NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

INTERDISCIPLINARY POSTGRADUATE PROGRAM
"INFORMATION TECHNOLOGIES IN MEDICINE AND BIOLOGY"

MASTER THESIS

Automating Free Energy Perturbation Calculations For Drug Design

Stamatia N. Zavitsanou

Supervisor: **Zoe Cournia**, Researcher - Assistant Professor Level

ATHENS
JULY 2018

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
"ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗΝ ΙΑΤΡΙΚΗ ΚΑΙ ΤΗ ΒΙΟΛΟΓΙΑ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αυτοματοποίηση υπολογισμού ελεύθερης ενέργειας πρόσδεσης για τον σχεδιασμό φαρμάκων

Σταματία Ν. Ζαβιτσάνου

Επιβλέπουσα: **Ζωή Κούρνια**, Ερευνήτρια Γ’

ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2018

MASTER THESIS

Free energy perturbation calculations for drug design

Stamatia N. Zavitsanou
S.N.: PIV0153

SUPERVISOR: **Zoe Cournia**, Researcher - Assistant Professor Level

EXAMINATION COMMITTEE:	**Zoe Cournia**, Researcher - Assistant Professor Level
	Stavros Perantonis, Research Director
	Ioannis Emiris, Professor Level

JULY 2018

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αυτοματοποίηση υπολογισμού ελεύθερης ενέργειας πρόσδεσης για τον σχεδιασμό φαρμάκων

Σταματία Ν. Ζαβιτσάνου
Α.Μ.: ΤΠΙΒ0153

ΕΠΙΒΛΕΠΟΥΣΑ: **Ζωή Κούρνια,** Ερευνήτρια Γ’

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Ζωή Κούρνια, Ερευνήτρια Γ’
Σταύρος Περαντώνης, Διευθυντής Έρευνας
Ιωάννης Εμίρης, Καθηγητής

ΙΟΥΛΙΟΣ 2018

ABSTRACT

The advent of technological advances of computer-aided drug design has streamlined the drug design process, rendering it more cost- and time-efficient. Nowadays, rational structure-based drug design may quantify underlying molecular interactions involved in ligand-protein binding by utilizing the 3D structure of the therapeutic target in the process. Accurate quantification of these interactions can aid the optimization of binding affinity,selectivity, and other off -target interactions, which are a critical part of hit-to-lead and lead optimization efforts in drug discovery. One of the most important tasks in the lead optimization phase of the drug design process is to predict, among a series of lead candidates, which ones will bind more strongly to the therapeutic target. In this direction, relative binding free energy methodologies have been developed, which rely on physics-based molecular simulations and rigorous statistical mechanics to calculate the differences in the free energy of binding between a parent candidate drug and analogues. For example, Free Energy Perturbation (FEP) calculations coupled with Molecular Dynamics (MD) simulations calculate the free energy difference between an initial (reference) and an analog (target) molecule to an average of a function of their energy difference evaluated by sampling for the initial state.

Such methodologies have shown significant potential in the lead optimization process, however, they have been limited by technical challenges such as manual creation of large numbers of input files to setup/run/analyze free energy simulations. Automating free energy perturbation calculations would streamline the use of FEP calculations and would be a step forward to delivering high throughput calculations for accurate predictions of relative binding affinities before a compound is synthesized, and consequently save enormous experimental and human resources.

In this thesis, an algorithm called FEPrepare, which automates the set up procedure for relative binding free energy simulations has been designed and implemented as the first web-based server. The web server automates the set-up procedure for FEP calculations within the context of NAMD, one of the major MD engines. The user has to upload the structure files to the web-server. The algorithm is written in Python, utilizes the structure files uploaded by the user in order to perform atom renaming, and partial charge redistribution and create the necessary input files for VMD, a molecular viewer program, that can be used to help set up NAMD simulations and to help analyse and visualize NAMD output, to generate all needed files for the calculations. After the algorithm confirms compatibility of the required files with NAMD, it provides the user with everything needed to run a simulation.

Relative binding free energy calculations in drug design have proven very effective in facilitating the lead optimization process both time and cost efficient. The automation of Free Energy Perturbation calculations to provide access to large-scale simulations for lead optimization has been presented in this thesis.

SUBJECT AREA: Computational Chemistry

KEYWORDS: free energy, drug design, NAMD, dual topology file, ligands, web development

ΠΕΡΙΛΗΨΗ

Η διαδικασία σχεδιασμού φαρμάκων έχει βελτιστοποιηθεί με τη βοήθεια των ηλεκτρονικών υπολογιστών, έχοντας γίνει πιο αποδοτική από πλευράς κόστους και χρόνου. Σήμερα, χρησιμοποιώντας την τρισδιάστατη δομή του θεραπευτικού στόχου ο ορθολογιστικός σχεδιασμός φαρμάκων μπορεί να ποσοτικοποιήσει τις μοριακές αλληλεπιδράσεις που εμπλέκονται στη δέσμευση προσδέτη-πρωτεΐνης. Η ακριβής αυτή ποσοτικοποίηση βοηθά στην βελτιστοποίηση αλληλεπιδράσεων εκτός στόχου, οι οποίες παίζουν σημαντικό ρόλο στην ανίχνευση των βέλτιστων προσδετών. Ένα από τα πιο σημαντικά καθήκοντα στον σχεδιασμό φαρμάκων είναι να προβλέψουμε μεταξύ μιας σειράς υποψήφιων ποιά από αυτά θα δεσμευτούν καλύτερα στον θεραπευτικό στόχο. Σε αυτή την κατεύθυνση έχουν αναπτυχθεί μεθοδολογίες σχετικής δέσμευσης της ελεύθερης ενέργειας, οι οποίες βασίζονται σε μοριακές προσομοιώσεις, στη φυσική και στην αυστηρή στατιστική μηχανική για τον υπολογισμό των διαφορών στην ελεύθερη ενέργεια σύνδεσης μεταξύ ενός γονικού υποψήφιου φαρμάκου και αναλόγων του. Για παράδειγμα, οι υπολογισμοί της Ελεύθερης Ενεργειακής Διαταραχής (FEP) σε συνδυασμό με τις προσομοιώσεις Μοριακής Δυναμικής (MD) υπολογίζουν την ελεύθερη διαφορά ενέργειας μεταξύ ενός αρχικού και ενός τελικού μορίου.

Αυτές οι μεθοδολογίες έχουν σημαντικές δυνατότητες, ωστόσο έχουν περιοριστεί από τεχνικές προκλήσεις όπως η χειροκίνητη δημιουργία μεγάλου αριθμού αρχείων εισόδου για την εγκατάσταση / εκτέλεση / ανάλυση ελεύθερων προσομοιώσεων ενέργειας. Η αυτοματοποίηση των υπολογισμών της διαταραχής της ελεύθερης ενέργειας απλοποιεί τη χρήση των υπολογισμών FEP και παρέχει υπολογισμούς υψηλής απόδοσης για ακριβείς προβλέψεις πριν από την σύνθεση μιας ένωσης και συνεπώς εξοικονομεί τεράστιο χρόνο και κόστος.

Σε αυτή τη διατριβή περιγράφεται ένας αλγόριθμος, ονομαζόμενος FEPrepare, ο οποίος αυτοματοποιεί τη διαδικασία στησίματος για σχετικές δεσμευτικές προσομοιώσεις ελεύθερης ενέργειας μέσω ενός ιστόποπου. Αυτοματοποιείται τη διαδικασία του στησίματος για υπολογισμούς FEP στο πλαίσιο του NAMD, ενός από τους σημαντικότερους μηχανισμούς MD. Ο χρήστης ανεβάζει τα αρχεία δομής πρωτεΐνης και των προσδεμάτων, ο αλγόριθμος ο οποίος είναι γραμμένος σε Python χρησιμοποιεί τα αρχεία αυτά για να μετονομάσει τα άτομα, να αναδιανείμει τα φορτία των ατόμων και να δημιουργήσει τα απαραίτητα αρχεία για το VMD, ένα πρόγραμμα μοριακής προβολής που μπορεί να χρησιμοποιηθεί για τη δημιουργία προσομοίωσης του NAMD και να βοηθήσει στην ανάλυση των δεδομένων που παράγει το NAMD, για να παράξει τα αρχεία που χρειάζονται, να κάνει την υδρόλυση και τον ιονισμό. Αφού ο αλγόριθμος επιβεβαιώσει οτι όλα τα αρχεία είναι συμβατά με το NAMD τα παρέχει στον χρήστη.

Οι υπολογισμοί σχετικής ελεύθερης ενέργειας στον σχεδιασμό φαρμάκων έχουν αποδειχθεί πολύ χρήσιμοι καθώς κάνουν την διαδικασία της βελτιστοποίησης πολύ πιο γρήγορη και φθηνή. Σε αυτή τη διπλωματική παρουσιάζεται η αυτοματοποιήση υπολογισμών ελεύθερης ενέργειας πρόσδεσης, για τη διαδικασία της βελτιστοποίησης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Υπολογιστική Χημεία

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ελεύθερη ενέργεια, σχεδιασμός φαρμάκων, NAMD, αρχείο διπλής τοπολογίας, ανάπτυξη διαδικτυακού εργαλείου

To my family

.

AKNOWLEDGMENTS

I would like to take this opportunity to acknowledge those who have helped me complete this thesis. First and foremost, I would like to express my gratitude to my supervisor, Dr. Zoe Cournia, Researcher - Assistant Professor Level, at the Biomedical Research Foundation of the Academy of Athens (BRFAA) - her encouragement, support, and thoughtful advice have been immensely valuable for me and my academic steps. Her guidance helped me in my research and in writing this thesis.

I would like to thank Alexandros Tsengenes, whose research was the groundwork of this diploma thesis for his useful advice and guidance in automating the steps of Free Energy Perturbation calculations. Also, I would like to extend my thanks to Giorgio Amendola, PhD student, University of Campania "Luigi Vanvitelli", Italy, who helped me very much at the beginning of this thesis to understand some of the basic concepts in free energy simulations.

For their precious advice, I would also like to thank the two members of my MSc diploma thesis examination committee, Dr. Ionannis Emiris, Professor Level, National and Kapodistrian University of Athens, Department of Informatics and Telecommunications, and Dr. Stavros Perantonis, Research Director Institute of Informatics and Telecommunications, National Center of Scientific Research “Demokritos”.

