visualize if not displayed in three-dimensions together with the molecular structures.

All the variables that control the execution of the surface area and the volume algorithms can be manipulated in this GUI (Figure 23-A). To make the calculations faster, the GUI contains a selection module that restricts the region of analysis (global and specific selections in Figure 23-A). The region can be selected through a yellow box that can be interactively handled by the user (Figure 23-A). In addition, the user can make use of a set of selection tools to specify more closely what he/she wishes to analyze. For instance these tools can be used to calculate the volume of a cavity occupied by a ligand (Figure 23-A). These selections can be of any of the types recognized by VMD.

All the results generated by VolArea are stored in a unique file that has a SQL database format. The same GUI can be used to analyze the results (Figure 24-B) or the data can be exported to other applications for further analysis. In the case of multi-frame structures the program also provides a graphic visualization of the variation of the surface and volume during a trajectory (Figure 23-B).

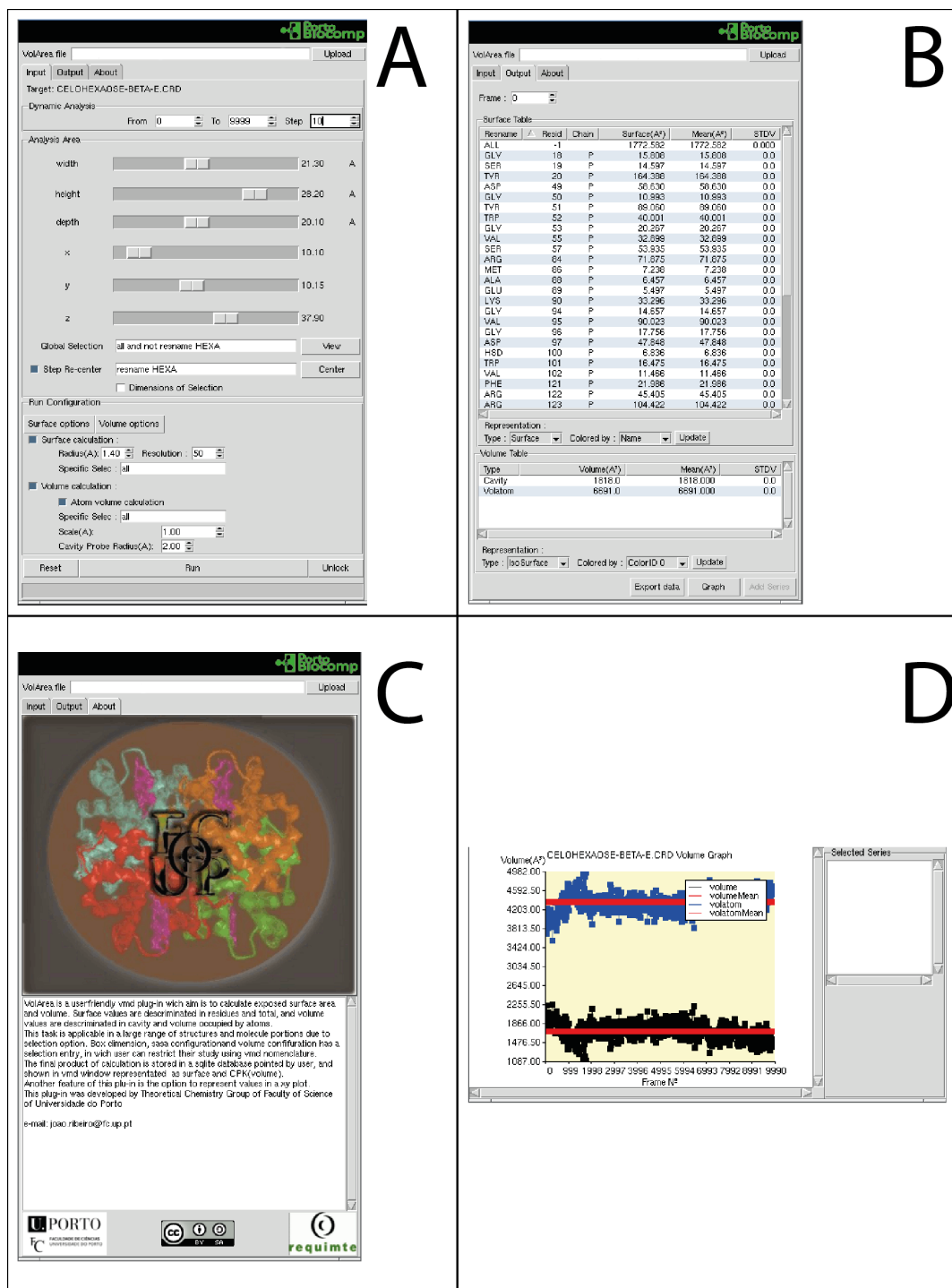


Figure 23: VoIArea Graphical interface: (A) The Input tab is the main window of the program - where the calculations of the surface and the volume are set and where the user defines the area that he/she wishes to analyze. (B) The Output tab is where the results are presented/analyzed or can be exported to other programs. (C) The About tab provides information on the VoIArea plug-in (D) The graphic window is part of the output tab and allows the user to plot the results produced by VoIArea.

3.3.1.4.2. Applications

Here we describe the potential of VolArea and how it can be used to get insights into the physical properties of biochemical systems.

The surface area measures the exposed area of a molecular structure. It can be used to calculate the solvent accessible area of a molecular surface or the contact area that is shared between two or more chemical structures that are tightly bound, such as protein:ligand complexes, protein:protein complexes, protein:DNA complexes, etc. In the case of proteins, VolArea provides the individual contribution of each residue for the total surface area. This information can then be used, for instance, to map all the residues that are present in the analyzed surface according to their exposure to the medium and to identify important electrostatic characteristics in binding sites.

The volume tool can be used to estimate the three-dimensional space that a molecular structure occupies as well as the free volume in a pre-selected region. The latter can be used to calculate the volume of internal cavities, surface clefts of proteins, as well as the free space between two interacting molecular structures such as protein:protein interfaces, protein:membrane complexes, protein:DNA complexes, etc. It finds wide application in predicting if a ligand fits inside the binding site of a protein, a pre-condition to select candidates for drug discovery programs.

Estimating the value of the surface area and of the volume can be a tricky job as it is largely influenced by the specific molecular conformation considered. Taking this into account, VolArea can calculate the surface area and the volume of single structures (such as the ones retrieved from X-Ray structures) as well as ensembles of structures (retrieved from molecular dynamics or Monte Carlo simulations as well as from Nuclear Magnetic Resonance (NMR) spectroscopy). In all cases the source data can assume any of the file formats that can be handled by the VMD software. This means that the user can use this software without any previous conversion or manipulation of the source data. PDB, mol2, xyz and other file types handled by VMD are well supported by VolArea, as well as multi-frame structures that are retrieved by popular molecular dynamics applications such as AMBER¹³⁵, GROMACS¹³⁶, CHARMM¹³⁷, or NAMD¹³⁸, among others.

In order to generalize the application of these algorithms to any problem, and to make it available to a wider audience, VolArea is distributed with a graphical user interface (GUI). The developed GUI has several modules that ensure a smooth execution of the program, requiring only minimal user intervention, such as selecting the area of the chemical structure that the user wishes to analyze. All the generated information is displayed on the graphical interface and on the VMD backend, but may also be exported to other programs for further analysis.

To demonstrate the potential and usefulness of this application, we present a set of examples involving proteins and small molecules. VolArea can handle isolated structures such as proteins, membranes, polysaccharide chains, DNA, etc., but also complexes made up of several proteins (protein:protein complexes) or of different chemical structures, such as protein:ligand complexes, protein:DNA complexes, protein:membranes, etc. In the latter cases, the program is even prepared to study each unit independently, which can be useful to derive structural information from the interaction between both subunits.

The examples that are described in the following sections are arranged in two groups: the static structures and the multi-framed structures.

3.3.1.4.2.1. Static structures assay

VolArea can be used to analyze the interface between two or more proteins. There is great interest in the identification of the residues that control the association and the interaction between proteins. It is known that just a few residues contribute for most of the binding affinity. The identification of these residues is not straightforward, but often they can be pointed out by the recognition of structural features that are indicative of their importance in the binding. This type of analysis can be performed by VolArea, using its selection tools and exploring the surface area in the interface region between the binding proteins. The resulting information can then be used to visualize the complementarity of the residues that are present on the protein interface and to understand which type of regions might be important for protein interaction. One example of such analysis is shown in Figure 24.

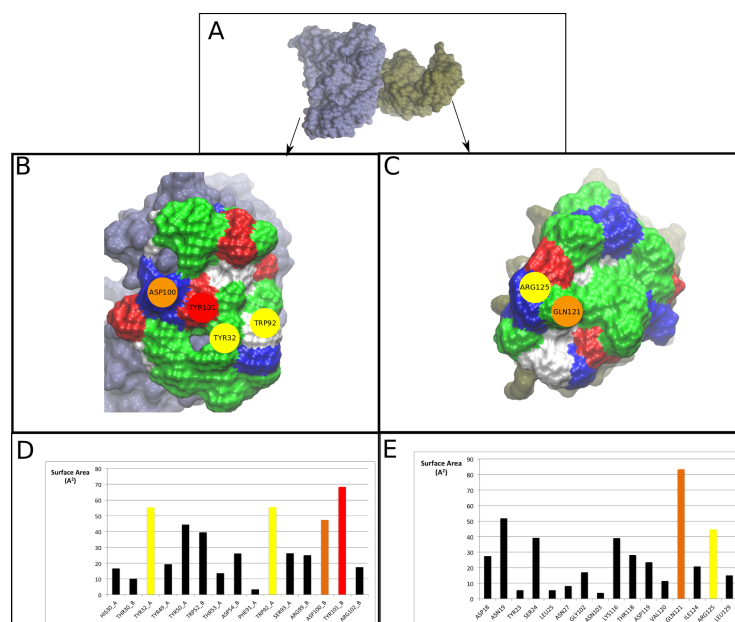


Figure 24: Surface values obtained with VolArea while analyzing the interface region of two proteins (retrieved from the PDB structure 1VFB): A) Protein complex; B and C) representation of the interface region of each protein (colored by residue polarity - non-polar residues (white), basic residues (blue), acidic residues (red) and polar residues (green)), with the identification of the hot, warm and null spots with colored circles; D) and E) The buried area upon dimerization for each of the interfacial residues. The hot spots are colored red, the warm spots are colored orange and the null spots are colored yellow.

This analysis was applied to the protein complex composed by two proteins: the hen egg lysozyme and the immunoglobulin (Figure 24A, PDB entry: 1VFB¹³⁹). In order to identify which residues are more important for protein binding, we explored the interface region of each protein and calculated the buried area upon dimerization. The importance of the residues in a protein-protein interface can be reflected in the binding energy variation upon mutation by an alanine residue, where a residue mutated by an alanine that implies a variation of the binding energy between 4 and 2 kcal/mol is denominated a hot-spot; a warm spot is associated with a variation in the binding energy between 2 and 0 kcal/mol and a null spot is associated with a variation in the binding energy below 0 kcal/mol^{140,141}. For this purpose, we started by calculating the exposed surface area for all residues located at the interface region, as if the proteins were dissociated from each other. Subsequently, we have subtracted this value to the exposed surface area of those residues when the complex is formed. The buried surface upon complexation is a very important property because it has been shown before¹⁴² that to have a very small area exposed to the solvent in the complex it is a necessary (but not sufficient) condition for a residue to behave as a hot spot.

This can be easily done using the set of selection options that are present in the VolArea graphical interface. The results are depicted in Figure 24D-E, and show that the residues that bury the most upon complexation are Tyr101, Asp100 and Gln121. These results are very similar to those obtained previously by experimental and computational means, in which

those residues were identified as hot (Tyr101) and warm spots (Asp100 and Gln121)^{27,139}. Besides those residues, VolArea has also identified other amino acids that present a significant fraction of their area buried upon complexation, when compared to the dissociated form (but in lesser degree than the previous ones), namely Tyr32, Trp92 and Arg125. These residues were not identified as hot or warm spots. On the contrary, they were correctly identified as null-spots^{27,139}. This confirms that a hot spot is buried in the interacting subunit, but a buried residue is not necessarily a hot spot.

This small, but illustrative example shows that VolArea can be used to highlight important regions of protein interfaces that can be crucial for their association/interaction. This means that it can be a useful tool to assist the screening of proteins interfaces in the quest of finding hot and warm spots, as a first approximation, and to help the experimentalist to choose the residues more suitable to carry out site-directed mutagenesis, avoiding long and costly mutations of residues that we can predict that cannot be hot-spots.

3.3.1.4.2.2. Multi-framed structures assay

To demonstrate the potential of VolArea in analyzing multi-framed structures, we checked the results obtained from a molecular dynamics simulation that was obtained in a previous work^{37,143}. The system presented here simulates the interaction of cellohexaose with the carbohydrate-binding modules (CBM) from family 11. The CBM 11 proteins belong to the cellulosome multiprotein superstructure. The main function of these proteins is to increase the efficiency of the enzymes that hydrolyze cellulose, by detaching cellulose strands from the cellulose network, breaking the hydrogen bonds between them and guiding them to the inner location of the protein/enzyme consortium, where the hydrolytic enzymes are located¹⁴³.

The initial structure of this complex was obtained computationally by molecular docking with our software MADAMM³⁸. The obtained complex was subsequently relaxed by a molecular dynamics simulation of 2 ns. The final results suggested that the binding of polysaccharides needed to be preceded (or induced cooperatively) by a conformational rearrangement of several residues of the active site, in particular of a set of tyrosine and aspartate residues. In spite of being small, these changes led to a significant modification of the binding site. There was good agreement between the induced-fit binding pose and experimental NMR results^{37,143}. At the time, it was difficult to follow and characterize these conformational modifications of the binding site, but now VolArea provides an excellent tool to perform this task. Accordingly, we first started by calculating the changes on the volume and on the exposed area of the binding cleft. These surface values were calculated excluding the contribution of the polysaccharide (ligand) in the region of the binding site, and

centering and resizing the selection box using the ligand dimensions. In the case of the calculation of the volume of the binding cleft, the box was centered and resized in the ligand as in the previous calculation, decreasing the dimensions to make sure that only the cleft volume was calculated (box dimensions of 16.87 Å width vs 16.06 Å height vs 20.10 Å depth). We used a resolution of 1.0 Å and a cavity probe radius of 5.0 Å, to guarantee that all the cleft was analyzed throughout the simulation, covering the variations in the binding cleft width. The final results are shown in Figure 25.

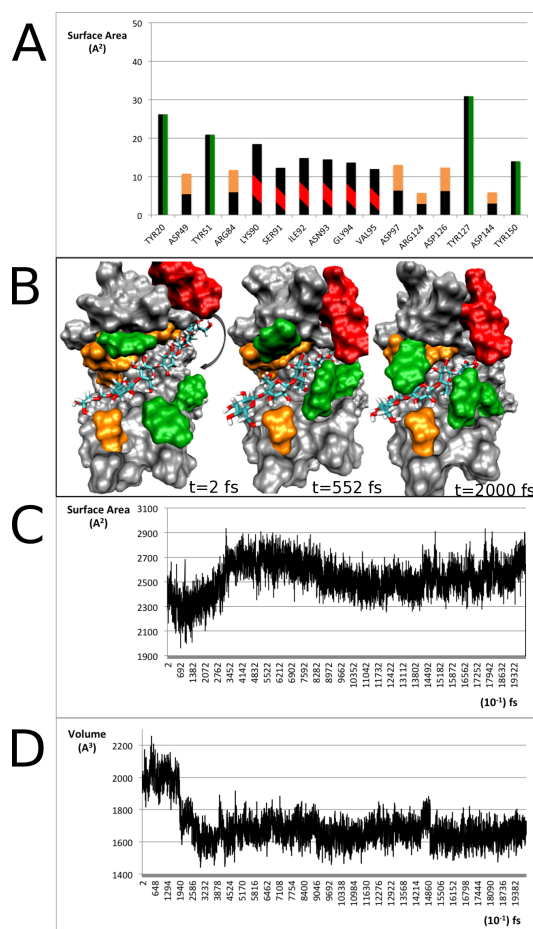


Figure 25: A) Exposed surface area of the residues located at the protein interface that interact more closely with the ligand. B) Snapshots of the MD simulation during the ligand-induced closing of the binding cleft. C) Total exposed area of the binding cavity through the molecular dynamics simulation, D) Volume of the binding cavity through the molecular dynamics simulation.

The values obtained with Volarea show that at the beginning of the simulation, the volume of the binding site decreases at the same time that the exposed area of the binding site increases. Subsequently, they remain almost constant (in particular the volume of the binding cavity), which indicates that the system becomes equilibrated in this respect. In fact, the decrease in the volume and the increase in the exposed area of the binding site indicate that the binding site shape changes to allow for the establishment of a larger number of contacts with the ligand. This is very interesting since it indicates that during the simulation the

binding site closes and embraces the ligand, probably improving protein:ligand affinity, a perfect example of the induced-fit paradigm. This is most of the time difficult to detect visually in the molecular dynamics simulation, but now it can be easily analyzed both qualitative and quantitatively with the data provided by VolArea.

As VolArea provides the individual contribution of the exposed area of each amino acid that is present on the protein surface, we analyzed these results and searched for the residues that were interacting more closely with the ligand. The results are depicted in Figure 10 and show that the residues that interact more closely with the ligand are four tyrosine residues (Tyr20, Tyr 51, Tyr127 and Tyr150) and several polar amino acids (Asp49, Lys90, Asn93, Asp97, Arg124, Asp126, Asp144). These results agree with our previous results and with what is observed experimentally^{37,143}. However, in our previous study this type of analysis was made with a set of complex scripts. Now with only a few clicks we can extract this kind of information, and visualize it with the powerful back-end of VMD.

3.3.1.5. Conclusion

We have developed the VolArea plug-in to compute the surface and volume of molecules as well as the volume of cavities and clefts. This small but agile VMD plug-in has revealed itself as a very useful tool in molecular surface area and molecular volume analyses. The inclusion of this program in the molecular visualizer VMD will contribute to an easier and more general usage of VolArea. Despite not being revolutionary, the possibility of application of Volarea to this kind of analyses in large scale multi-frame structures and its friendly graphical interface makes it very useful for the community working in the field.

This package calculates the volume of clefts and tunnels, and maps all the direct contacts between molecules in a molecular complex. Currently, these properties could only be obtained through a set of complicated selections, scripts and programs, but with VolArea they can be easily calculated with a single program, which offers a friendly graphical interface that can be used almost by everyone without requiring any previous knowledge in the field. Furthermore, the resulting information can be visually analyzed using the powerful graphic backend of VMD.

VolArea can be used to analyze static structures and to analyze multi-framed structures, such as those obtained from molecular dynamics simulations. Here, it is possible to follow the variation of the exposed surface area of proteins (or other chemical compound) and the variation of the volume of the binding site cavity throughout the full simulation. The final results provide meticulous data about how these properties progress in time, and can be directly visualized with the VMD graphical backend. These results show that VolArea can be

used as a new and very useful tool to analyze the trajectories of molecular dynamics simulations

3.4. VolArea and CUDA

Despite the good results described above, it was clear that this software, VolArea, had a great potential for improvements, especially in the computational time required by the volume algorithm. To provide this software of a faster volume calculation method, the algorithm was adapted to the new technology specialized for the computational demanding calculations, which is based in highly parallelized codes running in Graphics Processing Unit (GPU). The usage of GPUs in molecular simulations or for molecular studies purposes has produced good results, being adapted as an ordinary tool in these kind of computational technics^{25,144-147}. The generalized usage of these units for extremely demanding calculations has being supported by programming languages such as Nvidia CUDA (Compute Unified Device Architecture) and OpenCL. General gains retrieved from GPUs in Molecular studies and additional considerations are presented in John Stone *et al.* publication¹⁴⁸ and for more information, is advised the reading of the CUDA Manual¹⁴⁹. The general scheme of the GPU architecture is presented in Figure 26, where is important to highlight the single processing units “streaming multiprocessor” (SM), which are composed by numerous small processing units “streaming processors” (SP) in which the threads are performed. In this figure is also possible to visualize the different levels of memories, where the local SM cache memories present a faster memory access than the global and Level 2 Cache Memory.

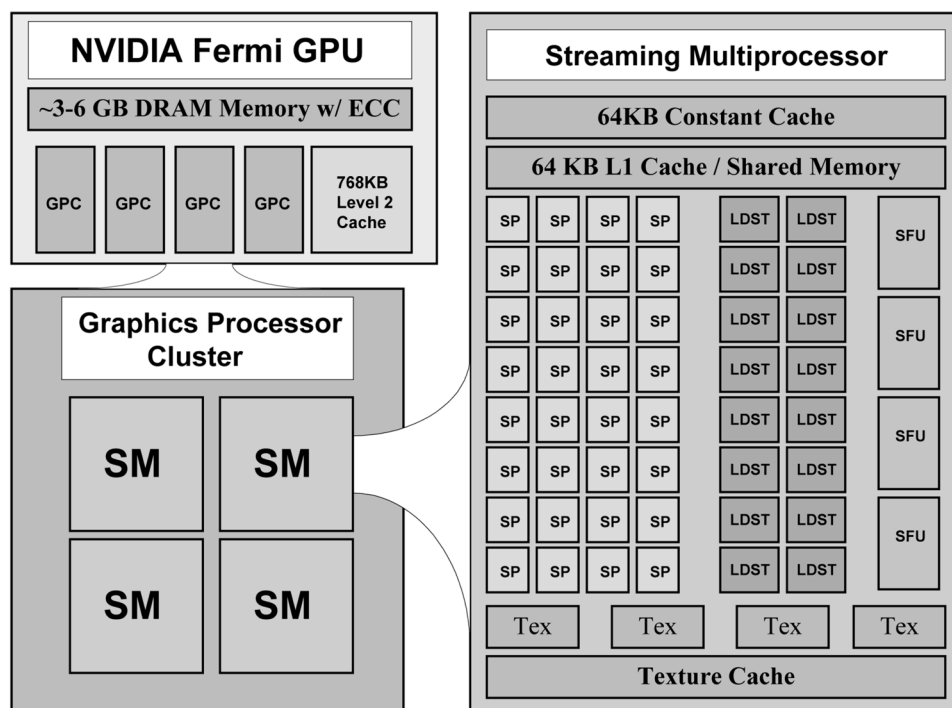


Figure 26: Simple representation of the NVIDIA Fermi GPU architecture. “A simplified hardware block diagram for the NVIDIA representing the arithmetic units “streaming processors” (SP) and “special function units” (SFU) for computing especial algebraic functions. Memory load/store units (LDST), texture units (TEX), fast on-chip data caches, and a high-bandwidth main memory system. Groups of 32 SPs, 16 LDSTs, 4 SFUs, and 4 TEXs compose a “streaming multiprocessor” (SM). One or more CUDA “thread blocks” execute concurrently on each SM, with each block containing 64 to 512 threads ¹⁴⁸.

In terms of the programming skills, CUDA programming language allows the programmers to apply GPU technology in their work in a well familiarly environment like C programming language, instead of the traditional graphics dedicated programming language like OpenGL. CPU-GPU codes are very similar to an ordinary C, C++ or Fortran code, adding only some demanding operations such memory transference. The GPU code itself has small differences inherent to GPUs, presenting at the same time a small period of training and learning. It is also possible to apply functional libraries containing several important and useful pre-defined functions like THRUST(<http://docs.nvidia.com/cuda/thrust/>) or CUDA Math library, very important for programmers that include these kind of libraries in their codes for arithmetic operations improvement and code simplification. One of the drawbacks of GPUs usage is the implicit hardware dependency that requires extremely attention in the development routines, which leads to several recommendations for maximum code boost from these hardware units. One of these recommendations deals with the reorganization of the variable values and how the variables are created. Here the maximum independency of the variable values is advisable and the sequential organization of the variables in classes or templates is also advisable in order to extract the maximum transference speed of information and efficiency in the memory access. It is based in these requirements that the VolArea algorithm was

adapted in order to take advantage of GPUs capabilities. The approach adopted in this work is based in the cutoff potential algorithm published in ¹⁵⁰.

3.4.1.1. Data Structure

One of the keystones for the development of GPU accelerated algorithms is the data structure. The optimization of the GPU code is achieved when the memory accesses are minimized and memory transferences between CPU and GPU are optimized. In the published work, the volume of the atoms was constructed by searching in the three spatial axis(x,y and z) for positions between the center of the atom and the van der Waals radius. Here, additional radius is added in order to find small spaces between the atoms, which can be part of the molecular volume. This search algorithm is performed by construct a search matrix for each atom, where the center of the search matrix is the center of the atom and the width, height and depth of the matrix have the dimensions of 3 times the van der Waals radius. In each grid position, the distances are calculated to evaluate if it is part of the volume or not. In this way, the search algorithm is dependent on the atoms creating a high degree of interdependence between the tested position and the atom center. This dependency must be avoided in order to adapt the algorithm to GPUs.