I also would like to thank my fellow lab mates Alexios Chatzigoulas, Christina Athanasiou, and Joana Dulaj for all the valuable discussions and suggestions to my thesis presentation.

Last but not least, I would like to thank my family: my parents, my siblings and all my friends for supporting me spiritually throughout this thesis and in my life in general.

CONTENTS

LIST OF FIGURES

PREFACE

The master thesis ‘Automating free energy perturbation calculations for drug design’ has been conducted at the Biomedical Research Foundation Academy of Athens for the completion of the Postgraduate Program "Information Technologies in Medicine and Biology" (I.T.M.B.), Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece.

The first chapter presents, the lead optimization phase in the drug development process, the different computational methods that are used to perform lead optimization and the application of free energy perturbation (FEP) in this lead optimization phase of drug development. In the end the motivation of the study is presented.

In the second chapter, the theoretical foundations of the present work are presented. First, the Molecular Mechanics theory is explained. Afterwards the Molecular Dynamics (MD) simulations and the FEP theory are introduced. Next, the data structures of FEP/MD files are described. In the end there is an introduction on MD tools, such as VMD, and LigParGen, since they are mentioned multiple times in this project thesis.

The results of the present thesis are presented in chapter three. First, the implementation of the algorithm is explained. Then the program use is shown, explaining all the required inputs, and the various outputs of the tool. Afterwards some use cases are shown for a further understanding of the program use, along with the validation of the results showing their reliability.

Finally, the conclusions constitute the epilogue of this thesis, along with the possible future extensions of the tool in chapters four and five.

1. INTRODUCTION

The drug design process is undoubtedly a time-consuming and expensive endeavor, with recent estimates classifying it as a $2.6 billion expenditure [1]. The cost of preclinical and clinical stages of the drug design process accounts for several millions of US dollars. Fortunately, since the number of validated protein targets relevant to therapeutic applications has risen, efforts targeting the efficacious treatment of protein-provoking diseases have been more systematic [2]. In addition, in the last two decades the advances in high-throughput screening (HTS) experiments allowed the assessment of thousands of molecules concurrently by employing robotic automation [3].

Nonetheless, HTS is still time-consuming and expensive, with its acquisition value and operational costs being prohibitive for most laboratories. Moreover, in order to avoid costly failures, careful decision making together with the tremendous advances in computational technologies led to the advent of rational, computer-aided drug design (CADD). The conventional drug discovery processes has been revolutionized by Molecular modeling techniques, which enable the reduction of time and resources allocated in the hit identification, hit-to-lead optimization and lead optimization phases of the drug discovery pipeline. Novel drug-like candidates are first examined *in silico* for their expected affinity to a therapeutic target (in the case of structure-based drug design) or their similarity with previously identified active compounds (ligand-based drug design), as well as the prediction of physicochemical properties with the aid of sophisticated methods and algorithms. If desirable results have been received, the experimental part commences with molecular modeling prioritizing organic synthesis efforts [4]. The substantial cost that derives from failures can be eliminated by excluding drug candidates bearing no chance of demonstrating success early in the process.

The improvement in computer graphics and the development of algorithms made possible the simulation of biomolecular systems. These efforts were intensified due to the rapid development of GPU coding [5], the improvement of methodologies in both theoretical and application level [6, 7], as well as optimized algorithms enabling more accurate atomistic description and treatment of interactions that new force fields provide [8-10]. Moreover, problems related to poor sampling and difficulty in surpassing energetic barriers have been addressed with pioneering enhanced sampling techniques [11-13]. To sum up, nowadays more than ever the assistance of the methods has been recognized as a tool inextricably linked with drug-design-oriented attempts.

In recent years, significant advances in structure-based drug design and molecular modeling have contributed to the discovery of several drugs now in the market such as Sunitinib (kinase inhibitor, gastrointestinal cancer, Pfizer, 2006), Crizotinib (ALK inhibitor, NSCLC, GSK, 2011), and Nilotinib, which was rationally designed based upon the crystal structure of Imatinib/Bcr-Abl tyrosine kinase complexes.

CADD is usually used for three major purposes, filter large compound libraries into smaller sets of predicted active compounds that can be tested experimentally, guide the optimization of lead compounds and design novel compounds.

It is generally recognized that drug discovery and development are very time- and resource-consuming processes, which justifies the ever-growing effort to apply computational techniques to streamline drug discovery, design, development and optimization.

To start with, an initial lead compound is identified which allows the drug candidates to be designed; this process is called Lead Optimization. After new active compounds are identified, they enter the hit-to-lead and, subsequently, lead optimization phase during which the potency, solubility, oral bioavailability, metabolic stability and most importantly the binding affinity are improved. The above can be seen in Figure 1.

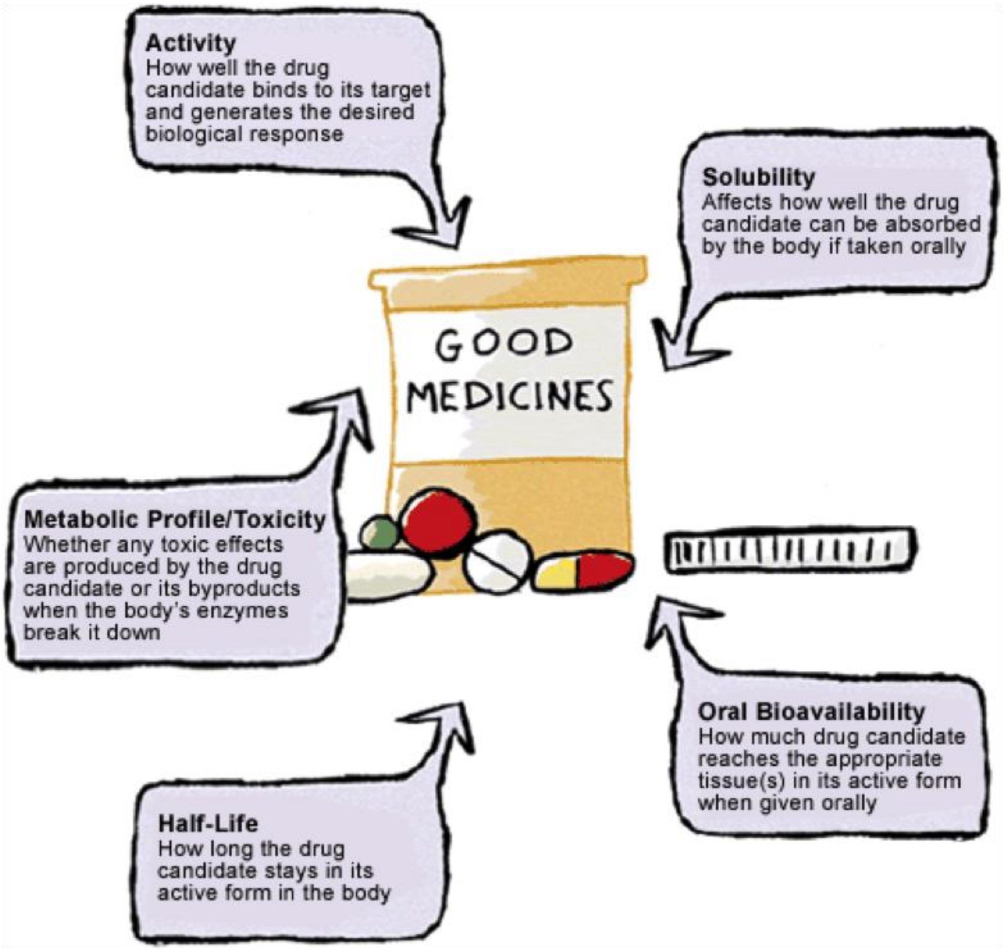

Activity
How well the drug candidate binds to its target and generates the desired biological response
Solubility
Affects how well the drug candidate can be absorbed by the body if taken orally
GOOD MEDICINES
Metabolic Profile/Toxicity
Whether any toxic effects are produced by the drug candidate or its byproducts when the body's enzymes break it down
Oral Bioavailability
How much drug candidate reaches the appropriate tissue(s) in its active form when given orally
Half-Life
How long the drug candidate stays in its active form in the body

Figure 1: Lead Optimization phase [18].

The lead discovery and lead optimization processes enhance various lead compounds and provide information to assist the selection process of the leads with the greatest potential to be developed into safe and effective medicines. One of the most important tasks in lead optimization is to predict among a series of lead candidates, which ones will bind more strongly to the therapeutic target. Relative binding free energy methodologies rely on physics-based molecular simulations and rigorous statistical mechanics to calculate the differences in the free energy of binding between a parent candidate drug and its analogues. In Figure 2 on the upper left hand a parent candidate drug is presented, the rest of the molecules are its analogues. Thousand of different analogues can occur with adding or removing small substituents.

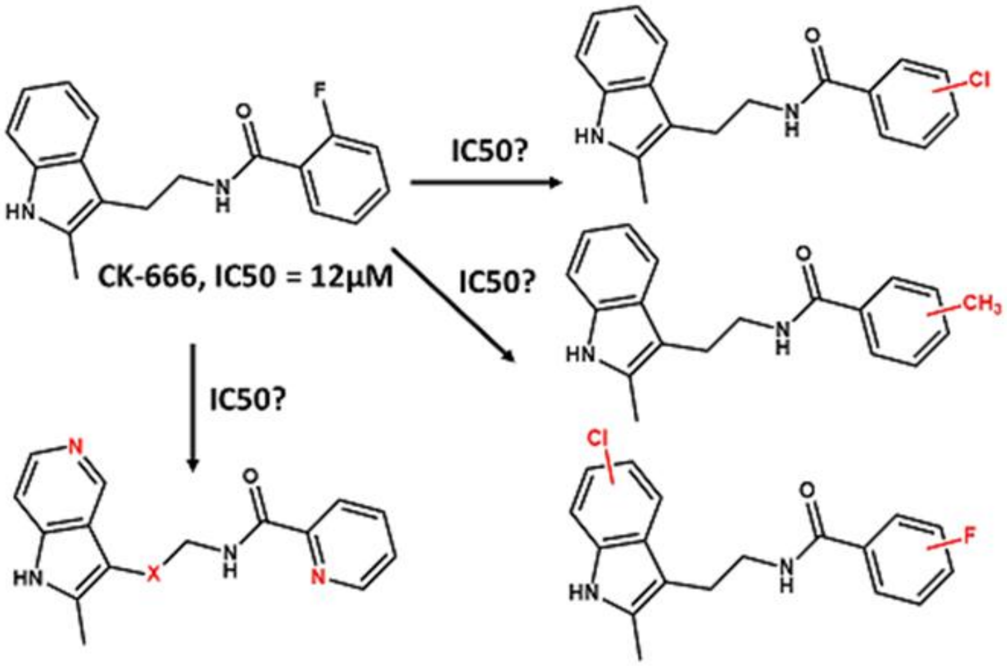


Figure 2: Lead optimization can occur with adding or removing small substituents to a lead compound.