The GPU version of this algorithm has the same algorithm of the previous work, changing only how the distances are calculated and their dependency on the atoms center. In this new version, the atoms are gathered into bins, which are calculated based on their positions and based on a cutoff distance that associates the atom and its bins (Figure 27). For instance, if the cavity search radius chosen by the user is 5 Å, thus these bins will have the dimension of 5x5x5, calculating which atom belongs to each bin as exemplified for coordinate X in equation (14):

$$X_{bin} = \left(\frac{X_{atom} + ScaleFactor}{Cutoff\ distance} \right) * Scale \quad (14)$$

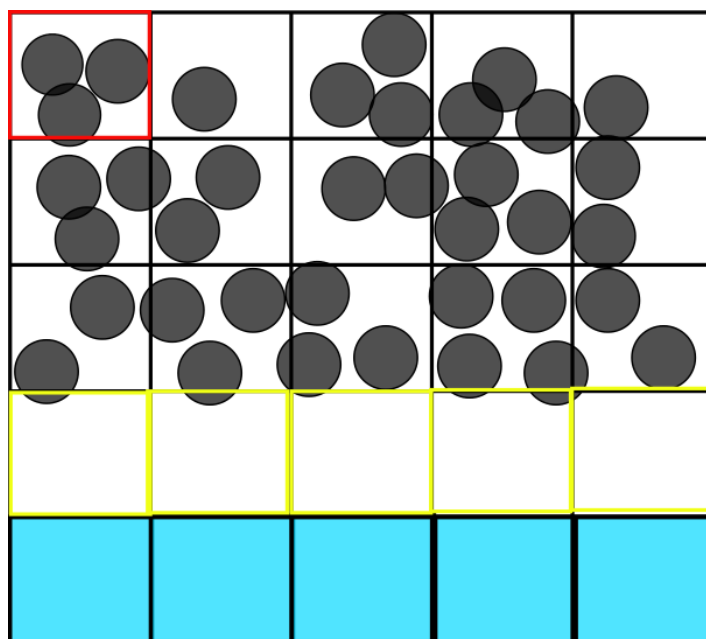


Figure 27: Schematization of the organization of the atoms in bins. Here, the red square is highlighting the atoms bin with the index 1. The yellow square is highlighting the bins that despite of being empty, they are taken into account due to the atom proximity, and the blue square are highlighting the volume excluded from all volume calculation.

In (14) the *ScaleFactor* is used to adapt the coordinates to the chosen grid *Scale* value and to avoid exceptions from out of limits references. This factor is translated by the equation 15:

$$ScaleFactor = (D_{cavity} + R_{max} * 1.5 + 1) \quad (15)$$

The *ScaleFactor* is equivalent to the cavity search distance D_{cavity} added to the maximum van der Waals radius R_{max} multiplied by the 1.5 as explained in the previous VolArea volume algorithm, and adds 1 Å to create “vacuum” around the protein to avoid *out of the limits* errors originated in rounding operations.

In the case of VolArea, this cutoff distance is the search distance defined by the user as cavity search radius, or, if only the atoms volume is intended, the cutoff distance adopted is the 1.5 of the maximum van der Waals radii. This organization allows the search algorithm to be only dependent on the grid position that is currently being tested. Here, this grid position will be referred to an atoms bin, and then, only the atoms inside of this bin will be tested, calculating the distances in order to evaluate if it is equal or shorter than van der Waals radius of one of these atoms. If the atom is located in the border of the bin, one ore more neighbor bins are selected and the correspondent atoms are tested.

Besides the faster access memory printed by this atoms bin approach, this same algorithm can save memory and time by intensifying processing work where it is needed. For instance, if the atoms are sorted by their bins position, and only the non-empty bins were

carried out for the distance calculation functions of the algorithm, the hardware memory alignment required is achieved and the empty spaces far from the atoms are excluded.

Implementing this data organization and transferring only the atoms center coordinates to GPU memory, each grid position can be treated as a single thread in the SP unit increasing tremendously the speed of all calculation. Higher efficiency is achieved, as mentioned before, when the memory transfers are minimized. In this algorithm this premise is fulfilled when only the molecular volume value, cavity volume value and representation coordinates are retrieved from GPU. All intermediated values are kept in GPU memory or even erased, and the CPU performs the management and instructions for these values.

3.4.1.2. Results

To test the gains of GPU version against the CPU version, a simple test were carried where the time of the GPU required to calculate the molecular volume were compared with the new C++ version of the volume algorithm. This test was performed against the previous pdb structured tested in the published version: 1lzt¹²⁴; 1ppn¹²⁷; 1rei¹²⁸; 1rhd¹³¹ and 2ctb¹³² and the 1vdv¹⁵¹. The inclusion of the 1vdv pdb structure is justified by the number of atoms (42000) and their volume (approximated 350,000 Å³ calculated with VolArea), making this structure a good example to demonstrate the GPUs potential.

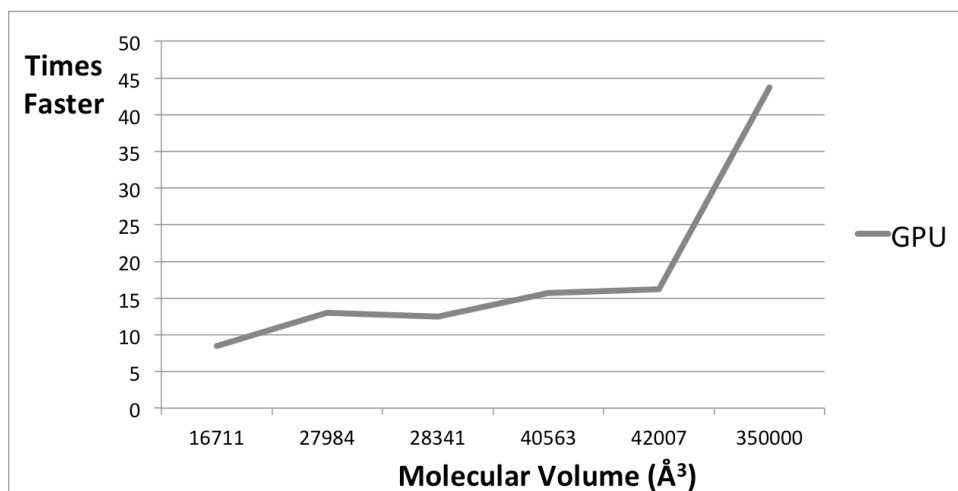


Figure 28: Chart displaying the gains retrieved from GPU. This test reflects how many times is GPU calculation faster than CPU version of the same algorithm. The test was conducted in a NVIDIA GPU prototype, equivalent to the new NVIDIA Tesla K20.

The chart presented in Figure 28 reflects the advantages of using the GPU for the very computational demanding operations. The gains are presented in terms of times faster, reaching almost a score of 45 times faster when GPU version is compared with the CPU version. The abrupt increase between 42,000 and 350,000 Å³ is a clear sign that as the

system size increase, the higher are the advantages in using GPU. The maximum improvement of the GPUs is retrieved when the right amount of job is sent to these units. A good practice in GPU coding is design the algorithm to create a number of threads between 10,000 and 30,000 (approximated), saving latency and memory transfer time.

The results presented here are only preliminaries demonstrating the potential that can be achieved when GPU technology is applied. As the algorithms becomes more optimized, it is expected a small variation in the differences of speed (could be lower), yet it is expected that the GPU version could at least save time in two or three order of magnitude.

The optimizations required in this VolArea GPU version are essentially in order to solve portability and compatibility issues. In the development of this algorithm, it was included some features that are exclusive of the newest models associated with the CUDA Compute Capability (CCC) version number (<https://developer.nvidia.com/cuda-gpus>). The algorithm was developed based on a model presenting a CCC number 3.0 or higher, which allows the appliance of the atomic operations and parallel reductions. For a general usage of this version, these operations must be redesigned or even replaced to a self-developed GPU counter and sum operator. It is also intended the realization of a broader benchmark test to analyze the behavior of this algorithm in a larger set of structures and GPU models.

3.4.1.3. Conclusion

GPUs technology is been widely applied for molecular studies purposes, returning good results, especially when the costs/processing potential are compared with CPU or CPU clusters. The adaptation of VolArea volume algorithm to this new technology revealed a great potential in terms of computational time gains. The preliminaries results shown that it is possible to improve the computational time more than two orders of magnitude by applying GPUs. Despite the good results, more work is required in order to guarantee the proper portability and compatibility between different GPU devices and different operating systems.

3.5. CompASM

The study of protein interactions, either with other proteins or with smaller molecules, can be a difficult and very demanding operation mainly due to the amount of information involved. The study of protein-protein interface surfaces, where the main purpose is to analyze the importance of the interfacial residues in the stabilization of the corresponding protein-protein complexes, is also characterized by the exponential growth of files and information. This analysis can be performed by the Computational Alanine Scanning Mutagenesis (CompASM) software developed in this work. We introduce here a package that drives the user through the main steps of the ASM algorithm: the ligand/receptor selection, the selection of the residues to mutate, the performance of molecular simulations and the visualization of the final results in a very intuitive and informative way. This protocol is based on Molecular Mechanics/ Poisson-Boltzmann Surface Area (MMPBSA) scripts that evaluates the difference between the Gibbs free energy of the wild and the mutated complexes. In order to provide the accuracy expected in this kind of calculation allied to the visual analysis mostly required in these studies, this tool requires also two other software programs already in the literature, Visual Molecular Dynamics (VMD) and the molecular dynamics package AMBER.

Article IV

“CompASM: an Amber-VMD alanine scanning mutagenesis plug-in“

Ribeiro J., Cerqueira N.M.F.S.A., Moreira I.S., Fernandes P.A., Ramos M. J.
Theor Chem Acc 131:1271-1278, 2012.

3.6. CompASM - an Amber-VMD Alanine Scanning Mutagenesis plug-in.

J. V. Ribeiro, N.M.F.S.A. Cerqueira, I.S. Moreira, P.A Fernandes, M.J. and Ramos

REQUIMTE , Departamento de Química e Bioquímica, Faculdade de Ciências, Universidade do Porto, Rua Campo Alegre s/n , 4169-007 Porto, Portugal.

3.6.1. Abstract

Alanine scanning mutagenesis (ASM) of protein–protein interfacial residues is a popular means to understand the structural and energetic characteristics of hot-spots in protein complexes. In this work, we present a computational approach that allows performing such type of analysis based on the Molecular Mechanics/Poisson-Boltzmann Surface Area (MMPBSA) method. This computational approach has been used largely in the past and has proven to give reliable results in a wide range of complexes. However, the sequential preparation and manual submission of dozens of files has been often a major obstacle in using it. To overcome these limitations, and turn this approach user-friendly, we have designed the plug-in CompASM (**Com**putational **A**lanine **S**canning **M**utagenesis). This software has an easy-to-use graphical interface to prepare the input files, run the calculations and analyze the final results. CompASM was built in TCL/TK programming language to be included in VMD as a plug-in. The CompASM package is distributed as an independent platform, with script code under the GNU Public License from <http://compbiochem.org/Software/compasm/Home.html>.

Keywords: Protein-protein interactions; Amber; VMD; MMPBSA; software.

3.6.2. Introduction

The association of proteins and the way they bind are a crucial topic in the study of living organisms. This importance stems from the fact that protein-protein interactions play a crucial role in the molecular recognition and cellular function. Mapping these interactions at the interface and revealing the key-stone residues, will provide important insight on how these structures combine and how it is possible to manage them, as well as improving or inhibiting their association^{152 153}.

One of the key features of these protein-protein interfaces is their sensitivity to mutations. This means that if we mutate a key interface residue by a residue alanine, there will be a significant variation in the protein-protein complex binding or association free energy. It has

been defined in the literature that if the increase in the binding free energy is above 4 kcal/mol, then the mutated residue is extremely important and it is called a hot-spot; if the energy increase upon mutation is between 2 and 4 kcal/mol, then this residue is relatively important for the protein-protein association and it is denominated a warm-spot and, finally, if the mutations originate a binding free energy variation below 2 kcal/mol, then the residue is not particularly relevant for the interaction and it is termed a null-spot^{27,154,155}.

Moreira *et al*²⁷ have developed a protocol (schematized in Figure 29), with low computational cost and high success rate that reproduces the quantitative free energy differences obtained from experimental mutagenesis procedures. This computational approach is transferable to any macromolecular complex and is a predictive model capable of anticipating the experimental results of mutagenesis, thus capable of guiding new experimental investigations. It is based on the all-atom methodology MMPBSA (Molecular Mechanics/Poisson-Boltzmann Surface Area)²⁹ to probe protein-protein interactions by calculating free energies combining molecular mechanics and a continuum solvent.

There are several web servers already available to compute this protein-protein interaction¹⁵⁶. Despite the large variety of available possibilities to study this kind of interactions, our method still proved to be better from the point of view of returning quantitative values of the binding free energy differences with molecular dynamics as the sampling method. The features contrast with the qualitative values and the analysis of only a few structures of other methods available. At the end of the Results section, values from other approaches obtained for our case studies are also presented for comparison.

3.6.3. Methodology

The first stage of CompASM involves the relaxation and equilibration of the wild-type complex that is being analyzed. This can be accomplished by a minimization procedure only or by a molecular dynamics simulation in a continuum medium, using the Generalized Born model. Subsequently, only the relaxed wild-type complex is divided in several alanine mutated complexes that were previously defined by the user, depending on the study that is intended to be performed. To the wild-type and mutated complexes it is then applied the MMPBSA script to calculate the respective binding free energy differences.

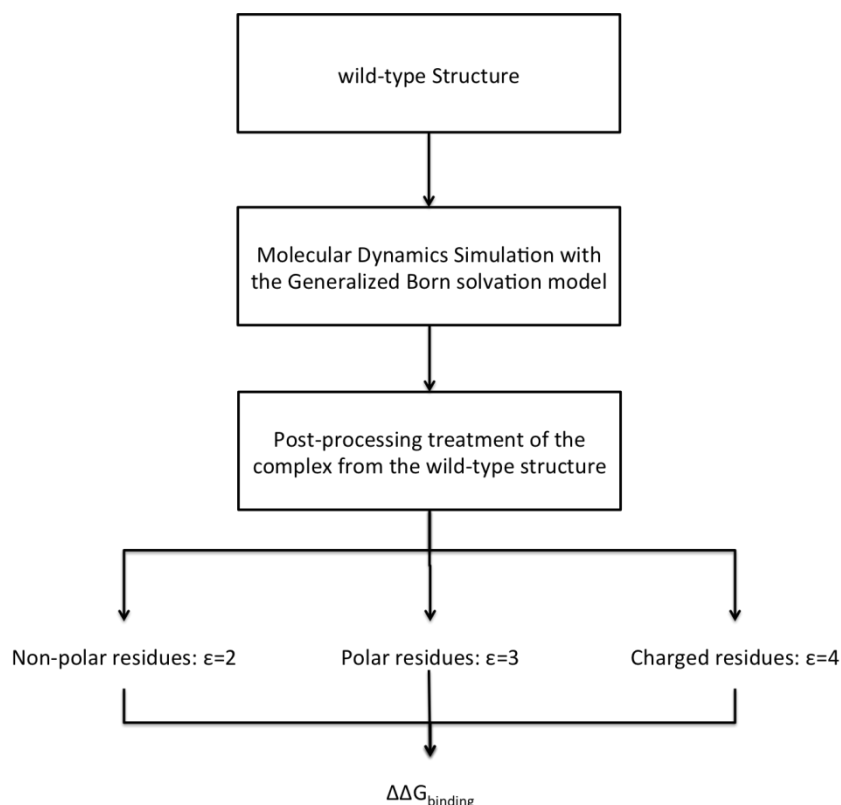


Figure 29: General algorithm of the CompASM procedure 27.

To generate the structure of the mutant complex, a simple truncation of the mutated side chain is carried out, replacing carbon atom C_γ with a hydrogen atom, and setting the C_β -H bond direction to that of the former C_β - C_γ . The corresponding binding free energy can be calculated using the thermodynamic cycle described in Methods section (2.1.4.1 Molecular Dynamics/Poisson-Boltzmann Surface Area).

For the energy calculations, CompASM attributes specific values to three internal dielectric constant values, which depend exclusively on the type of amino acid that is mutated. Therefore, for the charged amino acids (aspartic acid, glutamic acid, lysine, arginine, and histidine) a constant of 4 should be used, for the remaining polar residues (asparagine, glutamine, cysteine, tyrosine, serine, and threonine) not ionized at physiological pH the internal dielectric constant should be 3, and for the nonpolar amino acids (valine, leucine, isoleucine, phenylalanine, methionine, and tryptophan) the internal dielectric constant should be 2²⁷. The different internal dielectric constants account for the different degree of relaxation of the interface when different types of amino acids are mutated for alanine; the stronger the interactions these amino acids establish, the more extensive the relaxation should be, and the greater the internal dielectric constant value must be to mimic these effects.

However, for the sake of flexibility, in order to allow the user to set other values that he might see fit to the dielectric constants, we have added a feature in which the values of any number of dielectric constants can be changed from the default ones mentioned above or even added if necessary. This introduces flexibility to the plug-in and allows the user to improve the quality of the results, if considered necessary.

The nonpolar contribution to solvation free energy due to van der Waals interactions between the solute and the solvent and cavity formation, was modeled as a term that is dependent on the solvent-accessible surface area of the molecule. It was estimated using empirical relation (16),

$$\Delta G_{nonpolar} = \alpha A + \beta \quad (16)$$

where A is the solvent-accessible surface area that was estimated using the molsurf program, which is based on the idea primarily developed by Michael Connolly. Constants α and β are empirical, taking the values $0.00542 \text{ kcal } \text{\AA}^{-2} \text{ mol}^{-1}$ and $0.92 \text{ kcal mol}^{-1}$, respectively. The entropy term, obtained as the sum of translational, rotational, and vibrational components, was not calculated because it was assumed, on the basis of previous work^{55,56}, that its contribution to $\Delta\Delta G_{\text{binding}}$ is negligible.

The calculation of $\Delta\Delta G_{\text{binding}}$ is achieved applying several modules of the AMBER program³⁰, and despite the apparent simplicity, this kind of study can easily become cumbersome. Beyond the repetitive tasks, such as the generation of mutation structures and the input for the MMPBSA⁵⁶ calculation, the handling of a large amount of files can be a tricky job. Regarding these difficulties and combining the visual facilities provided by Visual Molecular Dynamics (VMD)³¹ with the extremely intuitive Graphical User Interface (GUI) and the AMBER molecular dynamics calculations, we propose a new VMD-AMBER plug-in, named CompASM, which allows even the non-expert user to perform easily Alanine Scanning Mutagenesis calculations.

3.6.4. Software Description

CompASM is a versatile tool created to study protein-protein interfaces, allowing the user to skip the repetitive task of creating input files and generating all necessary structures, as well as providing new options and procedures to perform a computational Alanine Scanning Mutagenesis experiment. The input file is based on an AMBER-format file. To maintain the functionality in almost all situations, we have divided the software in two main structures: a VMD plug-in GUI and an independent CORE.

The CompASM GUI was designed to drive the user through the different steps of ASM, allowing simultaneously all the freedom needed to treat all kind of structures that require different specifications. Beyond the ligand/receptor differentiation and mutation selection, this GUI allows the user to exclude or include non-protein structures (known as heteroatoms) as well as to insert their parameterisation files (.mol2 and/or .frcmod files). Another facility presented by CompASM GUI is the molecular minimisation/dynamics simulation set-up tab. Here, the user can set all variables to the values needed to submit a molecular dynamics simulation or just a minimization to AMBER. The user can add more variables directly in the input file following the instructions that pop up as one proceeds in the calculation. Another useful tool is the mutation selection by Non-Solvent Contact Area (NSCA-explained schematically in Figure 30) based selection (Figure 30).

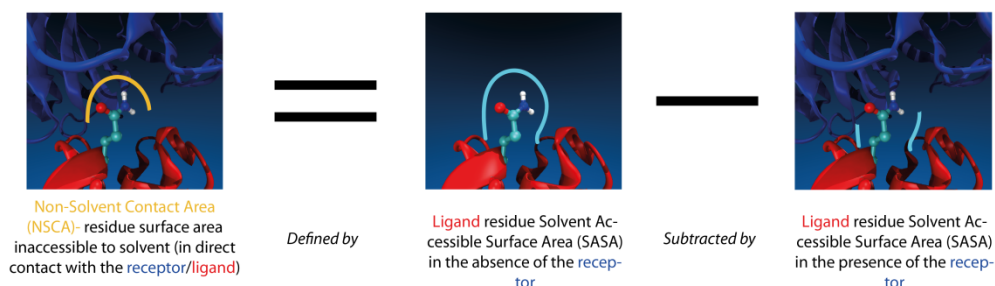


Figure 30: Scheme representing the Non-Solvent Contact Area (NSCA). This area intends to represent by which amount each residue is buried in the protein surface. All surface areas are evaluated by the “sasa measure” command present in the Visual Molecular Dynamics (VMD) software.

This is based on a “sasa measure” VMD command, and the residue is selected if the area that is in contact with another structure is larger than 40 Å² (more detail in supporting information). The final results are visualized in the VMD graphics window, using the same colour scheme of the summary and the detailed tables in the GUI.

The CompASM CORE is an independent set of procedures that load the structure, and perform the algorithm proposed by Moreira *et al.*²⁷ This CORE handles the files organization, performs the MMPBSA calculations and returns an output file (ASM.out) with all the information necessary to evaluate the final results. To improve the speed and make the best use of the computational resources, we parallelized the slower procedure, the MMPBSA calculation, running each independent calculation in different CPU cores. In the CORE's procedures, the molecular dynamics simulation is performed in sets of 1/10 of the total time requested by the user. In each set, the program checks if the coordinates from the structure (backbone only) are stabilized, evaluating several parameters of the linear regression of the Root Mean Square Deviation (RMSD) calculated by the ptraj AMBER tool. The simulation is considered stabilised if the slope of the straight, the standard deviation and the correlation

factor are in agreement with the values set by the user. For more information, please see the supporting information of this document. This software was developed in TCL/TK as the programming language, and it is available for the Amber8 and 9 versions, requiring the DELPHI package⁵¹⁻⁵³. The application of this software using higher versions of Amber is not presently possible, only due to the inexistence of the DELPHI package in the MMPBSA protocol (i.e. not available in the Amber 10 package).

3.6.5.Results

The data resulting from CompASM is displayed in a simplified table and in a more detailed table are displayed all the values obtained from the MMPBSA calculation, including the NSCA values. All this data can be analysed interactively in a VMD window, colouring the residue depending on the obtained score. Figure 31 shows the results obtained from CompASM concerning the protein-protein interface study of immunoglobulin complexed with an egg lysozyme, as an example.

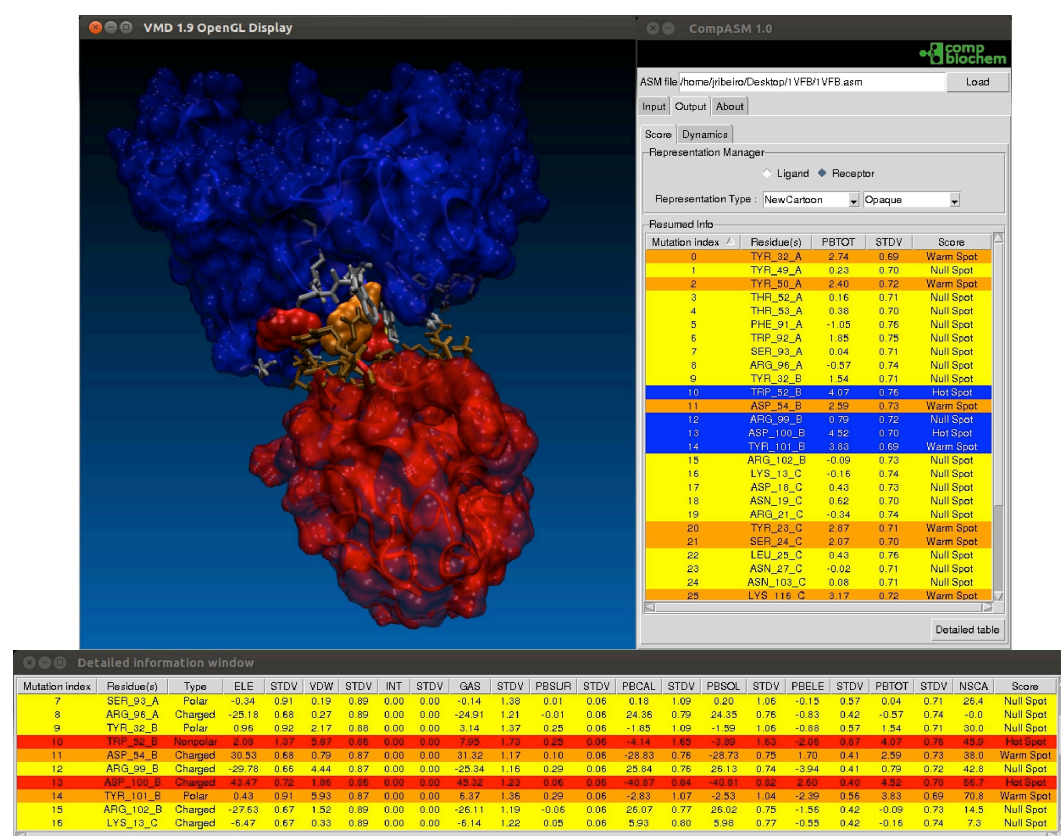


Figure 31: Results of the protein-protein interface study of immunoglobulin complexed with an egg lysozyme (detailed results in supporting information).