1.1 Lead optimization in drug discovery

Lead optimization studies the effect of compounds on molecular structures [14]. It can occur with adding or removing small substituents to a lead compound, as small changes can lead to significant improvement (or decrease) in potency. As there are hundreds of combinations of potential changes (substitutions) on the molecule, it would be inefficient to ask the organic chemist to compose all these possible combinations in order to test them. Improved rational drug design methods are needed to lower the cost and increase the success rate of drug discovery and development [15]. Studies of analogue structure-activity relationships increase the chemist’s ability to predict chemical structural parameters for a given pharmacological action [16]. Once a target has been established and an assay for activity has been developed, chemists must develop compounds that interact with the target. Through the screening process, some compounds emerge with sufficient activity to guarantee further investigation. The active compounds are examined thoroughly against a number of criteria. Compounds that satisfy the selection criteria are called leads and are obtained for further optimization of activity, selectivity, and biological behaviour. The technique of discovering active compounds through screening and selecting the most promising ones as leads, is known as lead discovery [17].

Lead Discovery and Lead Optimization aim to maximize the interactions of a drug with its target binding site in order to improve activity, selectivity, and to minimize side effects. Designing a drug that can be synthesized efficiently and affordably is another priority. In Figure 3 the cost of the drug discovery process is shown. In the past decades, there has been a remarkable progress in the development of state-of-the-art drug discovery technologies, with particular emphasis on the processes of hit identification and lead generation and optimization. In line with the radical scientific advances in the area of medicinal chemistry, drug design approaches have become much more versatile and powerful. The integration of experimental and computational methods continues to play a vital role in drug design, creating wonderful opportunities and challenges in several stages of the drug discovery and development process [18].

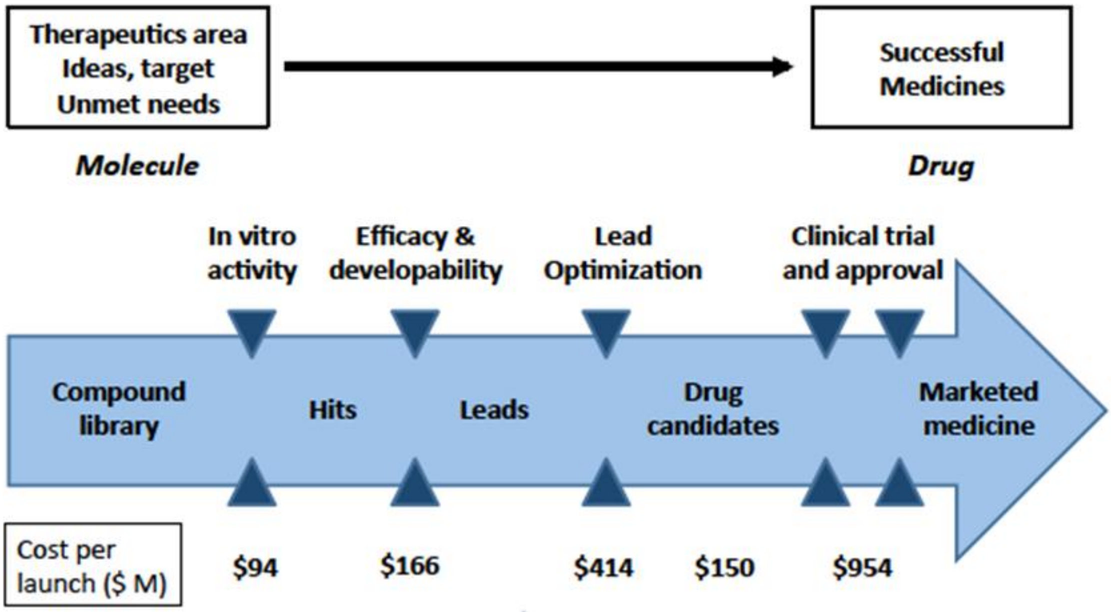


Figure 3: Drug Design Process [19].

1.2 Lead optimization with computational methods

The calculation of free energy by analytical methods is based on the laws of thermodynamics that govern the equilibria of molecular correlations. Over the last decade, the ability to implement models based on these laws has been gained, by developing better simulation algorithms and more powerful computing forces. These methods include Free Energy Perturbation (FEP), Thermodynamic Integration (TI), Molecular Mechanics Poisson Boltzman Surface Area (MM/PBSA) and Linear Interaction Energies (LIE).

FEP is one of the most accurate simulation techniques. Relative free binding energies between two ligands are calculated and compared in order to identify how strongly one molecule is bound to another. The basic criterion of the method is that the two ligands differ by a few individuals in terms of structure, which indicates that they are the same (Figure 4). It is important that the configurations taken by the first ligand, with a dynamic energy U_1, are largely identical to the configurations received by the other, with a dynamic energy U_2. This results in the dynamic actions of each molecule overlapping to such an extent that one condition (of the first ligand) to be considered as a "disorder" of the other [20]. The fact that the two molecules must be quite similar is also the reason why this method is very useful in optimizing driver molecules and is therefore used in this thesis.

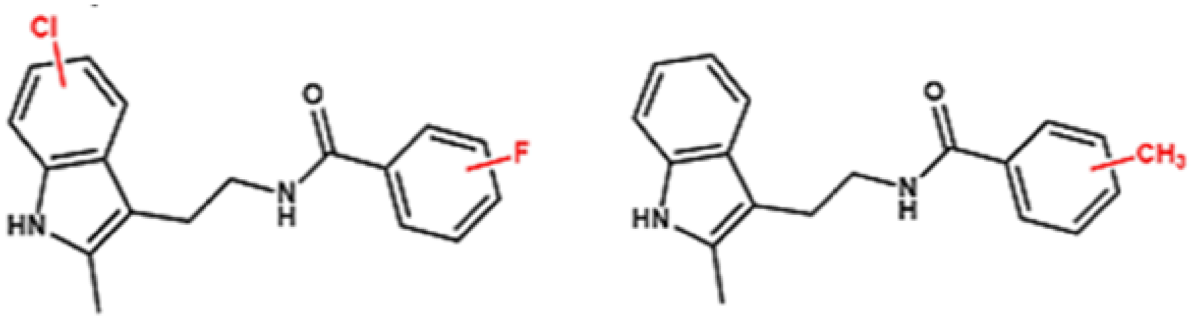


Figure 4: Two ligands that differ only by a few atoms.

The difference in free energy between two given states whose potential energies have different dependences on the spatial coordinates may be compared by using TI. The free energy of a system is not just a function of the phase space coordinates of the system but rather a function of the Boltzmann-weighted integral over phase space; as a result the free energy difference between two states cannot be calculated directly. During a thermodynamic integration, the free energy difference is calculated by defining a thermodynamic path between the states and integrating over ensemble-averaged enthalpy changes along the path. These paths can either be real chemical processes or alchemical processes [21].

In MM/PBSA the free energy of a molecule is calculated as the sum of its gas-phase energy, the solvation free energy, and a contribution due to the configurational entropy of the solute [22]. The molecular mechanics energy of the molecule calculates the gas-phase energy by determining the bonds, angles, and the torsion energies as well as Van der Waals and electrostatic interactions. Two contributions are considered to calculate the solvation free energy, a polar and a non-polar one [23, 24].

The polar contribution is calculated via a finite-difference solution of the Poisson–Boltzmann equation [25, 26]. Alternatively, an implicit solvent model based on Generalized Born (GB) theory can be used, which is a computationally more efficient approximation to Poisson theory. This then leads to the so-called MM-GBSA variant [27, 28].

The non-polar contribution is computed as the sum of an unfavourable energy resulting from cavity formation and a favourable energy stemming from attractive interactions between solute and solvent molecules [29, 30].

Finally, the configurational entropy of the solute is usually estimated using a rigid-rotor harmonic oscillator approximation, applying either normal mode analysis or quasi-harmonic analysis [31].

Expressions for LIE estimators for the binding of ligands to a protein receptor in implicit solvent are derived based on linear response theory and the cumulant expansion expression for the free energy. Using physical arguments, values of the LIE linear response proportionality coefficients are predicted for the explicit and implicit solvent electrostatic and Van der Waals terms. Motivated by the fact that the receptor and solution media may respond differently to the introduction of the ligand, a novel form of the LIE regression equation is proposed to model independently the processes of insertion of the ligand in the receptor and in solution [32].

1.3 Computational Tools with FEP

Numerous inputs need to be created to run the large numbers of simulations. However, creating the necessary input files for a simulation can be a laborious and time-consuming process. The key is to automate work–flows for simulation setup to the maximum extent reasonable. Not every step will be easily automated for various reasons such as limited development of present day algorithms and computing missing structural data from insufficient information. Nevertheless, the goal must still be to automate as many procedures as possible, but at the same time accept that they may not always be successful.

Today’s simulation packages still offer limited support for setup on relative free energy simulations. Therefore, alchemical simulation setup is an interesting target for automated simulations, especially considering its potential role in drug design and lead optimization. There have been reported several attempts at automating the setup of free energy calculations. For instance, Free Energy Workflow (FEW) [23] tool is available for AMBER [33, 34] for the setup of relative free energy simulations. PMX [35] is a program which automates the setup of relative free energy simulations of side–chain mutations for GROMACS [36]. GROMACS, also uses StaGE [37] for absolute hydration or binding free energy calculations. LOMAP [38] is a software project that reduces the graph of all possible ligand pairs to a minimum set based on a definition of similarity used to weight the graph's edges to solve the shortest path tree problem. Binding affinity calculator (BAC) [39] is an automation tool for rapid computation and analysis of ligand–receptor free energies.

The tools mentioned above may be effective, but they are stand alone codes that only implement one procedure each. Under these conditions in order to perform a GROMACS simulation one would have to use both PMX and StaGE. This can be confusing for an inexperienced user.