In this section we describe the validation process that was used to evaluate the performance of the CompASM software in the determination of the hot, warm and null-spots of three different protein-protein complexes. The results of this validation process are shown

in Table II. The structures analyzed are immunoglobulin complexed with an egg lysozyme (1VFB)¹³⁹; complexes that mediate bacterial cell division (1F47)¹⁵⁷ and human immunoglobulin IgG complexed with the C2 fragment of streptococcal protein G (1FCC)¹⁵⁸. Table II shows the values of NCSA, the experimental value of $\Delta\Delta G_{\text{binding}}$ of each mutation and the $\Delta\Delta G_{\text{binding}}$ evaluated by CompASM. All calculations involved molecular dynamics simulations using the default values of the CompASM GUI (Figure 31), and modifying only the MMPBSA frequency to achieve low standard deviation values. In the MMPBSA calculations of the 1VFB and 1F47 proteins, 50 structures were used, from the fourth and fifth nanosecond respectively of the simulation, and in the case of the 1FCC corresponding protein, 100 structures were used from the forth nanosecond. These same structures were also analyzed using well known web servers available, and the positive predictive value or specificity **P** (equation 17), true positive rate or sensitivity **R** (equation 18) and the F-measure test accuracy **F1** (equation 19) values were calculated and shown in Table III.

$$P = \frac{TP}{TP + FP} \quad (17)$$

$$R = \frac{TP}{TP + FN} \quad (18)$$

$$F1 = \frac{2PR}{P + R} \quad (19)$$

In equations (17), (18) and (19), *TP* corresponds to the correct computational prediction of the number of hot-spots (TP-true positive), i.e. when these residues are experimentally classified as hot-spots; *FP* corresponds to the number of computationally predicted hot-spots when these residues are experimentally classified as null-spots (FP-false positive); and finally *FN* corresponds to the number of computationally predicted null-spots when these residues are experimentally classified as hot-spots (FN-false negative). Therefore, the closer the values are to 100 the better they are, meaning in this case that the software is capable to predict correctly the numbers of hot-spots and null-spots.

To compare our software to those from web servers already available, we tested the same mutations in the proteins analyzed above using the following softwares: ISIS¹⁵⁹, Promate¹⁶⁰, Robetta¹⁶¹; K-FC2A and B¹⁶² and HotPoint¹⁶³.

Tables II and III demonstrate that the values resulting from CompASM are in the same range of those provided by experimental means, i.e. the CompASM software was able to detect all the hot-spots. This software was also sensitive enough to detect and to distinguish most warm- and null-spots. The results are very close to those obtained by Moreira *et al.*²⁷ in which the methodology of CompASM was based on. The small differences result from the

differences in the structures retrieved from the MD simulations and submitted to the MMPBSA calculations. CompASM has its own algorithm, which is used to control the molecular dynamics simulations. Once all the criteria are achieved, which the user can either set him/herself or accept the default, the MD simulation is stopped and the MMPBSA protocol starts. This algorithm is very useful because it makes the process straightforward and minimizes the computation time. Simultaneously, it allows the process to be reproducible. This makes it very handy not only for non-expert users, but also for more advanced users that can modify the criteria used in the CompASM GUI.

Table II - Results originated by CompASM. The Non-Solvent Contacting Area (NSCA) is calculated by the CompASM program.

Protein	MMPBSA Freq	Mutation	NSCA (Å ²)	Residue Type	Experimental	CompASM		SCORES	
					$\Delta\Delta G_{exp}$ (kcal/mol)	$\Delta\Delta G_{CompASM}$ (kcal/mol)	STDV	EXP	CompASM
1VFB	67	TYR_32_A	65.00	Polar	1.30	3.20	0.89	Null Spot	Warm Spot
1VFB	67	TYR_49_A	20.60	Polar	0.80	0.39	0.89	Null Spot	Null Spot
1VFB	67	TYR_50_A	37.80	Polar	0.40	2.40	0.91	Null Spot	Warm Spot
1VFB	67	THR_53_A	15.00	Polar	-0.23	0.26	0.87	Null Spot	Null Spot
1VFB	67	TRP_92_A	48.30	Nonpolar	1.71	1.71	0.94	Null Spot	Null Spot
1VFB	67	SER_93_A	26.40	Polar	0.11	-0.10	0.90	Null Spot	Null Spot
1VFB	67	TYR_32_B	30.00	Polar	0.50	1.23	0.88	Null Spot	Null Spot
1VFB	67	TRP_52_B	45.90	Nonpolar	1.23	3.90	0.89	Null Spot	Warm Spot
1VFB	67	ASP_54_B	38.00	Charged	1.95	2.55	0.94	Null Spot	Hot Spot
1VFB	67	ARG_99_B	42.80	Charged	0.47	0.86	0.93	Null Spot	Null Spot
1VFB	67	ASP_100_B	66.70	Charged	3.10	4.47	0.92	Warm Spot	Hot Spot
1VFB	67	TYR_101_B	70.80	Polar	4.00	4.04	0.91	Hot Spot	Hot Spot
1VFB	67	ASP_18_C	27.90	Charged	0.30	0.51	0.97	Null Spot	Null Spot
1VFB	67	ASN_19_C	69.00	Polar	0.40	0.78	0.89	Null Spot	Null Spot
1VFB	67	TYR_23_C	12.10	Polar	0.80	2.84	0.91	Null Spot	Warm Spot
1VFB	67	SER_24_C	52.90	Polar	0.70	2.12	0.91	Null Spot	Warm Spot
1VFB	67	LYS_116_C	57.60	Charged	0.70	3.06	0.94	Null Spot	Warm Spot
1VFB	67	THR_118_C	42.70	Polar	0.80	0.19	0.91	Null Spot	Null Spot
1VFB	67	ASP_119_C	43.70	Charged	1.00	3.48	0.96	Null Spot	Warm Spot
1VFB	67	VAL_120_C	9.70	Nonpolar	0.90	-0.39	0.92	Null Spot	Null Spot
1VFB	67	GLN_121_C	108.10	Polar	2.90	2.53	0.91	Warm Spot	Warm Spot
1VFB	67	ILE_124_C	28.70	Nonpolar	1.20	0.09	0.92	Null Spot	Null Spot
1VFB	67	ARG_125_C	48.80	Charged	1.80	3.21	0.93	Null Spot	Warm Spot
1VFB	67	LEU_129_C	23.60	Nonpolar	0.20	-9.93	0.93	Null Spot	Null Spot
1F47	67	ASP_4_A	43.20	Charged	0.69	-1.60	0.8	Null Spot	Null Spot
1F47	67	TYR_5_A	63.30	Polar	0.86	4.21	0.74	Null Spot	Hot Spot
1F47	67	LEU_6_A	81.70	Nonpolar	0.92	1.59	0.78	Null Spot	Null Spot
1F47	67	ASP_7_A	22.30	Charged	1.73	-0.46	0.78	Null Spot	Null Spot
1F47	67	ILE_8_A	50.70	Nonpolar	2.50	3.12	0.77	Warm Spot	Warm Spot
1F47	67	PHE_11_A	77.00	Nonpolar	2.44	3.60	0.79	Warm Spot	Warm Spot
1F47	67	LEU_12_A	53.10	Nonpolar	2.29	2.21	0.79	Warm Spot	Warm Spot
1FCC	35	GLU_27_C	45.80	Charged	4.90	10.91	1.17	Hot Spot	Hot Spot
1FCC	35	LYS_28_C	122.30	Charged	1.30	2.40	1.22	Null Spot	Warm Spot
1FCC	35	LYS_31_C	56.10	Charged	3.50	4.83	1.20	Warm Spot	Hot Spot
1FCC	35	ASN_35_C	61.80	Polar	2.40	0.95	1.12	Warm Spot	Null Spot
1FCC	35	ASP_40_C	44.20	Charged	0.30	0.49	1.23	Null Spot	Null Spot
1FCC	35	GLU_42_C	26.00	Charged	0.40	0.34	1.23	Null Spot	Null Spot
1FCC	35	TRP_43_C	37.80	Nonpolar	3.80	-0.13	1.09	Warm Spot	Null Spot

Table II shows the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values, as well as the accuracy of the different tested software.

Table III - Comparison of the values obtained using CompASM with those that result from the software/servers available. TP= true positive; TN= true negative; FN= false negative; FP= false positive; R= positive predictive; P= negative predictive and F1= F-measure.

Abrev.	ISIS	Promate	ROBETTA	KFC2-A	KFC2-B	HotPoint	CompASM
TP	1	0	7	5	6	6	8
TN	26	25	21	25	23	20	17
FP	2	3	7	3	5	8	11
FN	9	10	3	5	4	4	2
R	10	0	70	50	60	60	80
P	33	0	50	63	55	43	42
F1	15	0	58	56	57	50	55

Looking at the data resulting from CompASM, we can notice only 2 false negative values. There are, however, 11 false positives. In the majority of the complexes studied, an analysis of the dominant interactions suggests that van der Waals interactions and hydrophobic effects provide a reasonable basis for understanding binding affinities. In fact, the breakdown of equation (12) for all the cases studied with CompASM, shows that the DDE_{vdw} values are almost all positive indicating that the van der Waals interactions are favourable to complex binding, and that alanine mutation of the residues diminishes the vdW contacts at the interfaces. This is explained by the hydrophobic character of the interfaces.

To further examine the reliability and usefulness of CompASM, we have compared our predictions to the predictions of some of the more popular softwares that are available nowadays. The results are presented in Table III pointing to the fact that the methods Robetta, KFC2-A, KFC2-B, HotPoint and CompASM have similar performances, with the latter giving slightly better values. However, in a subsequent work¹⁶⁴, we provide a full study documenting the results obtained with CompASM based on a large number of structures, which points out to its good performance generating the better values amongst other methods.

We notice that only two of the above mentioned approaches, ours and Robetta's, provide quantitative values, which can be fully compared to the experimental ones, as opposed to qualitative values that classify the mutations between hot, warm and null spots only. However, in order to compare the results obtained with CompASM to the values returned by other softwares, we translated our values to the binominal qualitative classification (null and hot-spot) adopted by the tested softwares, which classify the residues in null-spots if the $\Delta\Delta G$ values are below 2 kcal/mol and hot-spots if the $\Delta\Delta G$ values are higher than 2 kcal/mol.

CompASM returns a value close to 100 (80) for the values of R (hot-spots prediction over false negatives). However, as CompASM is a software with which we can obtain quantitative values, we know directly how close is the residue to be a null or a hot-spot by comparing the obtained $\Delta\Delta G$ with the reference barriers of 2 and 4 kcal/mol, respectively.

3.6.6. Conclusions

Computational alanine scanning mutagenesis ²⁷ has proven to be an accurate means of detecting the residues that play an important role in protein-protein interfaces (hot-spots). Here, we present a VMD plug-in, CompASM, which facilitates the application of this approach thus simplifying the study of protein interfaces. CompASM guides the user through all ASM steps, from the ligand/receptor selection to the molecular dynamics simulation, and provides the visualization of the final results in a VMD window. This program can run either in local machines or in a cluster (multicomputer system). The GUI package is multiplatform and the CORE package works in a UNIX system (Mac OS and Linux). The CompASM package is distributed as an independent platform, with script code under the GNU Public License from <http://compbiochem.org/Software/compasm/Home.html>.

4. General Conclusions

The main purpose of the present work was to develop new bioinformatics tools to improve the Computer-Aided Drug Design process, reducing in this way the time and costs associated with the early steps of the drug development. The difficulties in applying these computational techniques are related with two major aspects: the numerous input files/commands preparations for a single job submission and the amount of information that is necessary to deal with in order to prepare these files and read the output results. These drawbacks can be overcome by expert users, specially by those with programming/scripting skills. In this work we presented four different tools aimed at calculating different properties or structural features, directed at being used by every type of users, even by the non-expert users.

In spite of the differences within the objectives of the developed tools, the guide lines behind them were basically the same. All developed software was based on the assumptions of being user-friendly, dismissing pre-knowledge from the user and requiring only a few clicks to perform the calculation; allowing a more sophisticated usage from experienced users; providing new algorithms capable of performing fast calculations and if possible, parallelizing the routines in order to return the maximum results from the available hardware. In the end, the final results must be visualized both in an intuitive graphical user interface and in the molecular visualizer VMD.

The first software to be developed was vsLab, which automatized the process of molecular docking, giving to the user the opportunity to perform molecular docking and/or virtual screening requiring only a small amount of clicks. Using vsLab is possible to perform the docking of a large set of ligands in the same protein and to visualize in the same program the final results. The inspection of the binding modes/positions can be performed by the analysis of the results table that presents the free energy and inhibition constant values calculated by AutoDock and simultaneously, the user can compare these values against the spatial arrangement of the complex (ligand and receptor) in the visualization window of VMD.

Taking into account this visual feature of the VMD software and the vast range of possibilities to select atoms and structure components (e.g. residues), we developed Chem-Path-Tracker. Rather than calculate a property, this software was developed to highlight “paths” between selected points, revealing cluster of atoms/residues that can constitute a chemical motif. For instance, this software turns easier the evaluation and the arrangement of the binding site revealing possibly existing networks of residues surrounding the ligand; the residues’ spatial sequence, forming cation- π interactions; the residues that interact with water in a water tunnel and the residues that contribute to the proton/electron transfer inside an enzyme. Again, contemplating only a few selections and parameters it is possible to

analyze a large number of pair-distances, revealing a wide range of possible interactions that would be very difficult to deduct manually.