1.3.1 FESetup

FESetup is a new pipeline tool which aims to accelerate and facilitate the setup of alchemical free energy simulations for molecular simulation packages such as AMBER, GROMACS, or NAMD [40, 41]. Its advantages over the tools that were mentioned above are that FESetup is designed to support alchemical free energy simulations in a range of different MD and Monte Carlo (MC) packages, the setup process is independent of a given MD or MC code, and it is flexible enough to work within larger workflows e.g. using docking software to provide receptor–drug structures. It has been built to be open source, in order to provide a strong foundation to build setup workflows for different free energy methods. Last but not least, it is not only a free software and can be installed locally, but also interested parties are free to contribute at all levels, including code contributions and interfacing.

A protein from the Protein Data Bank (PDB) and a ligand are combined and solvated. Binding free energy simulations are carried out by the input files that are created. Many different algorithms and codes are used by FESetup in order to automate the setup of the alchemical free energy simulations. The most important ones are: AM1–BCC to automatically parameterize ligands [11]. Atom mappings for a single topology description are computed with a maximum common substructure search (MCSS) algorithm [42]. In this case fmcs (which is a connected MCSS algorithm) is used [43]. An abstract molecular dynamics (MD) engine can be used for equilibration prior to free energy setup or standalone. Currently, all modern AMBER force fields are supported.

Input is handled through a shell script, called FESetup, which sets up the environment and calls dGprep.py. This is the actual code that handles the user's input. FESetup will create all topology and template control files required for simulation. The input files do not, however, prescribe a specific λ schedule. It is not clear a priori what λ path would guarantee a smooth gradient (TI) or sufficient energetic overlap (FEP/BAR). This will depend on the nature of the system and is still an open question.

FESetup sets up the alchemical free energy simulations easier thus the tool can flexibly be integrated into larger workflows receiving a wide variety of structures. It creates simulation input for the MD packages AMBER, GROMACS, Sire, and, to some extent, NAMD. Although FESetup works perfectly for AMBER and GROMACS, it does not fully support NAMD. This is because AMBER and GROMACS implement a hybrid approach which means that they allow the assignment of a single and a dual–topology region at the same time whereas NAMD only allows dual topology approach.

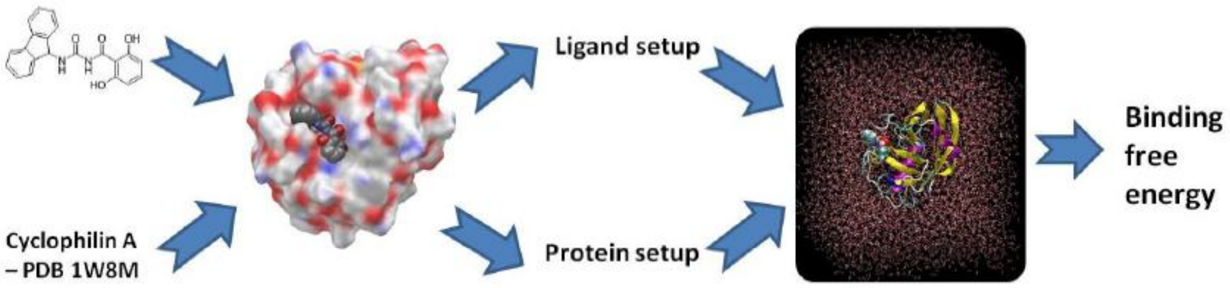


Figure 5: FESetup [40].

1.4 Study Objectives

The tools mentioned above implement algorithms for the set-up of different simulation packages, such as AMBER and GROMACS, but they do not automate the procedure for simulation packages that implement the dual topology approach, such as NAMD. Although FESetup is reported to be the most advanced and completed of all the tools mentioned, it has not yet implement an algorithm that automates the set-up for dual topology files either. The lack of such a tool, created the vital need for the implementation of a web-based server that fully automates the set-up procedure for simulation packages that implement the dual topology approach.

Commercial simulation packages cost thousands of Euros making it difficult for some labs to acquire them; on the other hand there are free simulation packages that are reported to be as efficient as the commercial ones. In this direction it had to be investigated whether free simulation packages that perform FEP, specifically, are able to compete against them and therefore predict accurately how strongly a molecule will bind to a protein. The above investigation led us to the conclusion that free simulation packages can actually predict correctly the binding affinity between a parent candidate drug and its analogues. The only setback is that packages that use the dual topology approach are difficult to set-up. Consequently we decided to automate the set-up procedure and create a FEP set-up automation engine.

The most challenging part of this project was not just the fact that at the time no such tool existed, but also there was no straight forward procedure published. Consequently the purpose was not only to code and automate a given workflow, but also to devise it.

2. METHODS

2.1 Molecular Mechanics

Molecular Modeling can be considered as the set of theoretical, computational and experimental methods used to model or mimic the behavior of the molecules. The common element of all methods is the atomic-level description of molecular systems. This means that they treat individuals as the smallest autonomous group (Molecular Mechanics) or model each atom's electrons (Quantum Mechanics) [44, 45].

The size of biological systems and the order of magnitude of time, in which interesting biological phenomena arise, are prohibitive for their study with the methods of Quantum Mechanics. Since it is not computationally possible to handle such systems with Quantum Mechanics, approximate methods have been developed that obey the principles of Molecular Engineering. The following are briefly presented with the approaches that make the transition from the Schrödinger equation to the laws governing Molecular Engineering.

Matter consists of atoms, which are composed of electrons and a nucleus. A molecule, consisting of several cores, can be considered as a molecular system. Classical Mechanics describes a system with the coordinates and velocity of each particle of the system. However, as the particles of the molecular systems are very small, they obey the laws of Quantum Mechanics, and so they can be described by a wave function of the positions of all the particles [46, 47]. This wave function must satisfy Schrödinger's time independent equation [48].

Although the Schrödinger's time-independent equation could predict most of the properties of a molecule, calculations based on it are very rare. The main reason for this is that the Schrödinger equation is very difficult to solve. One way to simplify the Schrödinger equation for molecular systems is the Born-Oppenheimer approach [49].

According to this, the movement of the nuclei is much smaller than that of the electrons, so the electrons adapt to negligible time in the nucleus shifts. This is why this approach divides the Schrödinger equation into two pieces: the "electron" and the "nuclear" piece. For each kernel shift, the solution of the Schrödinger electronic equation gives the electron wave function or else its Potential Energy Surface (PES) in which the cores "move" [50].

Due to the large size of biomolecules, the solution to Schrödinger's nuclear equation for kernel movement is still prohibitive. For this reason, a further assumption is made, according to which the dynamics of the nuclei are described by the laws of Classical Mechanics and obey Newton's equations of motion. This assumption introduces the concept of Molecular Engineering, the principle of which is that a molecular system can be considered as a tiny mechanical system. Interactions between atoms are determined by the function of dynamic energy, which calculates all the relative forces for Newton's equation of motion and which describes the microscopic dynamics of the molecular system.

Although the Born-Oppenheimer approach allows the response of the electron wave function as a function of the nuclear coordinates, the solution of the Schrödinger electronic equation is still required to calculate the dynamic energy. This is, computationally, extremely costly for the large number of electrons in biological systems. For this reason, the expression of the dynamic energy of the system is assumed to be the sum of simple analytical functions. These functions, combined with a set of empirical parameters, compose the field of forces of molecular engineering force field. In other words, a force field describes the dynamic energy as the sum of the actions of the various interactions between atoms [51]. Interactions are divided into two types: intramolecular and intermolecular. Therefore, the dynamic energy of the system consists of two "components": that of intramolecular interactions and intermolecular interactions.

The local part of the potential energy is described as follows:

$$U^{local}_{potensial}(R) = \sum_{bonds} \frac{1}{2} k_b \left(l_i - l_i^o\right)^2 \\ \sum_{angles} \frac{1}{2} k_\theta \left(\theta_i - \theta_i^o\right)^2 + \\ \sum_{dihedrals} \frac{1}{2} V_n \left[1 + cos\left(n\,\phi_i - \gamma\right)\right] \\ \sum_{impropers} \frac{1}{2} k_\omega \left(\omega_i - \omega_i^o\right)^2 \qquad (2.1)$$

The first term describes the energy of covalent bonds with the help of the Hooke Law (as spring energy), while the second describes the bend angle formed between three successive covalently bonded atoms. The third term simulates the dihedral angles, formed between four consecutive atoms of different levels, with an oscillation function, while the fourth term describes the off-level movements of the atoms.

The non-local part of the potential energy consists of the electrostatic forces and Van der Waals forces:

$$U^{non-local}_{potensial} = U_{electrost} + U_{van-der-waals} \qquad (2.2)$$

Where:

$$U_{electrost} = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \frac{q_i q_j}{4\pi\varepsilon_0 \left|r_i - r_j\right|}$$

And:

$$U_{vdw} = U_{LJ} = 4\varepsilon \left[\left(\frac{\sigma}{R}\right)^{12} - \left(\frac{\sigma}{R}\right)^{6}\right]$$

As can be seen from the above equations, the electrostatic forces are described by the Law of Coulomb, while the Van der Waals forces by the Lennard-Jones potential [44].

2.2 Molecular Dynamics Simulations

Molecular Dynamics (MD) is a computer simulation technique where classical engineering motion equations are used to describe the movement of atoms and molecules in order to observe the evolution of a system over time. In general, in order to study the structure and the dynamics of biomolecules the next three steps are implemented:

1. The total energy is the summation of the potential energy and the kinetic energy. In order to calculate the total energy, the potential energy has to be modeled and coordinates from experimental structures need to be used. Moreover initial velocities need to be assigned.
2. Integration of Newton's second law needs to be applied and the new velocities (**v**) of the system and the new coordinates (**r**) of the atoms need to be acquired.
3. Macroscopic properties can be expressed through **v** and **r** via statistical mechanics.

The Molecular Dynamics Method uses Newton's motion equations. For an N atomic molecular system, the force that each particle i will receive from all atoms at each time point is given by the relationship:

$$F_i(r,t) = -\nabla U_{total}(r,t) = m_i \frac{\partial^2 r_i}{\partial t^2} \tag{2.3}$$

U_{total} is the potential energy as defined in the previous section, which is calculated with the help of the force field and which determines the interaction energy of each with the other elements of the system. r_i is the position of each individual at any time.

As soon as all forces are calculated, Equation (2.3) is numerically completed using a suitable algorithm to generate new positions and new speeds for each particle of the system. Next, the new coordinates are used to calculate the potential energy. The steps in the equilibrium Molecular Dynamics simulation of the system are as follows:

1. Selection of the original conditions (coordinates) of the atoms of the molecular system.
2. Select their initial speeds.
3. Calculate the momentum for each atom by its velocity and mass.
4. Calculation of the forces exerted as a result of its interactions.
5. After a dt time, which is also the simulation step, atom's' new position in space is calculated.
6. Calculate the new speeds and accelerations.
7. Repeat steps 3-6 until the system has reached equilibrium.

2.3 FEP Theory

The important task is to know how strongly a molecule will attach to a receptor. This depends on the interactions formed between them. Generally, the association of two interacting molecules depends on enthalpy and entropy factors, meaning that the ligation process involves changes in the structure and dynamics of both molecules involved. Like any spontaneous process, the compounds of two molecules results only when it is characterized by negative free Gibbs binding energy. This is calculated from the following relationship:

$$\Delta G_b = \Delta H_b – T\ \Delta S_b \tag{2.4}$$

ΔG_b, ΔH_b, ΔS_b are the free energy changes of Gibbs, enthalpy, and entropy, respectively, upon binding.

According to the above relationship, attachment can be favored by enthalpy, entropy or both. Development of interactions such as hydrogen bonds, Van der Waals et al. is accompanied by a heat dissipation resulting in a reduction in enthalpy. On the contrary, the hydrophobicity effect leads to an entropic advantage.

Predicting the free energy binding of a ligand to a protein is one of the most important and demanding goals of rational drug design [52, 53]. Software that studies binding has two goals: finding the ligand configuration in the receptor cavity, and calculating the free binding energy. For the latter, a number of functions have been developed.

The method is shown in Figure 6. Accordingly, it compares the free binding action of ligand A with that of ligand B. This is done by calculating the difference in ΔG_A and ΔG_B values:

$$\Delta\Delta G = \Delta G_B - \Delta G_B \tag{2.5}$$

Negative ΔΔG value implies that the free binding energy of B (ΔG_B) is less than the free binding energy of A (ΔG_A), so the binding of B is the favored energy with respect to the binding of A. Therefore, if ligand A is the driver molecule, in order to optimize it, it is necessary to identify a ligand B so that the difference in its free binding actions is negative.

However, identifying the ΔG_A and ΔG_B computationally is not an easy task because of the large energy barriers that emerge in simulating the specific changes. In order to overcome this difficulty, a technique is used which indirectly calculates the value of ΔΔG, in which ligand A is alchemically altered or "mutated" in ligand B. The change of free energy in the conversion of A to B is therefore studied. This calculation is performed when the two molecules are in the solvent-water and when they are bound to the receptor.

According to Figure 6, therefore, a closed thermodynamic cycle is created. The sum of all the changes along the cycle equals zero because Gibbs free energy is a thermodynamic property.

$$\Delta G_A + \Delta G_2 - \Delta G_B - \Delta G_1 = 0 \Rightarrow \Delta G_A - \Delta G_B = \Delta G_1 - \Delta G_2 \tag{2.6}$$

So the difference is calculated indirectly by the equation:

$$\Delta\Delta G = \Delta G_2 - \Delta G_1 \tag{2.7}$$

method developed by Robert Zwanzig in 1954 and the change in free energy is calculated from the equation [54]:

$$\Delta G(A \rightarrow B) = G_B - G_A = -kTln\,\langle \exp\left(-\frac{U_B - U_A}{kT}\right)\rangle_A \tag{2.8}$$

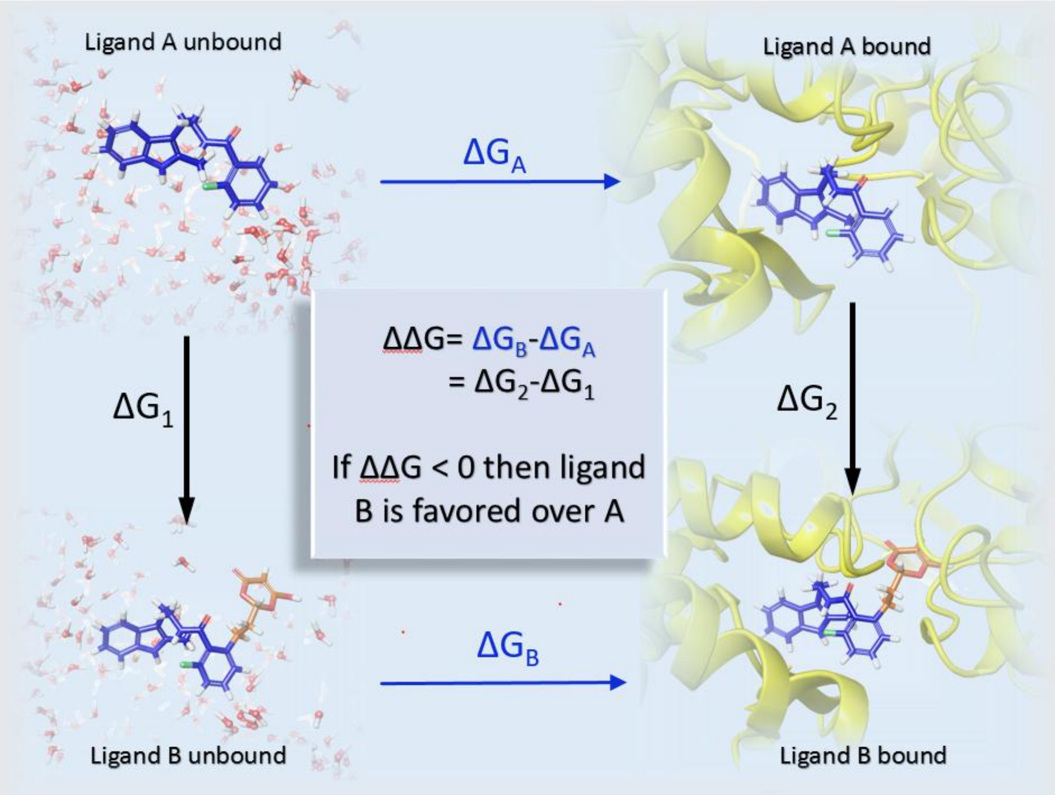


Figure 6: Free Energy Perturbation (FEP). The free binding energy of A and B is ΔG_A and ΔG_B respectively, whereas the free energy of conversion of A to B in the solvent and protein is ΔG_1 and ΔG_2, respectively.

In FEP/MD simulations, one ligand is gradually converted to the other. This gradual change is defined by a coupling parameter λ which is valued from 0 to 1, 0 for the original ligand and 1 for the final. This way, the simulation is divided into "windows". Each "window" corresponds to a specific value of λ. In the "windows" between the original and the final states, the molecules being simulated are intermediate, non-real molecules with intermediate parameters. In Figure 7 the technique of λ windows is shown. The recruitment of non-real intermediate molecules is the cause this particular method belongs to alchemical conversion methods. The reason for not directly converting one ligand to the other is the need for a smooth transition from one condition (one ligand) to the other (the other ligand). The FEP method requires that the two binders receive the same or similar configurations during the simulation so that the dynamic energy of one situation is only a "disruption" of the other's dynamic energy. Different binders are expected to adopt different configurations. Thus, this technique, through the intermediary "windows", is used to bridge the formative gap of the two different situations.

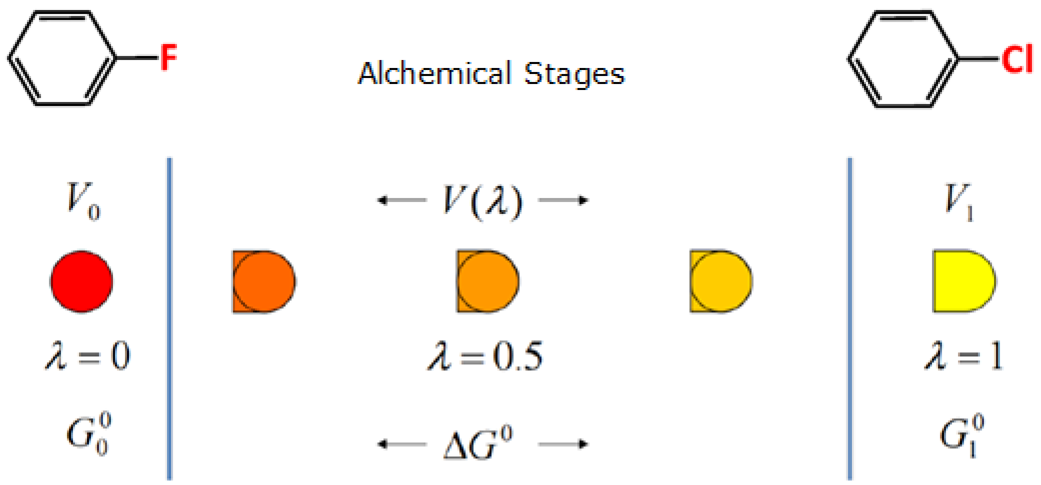


Figure 7 : λ windows technique [55].

2.4 Structure of FEP/MD files

The data structures include information about the molecule's chemical composition and connectivity as well as certain atomic properties and internal coordinates for the energy function. The topology and the parameter files contain this information for a particular class of molecule.

The Residue topology file (top.chm) contains the information of the residues which are used to build large molecules. It contains the atom type, mass, hydrogen bond donors and acceptors as well as atoms' partial charges in particular residue for each and every atom in the system.

The Parameter file (par.chm) is associated with the RTF file as it contains all the necessary parameters for calculating the energy of the molecule. These include equilibrium bond distances, angles for bond stretching, angle bending and dihedral angle terms in the potential energy function as well as the force constants and the Lennard Jones parameters.

The Protein structure file (PSF) is the most fundamental data structure for NAMD. It is generated for a specific molecule and it is the concatenation of the information contained in the RTF file. It provides detailed information on the composition and the connectivity of the atoms in the molecule of interest. It gives the total number of bonds and provides information as to which atoms connect to form a particular bond. The PSF file must be specified before any calculation is performed. It contains the molecular topology, but not any information regarding bond lengths, angles etc. This information is provided by the above topology and parameter files.

The Coordinate file (PDB) contains the Cartesian coordinates of all atoms in the system. These are mostly obtained by the X-ray or the NMR experiments. Missing coordinates can be built within CHARMM [56] using the internal coordinate facility. In addition, the hydrogens which are not present in the X-ray crystal structure can also be built in by CHARMM by using the module HBUILD.