Another developed plug-in dedicated to the structure analysis is VolArea. This tool was developed to calculate two different structural features: residues' exposed surfaces and the structural and empty volumes (e.g. cavities). VolArea revealed itself to be very useful and user-friendly when large surfaces need to be analyzed, calculating the contribution of each residue to the total surface. This usefulness is also linked to the volume calculation, where only the fraction of the structure that needs to be calculated is necessary to delimitate. The surface value is calculated using the VMD³¹ native command "measure sasa", while the volume algorithm was developed from scratch. This algorithm was designed to achieve a good compromise between accuracy and computational efficiency, which is why it was parallelized in order to use all the CPU available. Lately, a CUDA¹⁴⁹ version was developed to retrieve the maximum potential from the GPUs. These devices are well suited to solve problems containing variables with a high degree of independency, returning excellent results when transcribed to CUDA. Applying this new technology, the VolArea volume algorithm has shown a speed up of nearly 45 times. Despite the promising results, this algorithm requires special attention to the efficiency and consistence of the values. The existence of different devices and architectures raises portability and compatibility issues.

The last software to be presented in this thesis, named CompASM, was developed to guide the user through the steps needed for a Computational Alanine Scanning Mutagenesis calculation. Like vsLab, CompASM stands as a simplifying tool to perform complex and laborious calculations reflecting a fast learning process and reduced efforts from the user. Here it is possible to mutate large protein interfaces in order to evaluate the impact of the interacting residues in the free energy of the complex, determining their importance to the binding of the proteins. In terms of conception of the software, it encompasses many improvements in terms of programming skills and conceptual developments. These thoughts are reflected in a very agile interface that allows the user to perform a variety of operations, from the simple selection of ligands and receptors, to the inclusion and exclusion of residues, change/include molecular dynamics simulation parameters. The visualization of the final results was developed to make this GUI fully interactive. CompASM is composed by two independent packages, the GUI and CORE packages, in which the latter deals with all file management and all calculations using the AMBER³⁰ molecular package. The computational efficiency of the MMPBSA calculations was improved assigning each MMPBSA calculation to each CPU core. For instance, if the machine has an 8 core processor, than it is possible to calculate 8 mutations and respective $\Delta\Delta G_{\text{binding}}$ simultaneously. The independency of the GUI and CORE packages allows their use in different environments such as a regular Desktop or

Laptop to perform the results inspection using the GUI and run all heavy calculations in a Computer Cluster using only the CORE package.

Like any other software, the tools described in this thesis also require improvements and updates. For instance, vsLab is now on version 1.3, where several bugs were fixed and new options were included such as the possibility to change the pdbqt files before the AutoDock²⁶ calculations. The usage of a purchasable molecular package by CompASM requires the proper modification of the program in order to use an open source software, for instance NAMD¹⁶⁵, which also implies the development of a substitute routine of the DELPHI package⁵¹⁻⁵³ to calculate the electrostatic solvation free energy. As mentioned before, the development of the CUDA version of the VolArea plug-in revealed promising results, but still requires more work in order to ensure the proper portability and compatibility between devices. Regarding Chem-Path-Tracker, in spite of be the most recent software, the development of a mathematical expression in order to imprint a chemical sense to the distance evaluation routine could be important for the improvement of this tool.

The described tools are freely available for downloading in their websites:

vsLab: <http://www.fc.up.pt/pessoas/nscerque/vsLab/vLab/HomePage.html>;

Chem-Path-Tracker: <http://www2.fc.up.pt/PortoBioComp/database/doku.php?id=chem-path-tracker>;

VolArea: <http://www.fc.up.pt/portobiocomp/Software/Volarea/Home.html>; and

CompASM: <http://www.fc.up.pt/portobiocomp/Software/compasm/Home.html>.

Bibliography

- (1) Myers, S.; Baker, A. Drug discovery--an operating model for a new era. *Nature biotechnology* **2001**, *19*, 727-730.
- (2) DiMasi, J. A.; Hansen, R. W.; Grabowski, H. G. The price of innovation: new estimates of drug development costs. *J Health Econ* **2003**, *22*, 151-185.
- (3) Adams, C. P.; Brantner, V. V. Estimating the cost of new drug development: is it really 802 million dollars? *Health Aff (Millwood)* **2006**, *25*, 420-428.
- (4) Adams, C. P.; Brantner, V. V. Spending on new drug development1. *Health Econ* **2010**, *19*, 130-141.
- (5) Irwin, J. J.; Shoichet, B. K. ZINC--a free database of commercially available compounds for virtual screening. *J Chem Inf Model* **2005**, *45*, 177-182.
- (6) Available Chemical Directory (ACD, over 7 million compounds). <http://accelrys.com/products/databases/sourcing/available-chemicals-directory.html>.
- (7) National Cancer Institute compound database (NCI, over 260 000 compounds). <http://cactus.nci.nih.gov/download/nci/>.
- (8) MDDR Library (over 150 000 compounds). <http://accelrys.com/products/databases/bioactivity/mddr.html>.
- (9) Comprehensive Medicinal Chemistry. <http://accelrys.com/products/databases/bioactivity/comprehensive-medicinal-chemistry.html>.

- (10) Cambridge Structural Database (over 40 000 compounds). <http://www.ccdc.cam.ac.uk/Solutions/CSDSystem/Pages/CSD.aspx>.
- (11) Protein Data Bank. <http://www.rcsb.org/pdb/home/home.do>.
- (12) Berman, H.; Henrick, K.; Nakamura, H. Announcing the worldwide Protein Data Bank. *Nat Struct Biol* **2003**, *10*, 980-980.
- (13) Sali, A.; Kuriyan, J. Challenges at the frontiers of structural biology (Reprinted from Trends in Biochemical Science, vol 12, Dec., 1999). *Trends Cell Biol* **1999**, *9*, M20-M24.
- (14) Marti-Renom, M. A.; Stuart, A. C.; Fiser, A.; Sanchez, R.; Melo, F.; Sali, A. Comparative protein structure modeling of genes and genomes. *Annu Rev Bioph Biom* **2000**, *29*, 291-325.
- (15) Hillisch, A.; Pineda, L. F.; Hilgenfeld, R. Utility of homology models in the drug discovery process. *Drug Discov Today* **2004**, *9*, 659-669.
- (16) Homology-derived Secondary Structure of Proteins. <http://www.cmbi.kun.nl/gv/hssp>.
- (17) Sander, C.; Schneider, R. Database of Homology-Derived Protein Structures and the Structural Meaning of Sequence Alignment. *Proteins-Structure Function and Genetics* **1991**, *9*, 56-68.
- (18) Sousa, S. F.; Fernandes, P. A.; Ramos, M. J. Protein-ligand docking: Current status and future challenges. *Proteins* **2006**, *65*, 15-26.
- (19) Cerqueira, N. F. S. A.; Sousa, S. r.; Fernandes, P.; Ramos, M. o.: Virtual Screening of Compound Libraries. In *Ligand-Macromolecular Interactions in Drug Discovery*; Roque, A. I., Ed.; Methods in Molecular Biology; Humana Press, 2009; Vol. 572; pp 57-70.
- (20) Taft, C. A.; Da Silva, V. B.; Da Silva, C. H. Current topics in computer-aided drug design. *J Pharm Sci-Us* **2008**, *97*, 1089-1098.
- (21) Song, C. M.; Lim, S. J.; Tong, J. C. Recent advances in computer-aided drug design. *Brief Bioinform* **2009**, *10*, 579-591.
- (22) Huang, H. J.; Yu, H. W.; Chen, C. Y.; Hsu, C. H.; Chen, H. Y.; Lee, K. J.; Tsai, F. J.; Chen, C. Y. C. Current developments of computer-aided drug design. *J Taiwan Inst Chem E* **2010**, *41*, 623-635.
- (23) Ramos, M. J.; Fernandes, P. A. Atomic-Level Rational Drug Design. *Curr Comput-Aid Drug* **2006**, *2*, 57-81.
- (24) Liao, C. Z.; Sitzmann, M.; Pugliese, A.; Nicklaus, M. C. Software and resources for computational medicinal chemistry. *Future Med Chem* **2011**, *3*, 1057-1085.
- (25) Friedrichs, M. S.; Eastman, P.; Vaidyanathan, V.; Houston, M.; Legrand, S.; Beberg, A. L.; Ensign, D. L.; Bruns, C. M.; Pande, V. S. Accelerating Molecular Dynamic Simulation on Graphics Processing Units. *Journal of Computational Chemistry* **2009**, *30*, 864-872.
- (26) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry* **2009**, *30*, 2785-2791.
- (27) Moreira, I. S.; Fernandes, P. A.; Ramos, M. J. Computational alanine scanning mutagenesis—An improved methodological approach. *Journal of Computational Chemistry* **2007**, *28*, 644-654.
- (28) Clackson, T.; Wells, J. A. A Hot-Spot of Binding-Energy in a Hormone-Receptor Interface. *Science* **1995**, *267*, 383-386.
- (29) Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S. H.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; Donini, O.; Cieplak, P.; Srinivasan, J.; Case, D. A.; Cheatham, T. E. Calculating structures and free energies of complex molecules: Combining molecular mechanics and continuum models. *Accounts Chem Res* **2000**, *33*, 889-897.
- (30) Case, D. A.; Darden, T. A.; Cheatham; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Merz, K. M.; Pearlman, D. A.; Crowley, M.; Walker, R. C.; Zhang, W.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Wong, K. F.; Paesani, F.; Wu, X.; Brozell, S.; Tsui, V.; Gohlke, H.; Yang, L.; Tan, C.; Mongan, J.; Hornak, V.; Cui, G.; Beroza, P.; Mathews, D. H.; Schafmeister, C.; Ross, W. S.; Kollman, P. A.: *Amber* 9, 2006.
- (31) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J Mol Graphics* **1996**, *14*, 33-&.

- (32) Jensen, F.: *Introduction to Computational Chemistry*; John Wiley & Sons, 2006. pp. 12-69.
- (33) McCammon, J. A.; Gelin, B. R.; Karplus, M. Dynamics of folded proteins. *Nature* **1977**, 267, 585-590.
- (34) Alonso, H.; Bliznyuk, A. A.; Gready, J. E. Combining docking and molecular dynamic simulations in drug design. *Med Res Rev* **2006**, 26, 531-568.
- (35) Karplus, M.; McCammon, J. A. Molecular dynamics simulations of biomolecules. *Nat Struct Biol* **2002**, 9, 646-652.
- (36) Hansson, T.; Oostenbrink, C.; van Gunsteren, W. F. Molecular dynamics simulations. *Curr Opin Struct Biol* **2002**, 12, 190-196.
- (37) Bras, N. F.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J. Carbohydrate-binding modules from family 11: Understanding the binding mode of polysaccharides. *International Journal of Quantum Chemistry* **2008**, 108, 2030-2040.
- (38) Cerqueira, N. M. F. S. A.; Bras, N. F.; Fernandes, P. A.; Ramos, M. J. MADAMM: A multistaged docking with an automated molecular modeling protocol. *Proteins* **2009**, 74, 192-206.
- (39) Freddolino, P. L.; Arkhipov, A. S.; Larson, S. B.; McPherson, A.; Schulten, K. Molecular dynamics simulations of the complete satellite tobacco mosaic virus. *Structure* **2006**, 14, 437-449.
- (40) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. *Journal of Computational Chemistry* **2005**, 26, 1668-1688.
- (41) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. Charmm - a Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *Journal of Computational Chemistry* **1983**, 4, 187-217.
- (42) Christen, M.; Hunenberger, P. H.; Bakowies, D.; Baron, R.; Burgi, R.; Geerke, D. P.; Heinz, T. N.; Kastenholz, M. A.; Krautler, V.; Oostenbrink, C.; Peter, C.; Trzesniak, D.; Van Gunsteren, W. F. The GROMOS software for biomolecular simulation: GROMOS05. *Journal of Computational Chemistry* **2005**, 26, 1719-1751.
- (43) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J Comput Phys* **1977**, 23, 327-341.
- (44) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *J Chem Phys* **1983**, 79, 926-935.
- (45) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A Smooth Particle Mesh Ewald Method. *J Chem Phys* **1995**, 103, 8577-8593.
- (46) van Gunsteren, W. F.; Daura, X.; Mark, A. E. Computation of free energy. *Helv Chim Acta* **2002**, 85, 3113-3129.
- (47) Chipot, C.; Pearlman, D. A. Free energy calculations. The long and winding gilded road. *Mol Simulat* **2002**, 28, 1-12.
- (48) Straatsma, T. P.; Mccammon, J. A. Computational Alchemy. *Annual Review of Physical Chemistry* **1992**, 43, 407-435.
- (49) Beveridge, D. L.; Dicapua, F. M. Free-Energy Via Molecular Simulation - Applications to Chemical and Biomolecular Systems. *Annu Rev Biophys Bio* **1989**, 18, 431-492.
- (50) Massova, I.; Kollman, P. A. Combined molecular mechanical and continuum solvent approach (MM-PBSA/GBSA) to predict ligand binding. *Perspect Drug Discov* **2000**, 18, 113-135.
- (51) Honig, B.; Nicholls, A. Classical Electrostatics in Biology and Chemistry. *Science* **1995**, 268, 1144-1149.
- (52) Rocchia, W.; Sridharan, S.; Nicholls, A.; Alexov, E.; Chiabrera, A.; Honig, B. Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *Journal of Computational Chemistry* **2002**, 23, 128-137.