2.5 Single-Dual Topologies

For relative free energy simulations we can choose between the single–topology and the dual–topology approach [57, 58]. Codes like NAMD only allow the latter. Codes like AMBER or GROMACS implement hybrid approaches since they allow the assignment of a single and a dual–topology region at the same time.

In the dual-topology approach, both reference and target state atoms exist at the same time, reference state atoms disappear and target state atoms appear. The final end states describe a “non–existing” molecule [59]. This can be achieved if the reference ligand is merged with the mutant into one file. In single-topology dummy atoms change to the atoms that have to appear. The single–topology region keeps the ligands in place as the coordinates are shared and only direct conversion of one atom to another is allowed to occur. The atoms within this region are thus always present. In Figure 8 both techniques are represented.

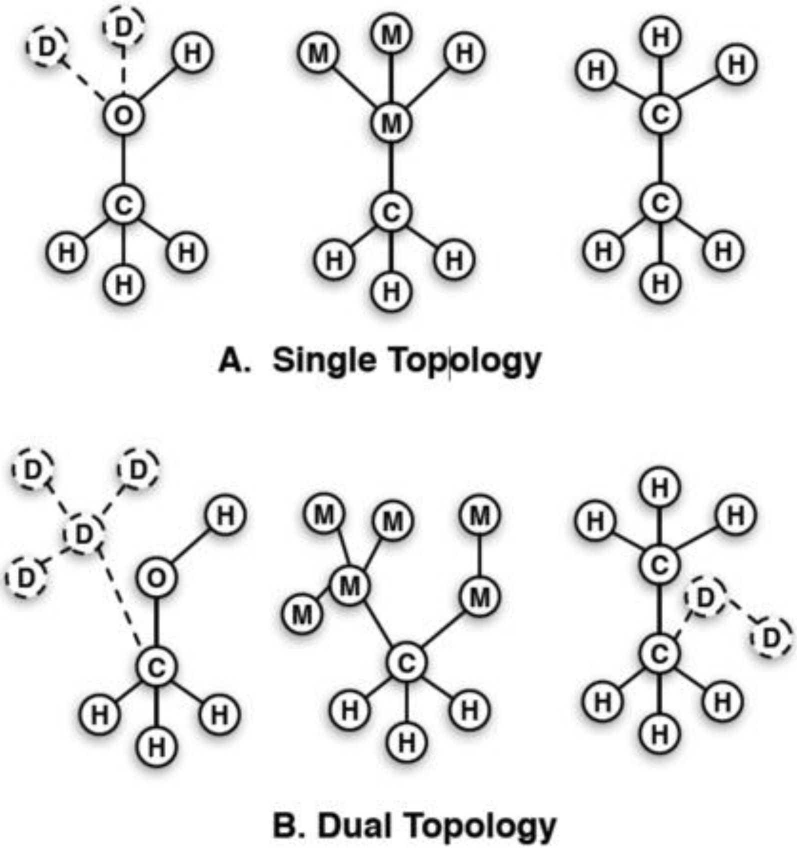


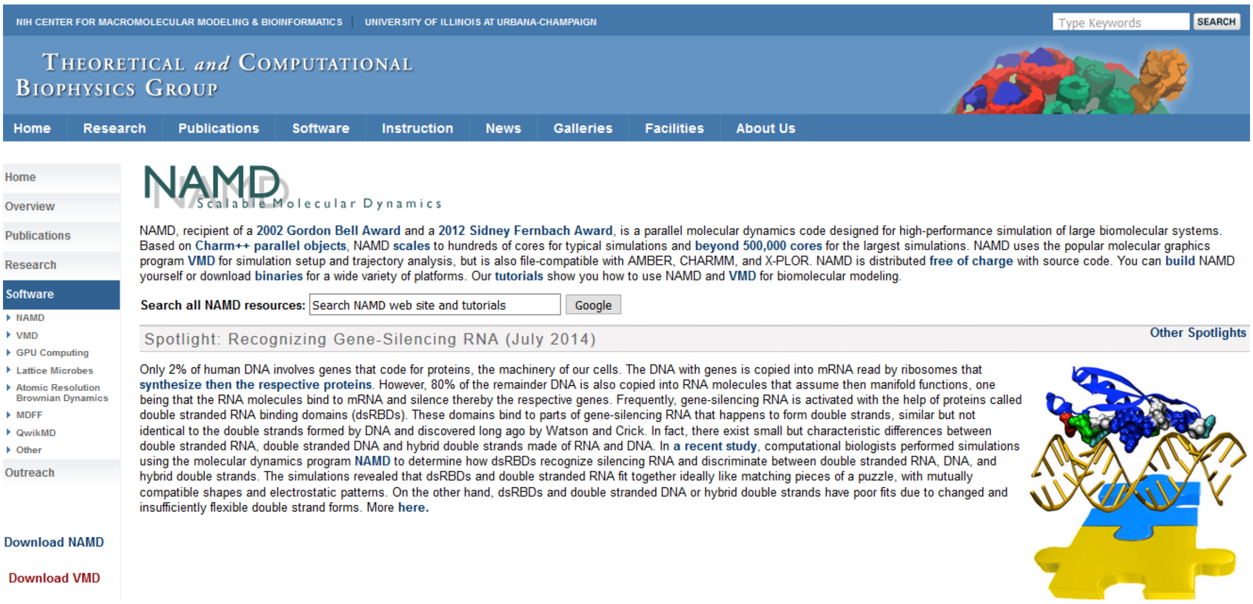
Figure 8: Single-Dual Topology approach [60].

2.6 NAMD

NAMD (Figure 9) is a parallel, MD simulation program, used to model biomolecular systems using high performance computing (HPC) clusters. When NAMD is run, patches are distributed evenly. Then, a larger number of compute objects responsible for calculating atomic interactions either within a single patch or between neighbouring patches is distributed across the processors, minimizing communication by grouping compute objects responsible for the same patch together on the same processors [61]. With NAMD we want to simplify access to dynamic information calculated from structures and provide a molecular modeling tool that can be used productively by a wide group of biomedical researchers. The purpose of NAMD is to enable high performance classical simulation of biomolecules in realistic environments of 100,000 atoms or more. A decade ago in its first release [62, 63], NAMD permitted simulation of a protein-DNA complex encompassing 36,000 atoms [64], the most recent release permitted the simulation of a protein-DNA complex of 314,000 atoms [65]. NAMD uses a stochastic coupling approach because it is easier to implement and the friction terms tend to enhance the dynamical stability. The (stochastic) Langevin equation [66] is used in NAMD to generate the Boltzmann distribution for canonical ensemble simulations.

One can download NAMD from:

http://www.ks.uiuc.edu/Research/namd/



NAMD, recipient of a 2002 Gordon Bell Award and a 2012 Sidney Fernbach Award, is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. Based on Charm++ parallel objects, NAMD scales to hundreds of cores for typical simulations and beyond 500,000 cores for the largest simulations. NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER, CHARMM, and X-PLOR. NAMD is distributed free of charge with source code. You can build NAMD yourself or download binaries for a wide variety of platforms. Our tutorials show you how to use NAMD and VMD for biomolecular modeling.

Search all NAMD resources: Search NAMD web site and tutorials

Spotlight: Recognizing Gene-Silencing RNA (July 2014)

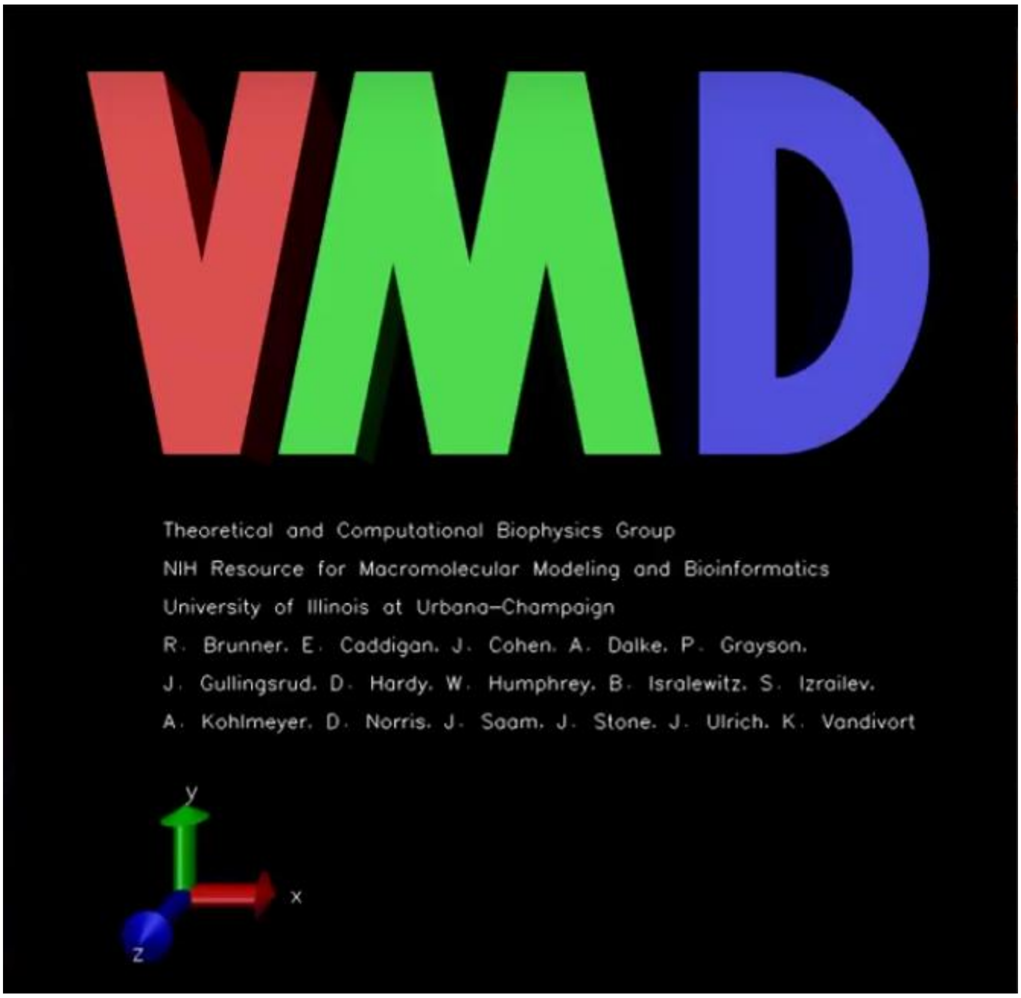
Only 2% of human DNA involves genes that code for proteins, the machinery of our cells. The DNA with genes is copied into mRNA read by ribosomes that synthesize then the respective proteins. However, 80% of the remainder DNA is also copied into RNA molecules that assume then manifold functions, one being that the RNA molecules bind to mRNA and silence thereby the respective genes. Frequently, gene-silencing RNA is activated with the help of proteins called double stranded RNA binding domains (dsRBDs). These domains bind to parts of gene-silencing RNA that happens to form double strands, similar but not identical to the double strands formed by DNA and discovered long ago by Watson and Crick. In fact, there exist small but characteristic differences between double stranded RNA, double stranded DNA and hybrid double strands made of RNA and DNA. In a recent study, computational biologists performed simulations using the molecular dynamics program NAMD to determine how dsRBDs recognize silencing RNA and discriminate between double stranded RNA, DNA, and hybrid double strands. The simulations revealed that dsRBDs and double stranded RNA fit together ideally like matching pieces of a puzzle, with mutually compatible shapes and electrostatic patterns. On the other hand, dsRBDs and double stranded DNA or hybrid double strands have poor fits due to changed and insufficiently flexible double strand forms. More here.

Figure 9: NAMD [49].

2.6.1 Visual Molecular Dynamics

Visual Molecular Dynamics (VMD) (Figure 10) is a molecular viewer program that can be used to help set up NAMD simulation and help analyze and visualize NAMD output. VMD displays structures using a wide variety of methods. It provides a complete graphical user interface, as well as a text interface using the Tool Command Language (Tcl) embeddable parser to allow for complex scripts run. By generating input scripts it produces high-resolution images of displayed molecules. VMD also animates molecular dynamics (MD) simulation trajectories, imported either from files or from a direct connection to a running MD simulation [67]. VMD can be freely obtained from:

http://www.ks.uiuc.edu/Research/vmd/



Theoretical and Computational Biophysics Group
NIH Resource for Macromolecular Modeling and Bioinformatics
University of Illinois at Urbana–Champaign
R. Brunner, E. Caddigan, J. Cohen, A. Dalke, P. Grayson,
J. Gullingsrud, D. Hardy, W. Humphrey, B. Isralewitz, S. Izrailev,
A. Kohlmeyer, D. Norris, J. Saam, J. Stone, J. Ulrich, K. Vandivort

Figure 10: VMD [55].

2.6.2 LigParGen

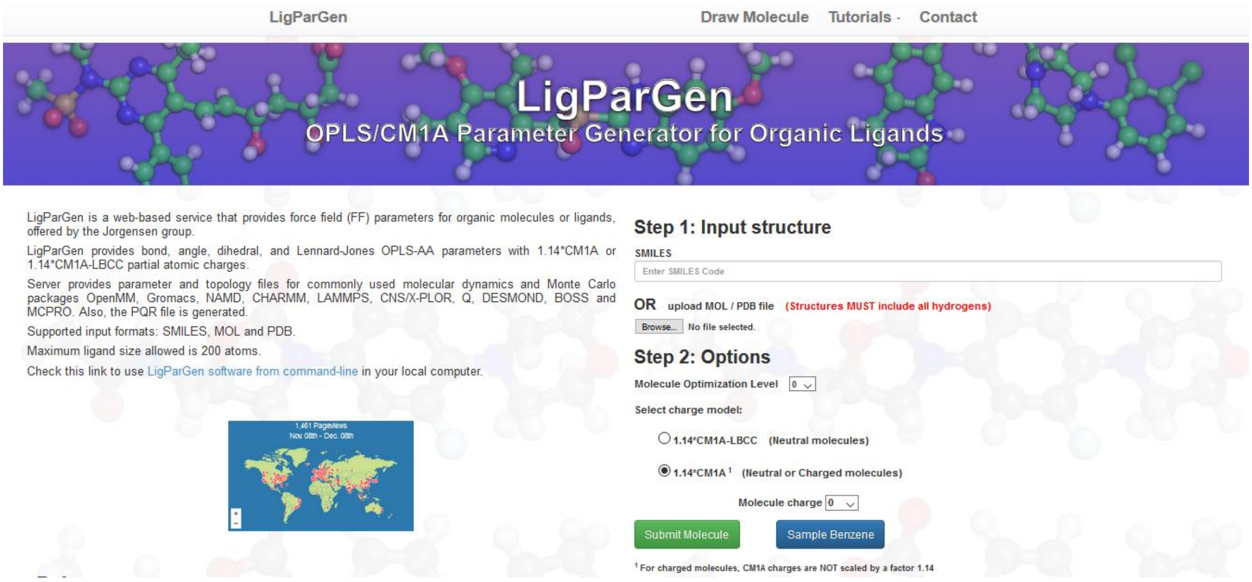
The Jorgensen group has developed a web-server that provides force field (FF) parameters for organic molecules or ligands (Figure 11).

In molecular mechanics and molecular dynamics simulations Force Field parameters are referred as the functional form and parameter sets, used to calculate the potential energy of a system of atoms or coarse-grained particles. The parameters of the energy functions may be derived from experiments in physics or chemistry, calculations in quantum mechanics, or both. A force field parameter file contains all of the numerical constants needed to evaluate forces and energies, given a structure file and atomic coordinates [44, 68-70].

The accurate calculation of ligand interactions by force field-based methods requires a precise description of the energetics of intermolecular interactions. Despite the progress made in force fields, small molecule parameterization remains an open problem due to the magnitude of the chemical space; the most critical issue is the estimation of a balanced set of atomic charges with the ability to reproduce experimental properties. The LigParGen web server provides an interface for generating Optimized Potentials for Liquid Simulations, OPLS-AA/1.14*CM1A(-LBCC) force field parameters for organic ligands, in the formats of commonly used molecular dynamics and Monte Carlo simulation packages [71].

LigParGen can be accessed through:

http://zarbi.chem.yale.edu/ligpargen/



LigParGen is a web-based service that provides force field (FF) parameters for organic molecules or ligands, offered by the Jorgensen group.

LigParGen provides bond, angle, dihedral, and Lennard-Jones OPLS-AA parameters with 1.14*CM1A or 1.14*CM1A-LBCC partial atomic charges.

Server provides parameter and topology files for commonly used molecular dynamics and Monte Carlo packages OpenMM, Gromacs, NAMD, CHARMM, LAMMPS, CNS/X-PLOR, Q, DESMOND, BOSS and MCPRO. Also, the PQR file is generated.

Supported input formats: SMILES, MOL and PDB.

Maximum ligand size allowed is 200 atoms.

Check this link to use LigParGen software from command-line in your local computer.

Step 1: Input structure

SMILES: Enter SMILES Code

OR upload MOL / PDB file (Structures MUST include all hydrogens)

No file selected.

Step 2: Options

Molecule Optimization Level: 0

Select charge model:

○ 1.14*CM1A-LBCC (Neutral molecules)

● 1.14*CM1A [1] (Neutral or Charged molecules)

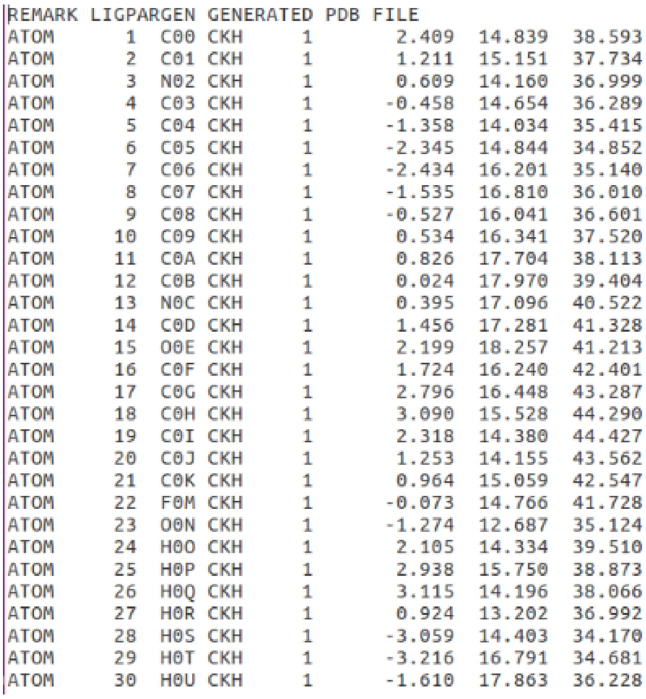
Molecule charge: 0

[1] For charged molecules, CM1A charges are NOT scaled by a factor 1.14

Figure 11: LigParGen [71].

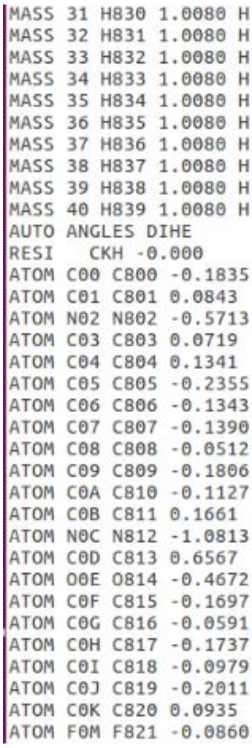
2.6.3 Data Structures

As explained in chapter 2.2.2 small molecules have no standard parameters, such as partial charges or force field parameters, so these need to be generated. This can be achieved through LigParGen. For our calculations the acquisition of the PDB (coordinates-Figure 12), RTF (topology-Figure 13) and PRM (parameters-Figure 14) files is needed. Afterwards it is explained how exactly this can be done.