- (53) Rocchia, W.; Alexov, E.; Honig, B. Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions. *Journal of Physical Chemistry B* **2001**, *105*, 6507-6514.
- (54) Connolly, M. L. Analytical Molecular-Surface Calculation. *J Appl Crystallogr* **1983**, *16*, 548-558.
- (55) Massova, I.; Kollman, P. A. Computational alanine scanning to probe protein-protein interactions: A novel approach to evaluate binding free energies. *J Am Chem Soc* **1999**, *121*, 8133-8143.
- (56) Huo, S.; Massova, I.; Kollman, P. A. Computational alanine scanning of the 1 : 1 human growth hormone-receptor complex. *Journal of Computational Chemistry* **2002**, *23*, 15-27.
- (57) Sousa, S. F.; Ribeiro, A. J.; Coimbra, J. T.; Neves, R. P.; Martins, S. A.; Moorthy, N. S.; Fernandes, P. A.; Ramos, M. J. Protein-ligand docking in the new millennium - a retrospective of 10 years in the field. *Curr Med Chem* **2013**, *20*, 2296-2314.
- (58) Dominguez, C.; Boelens, R.; Bonvin, A. M. J. J. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. *J Am Chem Soc* **2003**, *125*, 1731-1737.
- (59) Bras, N. F.; Fernandes, P. A.; Ramos, M. J. Molecular dynamics studies on both bound and unbound renin protease. *Journal of biomolecular structure & dynamics* **2013**.
- (60) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking. *J Mol Biol* **1997**, *267*, 727-748.
- (61) Jones, G.; Willett, P.; Glen, R. C. Molecular Recognition of Receptor-Sites Using a Genetic Algorithm with a Description of Desolvation. *J Mol Biol* **1995**, *245*, 43-53.
- (62) Wang, R. X.; Lu, Y. P.; Wang, S. M. Comparative evaluation of 11 scoring functions for molecular docking. *J Med Chem* **2003**, *46*, 2287-2303.
- (63) SHOICHET, B.; BODIAN, D.; KUNTZ, I. MOLECULAR DOCKING USING SHAPE DESCRIPTORS. *J Comput Chem* **1992**, *13*, 380-397.
- (64) Kramer, B.; Rarey, M.; Lengauer, T. Evaluation of the FLEXX incremental construction algorithm for protein-ligand docking. *Proteins* **1999**, *37*, 228-241.
- (65) Friesner, R.; Banks, J.; Murphy, R.; Halgren, T.; Klicic, J.; Mainz, D.; Repasky, M.; Knoll, E.; Shelley, M.; Perry, J.; Shaw, D.; Francis, P.; Shenkin, P. Glide: A new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of Medicinal Chemistry* **2004**, *47*, 1739-1749.
- (66) Joseph-McCarthy, D.; Thomas, B.; Belmarsh, M.; Moustakas, D.; Alvarez, J. Pharmacophore-based molecular docking to account for ligand flexibility. *Proteins* **2003**, *51*, 172-188.
- (67) Jain, A. Surflex: Fully automatic flexible molecular docking using a molecular similarity-based search engine. *Journal of Medicinal Chemistry* **2003**, *46*, 499-511.
- (68) Cerqueira, N.; Bras, N.; Fernandes, P.; Ramos, M. MADAMM: A multistaged docking with an automated molecular modeling protocol. *Proteins* **2009**, *74*, 192-206.
- (69) Cummings, M.; DesJarlais, R.; Gibbs, A.; Mohan, V.; Jaeger, E. Comparison of automated docking programs as virtual screening tools. *Journal of Medicinal Chemistry* **2005**, *48*, 962-976.
- (70) Verdonk, M.; Cole, J.; Hartshorn, M.; Murray, C.; Taylor, R. Improved protein-ligand docking using GOLD. *Proteins* **2003**, *52*, 609-623.
- (71) Sousa, S. F.; Fernandes, P. A.; Ramos, M. J. Protein-ligand docking: Current status and future challenges. *Proteins* **2006**, *65*, 15-26.
- (72) Kontoyianni, M.; McClellan, L.; Sokol, G. Evaluation of docking performance: Comparative data on docking algorithms. *Journal of Medicinal Chemistry* **2004**, *47*, 558-565.
- (73) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J Mol Graphics* **1996**, *14*, 33-&.
- (74) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility. *J Comput Chem* **2009**, *30*, 2785-2791.

- (75) Morris, G.; Goodsell, D.; Halliday, R.; Huey, R.; Hart, W.; Belew, R.; Olson, A. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J Comput Chem* **1998**, *19*, 1639-1662.
- (76) Roehrig, U. F.; Grosdidier, A.; Zoete, V.; Michielin, O. Docking to Heme Proteins. *J Comput Chem* **2009**, *30*, 2305-2315.
- (77) Viegas, A.; Bras, N.; Cerqueira, N.; Fernandes, P.; Prates, J.; Fontes, C.; Bruix, M.; Romao, M.; Carvalho, A.; Ramos, M.; Macedo, A.; Cabrita, E. Molecular determinants of ligand specificity in family 11 carbohydrate binding modules - an NMR, X-ray crystallography and computational chemistry approach. *Febs J* **2008**, *275*, 2524-2535.
- (78) Bras, N.; Cerqueira, N.; Fernandes, P.; Ramos, M. Carbohydrate-binding modules from family 11: Understanding the binding mode of polysaccharides. *Int J Quantum Chem* **2008**, *108*, 2030-2040.
- (79) Chen, D.; Menche, G.; Power, T. D.; Sower, L.; Peterson, J. W.; Schein, C. H. Accounting for ligand-bound metal ions in docking small molecules on adenyllyl cyclase toxins. *Proteins* **2007**, *67*, 593-605.
- (80) Schames, J.; Henchman, R.; Siegel, J.; Sotriffer, C.; Ni, H.; McCammon, J. Discovery of a novel binding trench in HIV integrase. *Journal of Medicinal Chemistry* **2004**, *47*, 1879-1881.
- (81) Vaque, M.; Arola, A.; Aliagas, C.; Pujadas, G. BDT: an easy-to-use front-end application for automation of massive docking tasks and complex docking strategies with AutoDock. *Bioinformatics* **2006**, *22*, 1803-1804.
- (82) Hazai, E.; Bikadi, Z.; Zsila, F.; Lockwood, S. F. Molecular modeling of the non-covalent binding of the dietary tomato carotenoids lycopene and lycopyll, and selected oxidative metabolites with 5-lipoxygenase. *Bioorganic & Medicinal Chemistry* **2006**, *14*, 6859-6867.
- (83) Bystroff, C.; Simons, K. T.; Han, K. F.; Baker, D. Local sequence-structure correlations in proteins. *Curr Opin Biotech* **1996**, *7*, 417-421.
- (84) Kasuya, A.; Thornton, J. M. Three-dimensional structure analysis of PROSITE patterns. *J Mol Biol* **1999**, *286*, 1673-1691.
- (85) Kleywegt, G. J. Recognition of spatial motifs in protein structures. *J Mol Biol* **1999**, *285*, 1887-1897.
- (86) Gesto, D. S.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J. Unraveling the Enigmatic Mechanism of L-Asparaginase II with QM/QM Calculations. *J Am Chem Soc* **2013**, *135*, 7146-7158.
- (87) Oliveira, E. F.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J. Mechanism of Formation of the Internal Aldimine in Pyridoxal 5 '-Phosphate-Dependent Enzymes. *J Am Chem Soc* **2011**, *133*, 15496-15505.
- (88) Mota, C. S.; Rivas, M. G.; Brondino, C. D.; Moura, I.; Moura, J. J. G.; Gonzalez, P. J.; Cerqueira, N. M. F. S. A. The mechanism of formate oxidation by metal-dependent formate dehydrogenases. *J Biol Inorg Chem* **2011**, *16*, 1255-1268.
- (89) Siegbahn, P. E. M.; Himo, F. The quantum chemical cluster approach for modeling enzyme reactions. *Wires Comput Mol Sci* **2011**, *1*, 323-336.
- (90) Alberto, M. E.; Marino, T.; Ramos, M. J.; Russo, N. Atomistic details of the Catalytic Mechanism of Fe(III)-Zn(II) Purple Acid Phosphatase. *J Chem Theory Comput* **2010**, *6*, 2424-2433.
- (91) Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J. Computational Mechanistic Studies Addressed to the Transimination Reaction Present in All Pyridoxal 5 '-Phosphate-Requiring Enzymes. *J Chem Theory Comput* **2011**, *7*, 1356-1368.
- (92) Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Gonzalez, P. A.; Moura, J. J. G.; Ramos, M. J. The Sulfur Shift: An Activation Mechanism for Periplasmic Nitrate Reductase and Formate Dehydrogenase. *Inorg. Chem.* **2013**, *52* 10766-10772.
- (93) Snyder, F. F.; Yuan, R. G.; Bin, J. C.; Carter, K. L.; McKay, D. J. Human guanine deaminase: Cloning, expression and characterisation. *Adv Exp Med Biol* **2000**, *486*, 111-114.

- (94) Thoma, R.; Schulz-Gasch, T.; D'Arcy, B.; Benz, J.; Aebi, J.; Dehmlow, H.; Hennig, M.; Stihle, M.; Ruf, A. Insight into steroid scaffold formation from the structure of human oxidosqualene cyclase. *Nature* **2004**, 432, 118-122.
- (95) Huff, M. W.; Telford, D. E. Lord of the rings - the mechanism for oxidosqualene : lanosterol cyclase becomes crystal clear. *Trends Pharmacol Sci* **2005**, 26, 335-340.
- (96) Hirschi, J.; Singleton, D. A.; Scott, A. I. Mechanism of oxidosqualene cyclase. *Abstr Pap Am Chem S* **2005**, 229, U402-U402.
- (97) Wendt, K. U. Enzyme mechanisms for triterpene cyclization: New pieces of the puzzle. *Angew Chem Int Edit* **2005**, 44, 3966-3971.
- (98) Istvan, E. S.; Deisenhofer, J. Structural mechanism for statin inhibition of HMG-CoA reductase. *Science* **2001**, 292, 1160-1164.
- (99) Gallivan, J. P.; Dougherty, D. A. Cation-pi interactions in structural biology. *P Natl Acad Sci USA* **1999**, 96, 9459-9464.
- (100) Devos, A. M.; Ultsch, M.; Kossiakoff, A. A. Human Growth-Hormone and Extracellular Domain of Its Receptor - Crystal-Structure of the Complex. *Science* **1992**, 255, 306-312.
- (101) Gonen, T.; Walz, T. The structure of aquaporins. *Quarterly Reviews of Biophysics* **2006**, 39, 361-396.
- (102) Tani, K.; Mitsuma, T.; Hiroaki, Y.; Kamegawa, A.; Nishikawa, K.; Tanimura, Y.; Fujiyoshi, Y. Mechanism of Aquaporin-4's Fast and Highly Selective Water Conduction and Proton Exclusion. *J Mol Biol* **2009**, 389, 694-706.
- (103) Chakrabarti, N.; Roux, B.; Pomes, R. Structural determinants of proton blockage in aquaporins. *J Mol Biol* **2004**, 343, 493-510.
- (104) Cerqueira, N. M. F. S. A.; Pereira, S.; Fernandes, P. A.; Ramos, M. J. Overview of ribonucleotide reductase inhibitors: An appealing target in anti-tumour therapy. *Curr Med Chem* **2005**, 12, 1283-1294.
- (105) Seyedsayamdost, M. R.; Yee, C. S.; Reece, S. Y.; Nocera, D. G.; Stubbe, J. pH Rate profiles of FnY356-R2s (n = 2, 3, 4) in Escherichia coli ribonucleotide reductase: evidence that Y356 is a redox-active amino acid along the radical propagation pathway. *J Am Chem Soc* **2006**, 128, 1562-1568.
- (106) Pereira, S.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J. Computational studies on class I ribonucleotide reductase: Understanding the mechanisms of action and inhibition of a cornerstone enzyme for the treatment of cancer. *Eur Biophys J* **2006**, 35, 125-135.
- (107) Case, D. A.; Darden, T. A.; Cheatham, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Merz, K. M.; Pearlman, D. A.; Crowley, M.; Walker, R. C.; Zhang, W.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Wong, K. F.; Paesani, F.; Wu, X.; Brozell, S.; Tsui, V.; Gohlke, H.; Yang, L.; Tan, C.; Mongan, J.; Hornak, V.; Cui, G.; Beroza, P.; Matthews, D. H.; Schafmeister, C.; Ross, W. S.; Kollman, P. A.: AMBER9. San Francisco, 2006.
- (108) Hou, T. J.; Xu, X. J. ADME evaluation in drug discovery. 2. Prediction of partition coefficient by atom-additive approach based on atom-weighted solvent accessible surface areas (vol 43, pg 1058, 2003). *J Chem Inf Comp Sci* **2004**, 44, 1516-1516.
- (109) Wang, J. M.; Hou, T. J. Develop and Test a Solvent Accessible Surface Area-Based Model in Conformational Entropy Calculations. *J Chem Inf Model* **2012**, 52, 1199-1212.
- (110) Hou, T. J.; Qiao, X. B.; Zhang, W.; Xu, X. J. Empirical aqueous solvation models based on accessible surface areas with implicit electrostatics. *Journal of Physical Chemistry B* **2002**, 106, 11295-11304.
- (111) Otyepka, M.; Petrek, M.; Banas, P.; Kosinova, P.; Koca, J.; Damborsky, J. CAVER: a new tool to explore routes from protein clefts, pockets and cavities. *Bmc Bioinformatics* **2006**, 7.
- (112) Ho, B. K.; Gruswitz, F. HOLLOW: Generating Accurate Representations of Channel and Interior Surfaces in Molecular Structures. *Bmc Struct Biol* **2008**, 8.

- (113) Voss, J.; Shi, Q.; Jacobsen, H. S.; Zamponi, M.; Lefmann, K.; Vegge, T. Hydrogen dynamics in Na₃AlH₆: A combined density functional theory and quasielastic neutron scattering study. *Journal of Physical Chemistry B* **2007**, *111*, 3886-3892.
- (114) Ullmann, G. M.; Till, M. S. McVol - A program for calculating protein volumes and identifying cavities by a Monte Carlo algorithm. *Journal of Molecular Modeling* **2010**, *16*, 419-429.
- (115) Schmidtke, P.; Bidon-Chanal, A.; Luque, F. J.; Barril, X. MDpocket: open-source cavity detection and characterization on molecular dynamics trajectories. *Bioinformatics* **2011**, *27*, 3276-3285.
- (116) Chandra, N.; Yeturu, K. PocketMatch: A new algorithm to compare binding sites in protein structures. *Bmc Bioinformatics* **2008**, *9*.
- (117) Voss, N. R.; Gerstein, M.; Steitz, T. A.; Moore, P. B. The geometry of the ribosomal polypeptide exit tunnel. *J Mol Biol* **2006**, *360*, 893-906.
- (118) .
- (119) Lee, B.; Richards, F. M. Interpretation of Protein Structures - Estimation of Static Accessibility. *J Mol Biol* **1971**, *55*, 379-&.
- (120) Futamura, N.; Aluru, S.; Ranjan, D.; Hariharan, B. Efficient parallel algorithms for solvent accessible surface area of proteins. *Ieee T Parall Distr* **2002**, *13*, 544-555.
- (121) Murphy, L. R.; Matubayasi, N.; Payne, V. A.; Levy, R. M. Protein hydration and unfolding - insights from experimental partial specific volumes and unfolded protein models. *Folding & Design* **1998**, *3*, 105-118.
- (122) Marquart, M.; Walter, J.; Deisenhofer, J.; Bode, W.; Huber, R. The Geometry of the Reactive Site and of the Peptide Groups in Trypsin, Trypsinogen and Its Complexes with Inhibitors. *Acta Crystallogr B* **1983**, *39*, 480-490.
- (123) Howlin, B.; Moss, D. S.; Harris, G. W. Segmented Anisotropic Refinement of Bovine Ribonuclease-a by the Application of the Rigid-Body Tls Model. *Acta Crystallogr A* **1989**, *45*, 851-861.
- (124) Hodsdon, J. M.; Brown, G. M.; Sieker, L. C.; Jensen, L. H. Refinement of Triclinic Lysozyme .1. Fourier and Least-Squares Methods. *Acta Crystallogr B* **1990**, *46*, 54-62.
- (125) Takano, T. Structure of Myoglobin Refined at 2.0 a Resolution .1. Crystallographic Refinement of Metmyoglobin from Sperm Whale. *J Mol Biol* **1977**, *110*, 537-568.
- (126) Dreusicke, D.; Karplus, P. A.; Schulz, G. E. Refined Structure of Porcine Cytosolic Adenylate Kinase at 2.1-a Resolution. *J Mol Biol* **1988**, *199*, 359-371.
- (127) Pickersgill, R. W.; Harris, G. W.; Garman, E. Structure of Monoclinic Papain at 1.60-a Resolution. *Acta Crystallogr B* **1992**, *48*, 59-67.
- (128) Epp, O.; Lattman, E. E.; Schiffer, M.; Huber, R.; Palm, W. The molecular structure of a dimer composed of the variable portions of the Bence-Jones protein REI refined at 2.0-A resolution. *Biochemistry* **1975**, *14*, 4943-4952.
- (129) Reeke, G. N., Jr.; Becker, J. W.; Edelman, G. M. The covalent and three-dimensional structure of concanavalin A. IV. Atomic coordinates, hydrogen bonding, and quaternary structure. *The Journal of biological chemistry* **1975**, *250*, 1525-1547.
- (130) Meyer, E.; Cole, G.; Radhakrishnan, R.; Epp, O. Structure of Native Porcine Pancreatic Elastase at 1.65 a Resolution. *Acta Crystallogr B* **1988**, *44*, 26-38.
- (131) Ploegman, J. H.; Drent, G.; Kalk, K. H.; Hol, W. G. J. Structure of Bovine Liver Rhodanese .1. Structure Determination at 2.5 a Resolution and a Comparison of Conformation and Sequence of Its 2 Domains. *J Mol Biol* **1978**, *123*, 557-594.
- (132) Teplyakov, A.; Wilson, K. S.; Orioli, P.; Mangani, S. High-Resolution Structure of the Complex between Carboxypeptidase-a and L-Phenyl Lactate. *Acta Crystallogr D* **1993**, *49*, 534-540.
- (133) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J Cheminformatics* **2011**, *3*.
- (134) Ousterhout, J. K. *Tcl: An Embeddable Command Language* 1989.
- (135) D.A. Case, T. A. D., T.E. Cheatham, III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, K.M. Merz, B. Wang, D.A. Pearlman, M. Crowley, S. Brozell, V. Tsui, H. Gohlke, J.

Mongan, V. Hornak, G. Cui, P. Beroza, C. Schafmeister, J.W. Caldwell, W.S. Ross, and P.A. Kollman. AMBER 8. *University of California, San Francisco* **2004**.

(136) Lindahl, E.; Hess, B.; van der Spoel, D. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling* **2001**, 7, 306-317.

(137) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. CHARMM: The Biomolecular Simulation Program. *Journal of Computational Chemistry* **2009**, 30, 1545-1614.

(138) Nelson, M. T.; Humphrey, W.; Gursoy, A.; Dalke, A.; Kale, L. V.; Skeel, R. D.; Schulten, K. NAMD: A parallel, object oriented molecular dynamics program. *Int J Supercomput Ap* **1996**, 10, 251-268.

(139) Bhat, T. N.; Bentley, G. A.; Boulot, G.; Greene, M. I.; Tello, D.; Dallacqua, W.; Souchon, H.; Schwarz, F. P.; Mariuzza, R. A.; Poljak, R. J. Bound Water-Molecules and Conformational Stabilization Help Mediate an Antigen-Antibody Association. *P Natl Acad Sci USA* **1994**, 91, 1089-1093.

(140) Bogan, A. A.; Thorn, K. S. Anatomy of hot spots in protein interfaces. *J Mol Biol* **1998**, 280, 1-9.

(141) Sharma, S. K.; Ramsey, T. M.; Bair, K. W. Protein-protein interactions: lessons learned. *Curr Med Chem Anticancer Agents* **2002**, 2, 311-330.

(142) Ribeiro, J. V.; Cerqueira, N. M. F. S. A.; Moreira, I. S.; Fernandes, P. A.; Ramos, M. J. CompASM: an Amber-VMD alanine scanning mutagenesis plug-in. *Theor Chem Acc* **2012**, 131.

(143) Viegas, A.; Bras, N. F.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Prates, J. A. M.; Fontes, C. M. G. A.; Bruix, M.; Romao, M. J.; Carvalho, A. L.; Ramos, M. J.; Macedo, A. L.; Cabrita, E. J. Molecular determinants of ligand specificity in family 11 carbohydrate binding modules - an NMR, X-ray crystallography and computational chemistry approach. *Febs J* **2008**, 275, 2524-2535.

(144) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field (SCF) Implementation (vol 5, pg 1004, 2009). *J Chem Theory Comput* **2009**, 5, 3138-3138.

(145) Ufimtsev, I. S.; Martinez, T. J. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J Chem Theory Comput* **2008**, 4, 222-231.

(146) Petruk, A. A.; Defelipe, L. A.; Limardo, R. G. R.; Bucci, H.; Marti, M. A.; Turjanski, A. G. Molecular Dynamics Simulations Provide Atomistic Insight into Hydrogen Exchange Mass Spectrometry Experiments. *J Chem Theory Comput* **2013**, 9, 658-669.

(147) Liu, W. G.; Schmidt, B.; Voss, G.; Muller-Wittig, W. Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA. *Comput Phys Commun* **2008**, 179, 634-641.

(148) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. GPU-accelerated molecular modeling coming of age. *Journal of Molecular Graphics & Modelling* **2010**, 29, 116-125.

(149) NVIDIA: CUDA C PROGRAMMING GUIDE. 2012.

(150) Rodrigues, C.; Hardy, D.; Stone, J.; Schulten, K.; Wen, M. In *Tilte*, Ischia, Italy2008; ACM.

(151) Fukunari, A.; Okamoto, K.; Nishino, T.; Eger, B. T.; Pai, E. F.; Kamezawa, M.; Yamada, I.; Kato, N. Y-700 [1-[3-cyano-4-(2,2-dimethylpropoxy)phenyl]-1H-pyrazole-4-carboxylic acid]: A potent xanthine oxidoreductase inhibitor with hepatic excretion. *J Pharmacol Exp Ther* **2004**, 311, 519-528.

(152) Bordner, A. J.; Abagyan, R. Statistical analysis and prediction of protein-protein interfaces. *Proteins* **2005**, 60, 353-366.

- (153) Thorn, K. S.; Bogan, A. A. ASEdb: a database of alanine mutations and their effects on the free energy of binding in protein interactions. *Bioinformatics* **2001**, *17*, 284-285.
- (154) Ramos, M. J.; Moreira, I. S.; Fernandes, P. A. Hot spots-A review of the protein-protein interface determinant amino-acid residues. *Proteins* **2007**, *68*, 803-812.
- (155) Moreira, I. S.; Fernandes, P. A.; Ramos, M. J. Hot spot occlusion from bulk water: A comprehensive study of the complex between the lysozyme HEL and the antibody FVD1.3. *Journal of Physical Chemistry B* **2007**, *111*, 2697-2706.
- (156) Fernandez-Recio, J. Prediction of protein binding sites and hot spots. *Wires Comput Mol Sci* **2011**, *1*, 680-698.
- (157) Somers, W. S.; Mosyak, L.; Zhang, Y.; Glasfeld, E.; Haney, S.; Stahl, M.; Seehra, J. The bacterial cell-division protein ZipA and its interaction with an FtsZ fragment revealed by X-ray crystallography. *Embo Journal* **2000**, *19*, 3179-3191.
- (158) Sauereriksson, A. E.; Kleywegt, G. J.; Uhl, M.; Jones, T. A. Crystal-Structure of the C2 Fragment of Streptococcal Protein-G in Complex with the Fc Domain of Human-IgG. *Structure* **1995**, *3*, 265-278.
- (159) Ofra, Y.; Rost, B. ISIS: interaction sites identified from sequence. *Bioinformatics* **2007**, *23*, E13-E16.
- (160) Neuvirth, H.; Raz, R.; Schreiber, G. ProMate: A structure based prediction program to identify the location of protein-protein binding sites. *J Mol Biol* **2004**, *338*, 181-199.
- (161) Kortemme, T.; Baker, D. A simple physical model for binding energy hot spots in protein-protein complexes. *P Natl Acad Sci USA* **2002**, *99*, 14116-14121.
- (162) Darnell, S. J.; Page, D.; Mitchell, J. C. An automated decision-tree approach to predicting protein interaction hot spots. *Proteins* **2007**, *68*, 813-823.
- (163) Tuncbag, N.; Keskin, O.; Gursoy, A. HotPoint: hot spot prediction server for protein interfaces. *Nucleic Acids Research* **2010**, *38*, W402-W406.
- (164) Results to be published.
- (165) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Schulten, K. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry* **2005**, *26*, 1781-1802.