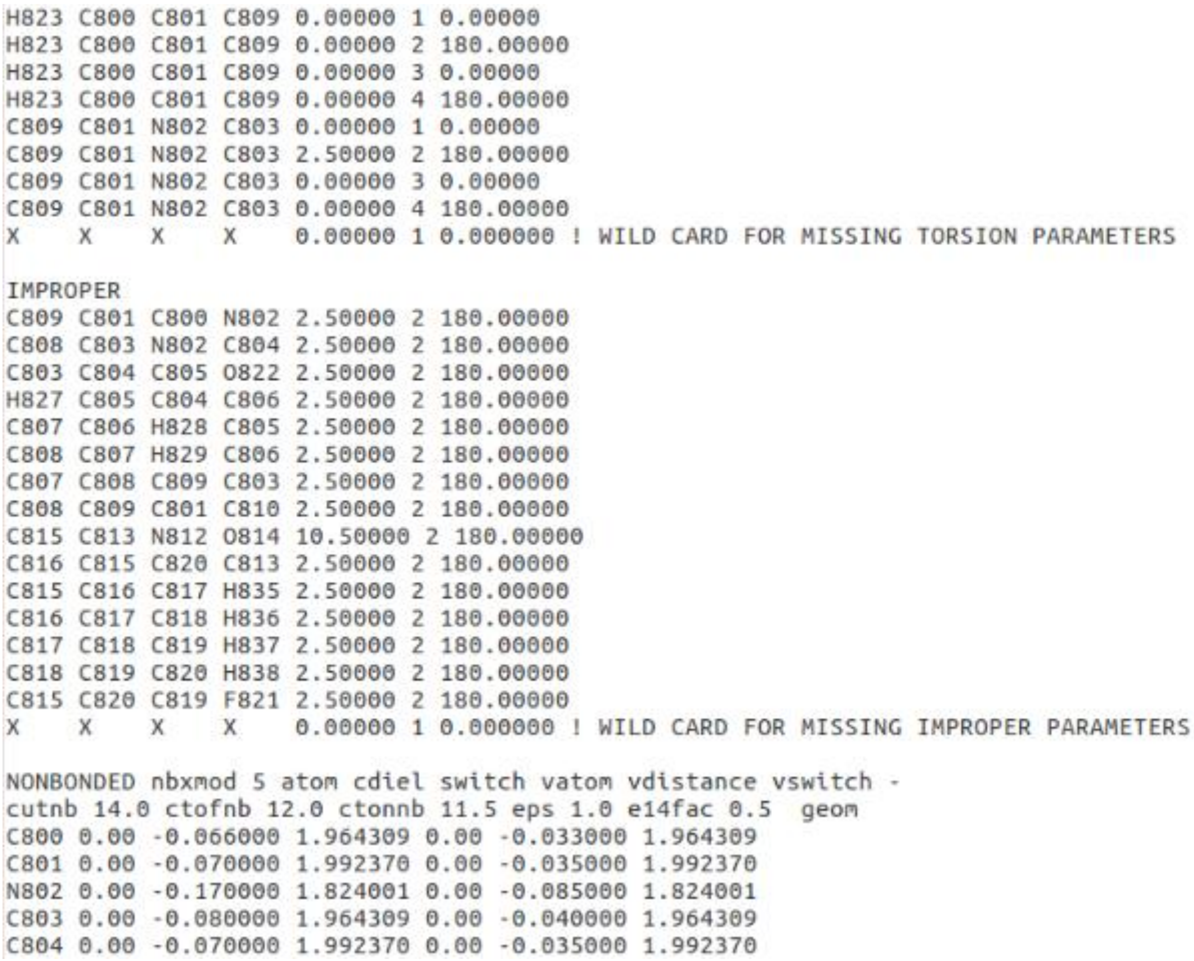
```
REMARK LIGPARGEN GENERATED PDB FILE
ATOM      1  C00 CKH     1       2.409  14.839  38.593
ATOM      2  C01 CKH     1       1.211  15.151  37.734
ATOM      3  N02 CKH     1       0.609  14.160  36.999
ATOM      4  C03 CKH     1      -0.458  14.654  36.289
ATOM      5  C04 CKH     1      -1.358  14.034  35.415
ATOM      6  C05 CKH     1      -2.345  14.844  34.852
ATOM      7  C06 CKH     1      -2.434  16.201  35.140
ATOM      8  C07 CKH     1      -1.535  16.810  36.010
ATOM      9  C08 CKH     1      -0.527  16.041  36.601
ATOM     10  C09 CKH     1       0.534  16.341  37.520
ATOM     11  C0A CKH     1       0.826  17.704  38.113
ATOM     12  C0B CKH     1       0.024  17.970  39.404
ATOM     13  N0C CKH     1       0.395  17.096  40.522
ATOM     14  C0D CKH     1       1.456  17.281  41.328
ATOM     15  O0E CKH     1       2.199  18.257  41.213
ATOM     16  C0F CKH     1       1.724  16.240  42.401
ATOM     17  C0G CKH     1       2.796  16.448  43.287
ATOM     18  C0H CKH     1       3.090  15.528  44.290
ATOM     19  C0I CKH     1       2.318  14.380  44.427
ATOM     20  C0J CKH     1       1.253  14.155  43.562
ATOM     21  C0K CKH     1       0.964  15.059  42.547
ATOM     22  F0M CKH     1      -0.073  14.766  41.728
ATOM     23  O0N CKH     1      -1.274  12.687  35.124
ATOM     24  H0O CKH     1       2.105  14.334  39.510
ATOM     25  H0P CKH     1       2.938  15.750  38.873
ATOM     26  H0Q CKH     1       3.115  14.196  38.066
ATOM     27  H0R CKH     1       0.924  13.202  36.992
ATOM     28  H0S CKH     1      -3.059  14.403  34.170
ATOM     29  H0T CKH     1      -3.216  16.791  34.681
ATOM     30  H0U CKH     1      -1.610  17.863  36.228
```

Figure 12: Structure of a PDB file.



```
MASS 31 H830 1.0080 H
MASS 32 H831 1.0080 H
MASS 33 H832 1.0080 H
MASS 34 H833 1.0080 H
MASS 35 H834 1.0080 H
MASS 36 H835 1.0080 H
MASS 37 H836 1.0080 H
MASS 38 H837 1.0080 H
MASS 39 H838 1.0080 H
MASS 40 H839 1.0080 H
AUTO ANGLES DIHE
RESI   CKH -0.000
ATOM C00 C800 -0.1835
ATOM C01 C801 0.0843
ATOM N02 N802 -0.5713
ATOM C03 C803 0.0719
ATOM C04 C804 0.1341
ATOM C05 C805 -0.2355
ATOM C06 C806 -0.1343
ATOM C07 C807 -0.1390
ATOM C08 C808 -0.0512
ATOM C09 C809 -0.1806
ATOM C0A C810 -0.1127
ATOM C0B C811 0.1661
ATOM N0C N812 -1.0813
ATOM C0D C813 0.6567
ATOM O0E O814 -0.4672
ATOM C0F C815 -0.1697
ATOM C0G C816 -0.0591
ATOM C0H C817 -0.1737
ATOM C0I C818 -0.0979
ATOM C0J C819 -0.2011
ATOM C0K C820 0.0935
ATOM F0M F821 -0.0860
```

Figure 13: Structure of an RTF



```
H823 C800 C801 C809 0.00000 1 0.00000
H823 C800 C801 C809 0.00000 2 180.00000
H823 C800 C801 C809 0.00000 3 0.00000
H823 C800 C801 C809 0.00000 4 180.00000
C809 C801 N802 C803 0.00000 1 0.00000
C809 C801 N802 C803 2.50000 2 180.00000
C809 C801 N802 C803 0.00000 3 0.00000
C809 C801 N802 C803 0.00000 4 180.00000
X    X    X    X    0.00000 1 0.000000 ! WILD CARD FOR MISSING TORSION PARAMETERS

IMPROPER
C809 C801 C800 N802 2.50000 2 180.00000
C808 C803 N802 C804 2.50000 2 180.00000
C803 C804 C805 O822 2.50000 2 180.00000
H827 C805 C804 C806 2.50000 2 180.00000
C807 C806 H828 C805 2.50000 2 180.00000
C808 C807 H829 C806 2.50000 2 180.00000
C807 C808 C809 C803 2.50000 2 180.00000
C808 C809 C801 C810 2.50000 2 180.00000
C815 C813 N812 O814 10.50000 2 180.00000
C816 C815 C820 C813 2.50000 2 180.00000
C815 C816 C817 H835 2.50000 2 180.00000
C816 C817 C818 H836 2.50000 2 180.00000
C817 C818 C819 H837 2.50000 2 180.00000
C818 C819 C820 H838 2.50000 2 180.00000
C815 C820 C819 F821 2.50000 2 180.00000
X    X    X    X    0.00000 1 0.000000 ! WILD CARD FOR MISSING IMPROPER PARAMETERS

NONBONDED nbxmod 5 atom cdiel switch vatom vdistance vswitch -
cutnb 14.0 ctofnb 12.0 ctonnb 11.5 eps 1.0 e14fac 0.5  geom
C800 0.00 -0.066000 1.964309 0.00 -0.033000 1.964309
C801 0.00 -0.070000 1.992370 0.00 -0.035000 1.992370
N802 0.00 -0.170000 1.824001 0.00 -0.085000 1.824001
C803 0.00 -0.080000 1.964309 0.00 -0.040000 1.964309
C804 0.00 -0.070000 1.992370 0.00 -0.035000 1.992370
```

Figure 14: Structure of a PRM file.

2.7 Web Development

The idea for this thesis project is to create a website which will turn the algorithm into an easily accessible tool for the users, without installation and account creation.

This work was done using PHP (Hypertext Preprocessor), which is a widely-used, free, and efficient server scripting language, and a powerful tool for making dynamic and interactive web pages [72] (Figure 15).

Figure 15: PHP.

Using PHP the site downloads the files input, which can be acquired from LigParGen, into text files for the Python code to read them. Then again using PHP the site uploads the produced files that the user needs to run his simulation. Lastly in the same way the user can download the example which is on the site and also read the manual (Annex I), these help the users understand how to use the tool.

In addition to the previous PHP code, of course HTML (Hypertext Markup Language) [73]and CSS (Cascading Style Sheets) [74] codes were implemented.

3. RESULTS

3.1 Workflow of the algorithm

The algorithm assumes that the protein and the ligands have been prepared and aligned. This can be done from any OPLS that the user might want. The steps in chapters 3.1.1 and 3.1.2 are not part of the algorithm and the user has to account for them. On the other hand the steps explained in chapters 3.1.3 to 3.1.7 are part of the algorithm and they have been automated. A workflow can be seen in Figure 16. On the left hand side the first two steps refer to chapters 3.1.1 and 3.1.2 respectively. The steps “Atom renaming”, “Partial charge distribution”, “Dual topology file”, “Hybrid PDB file”, and “Complex PDB file” are explained in chapters 3.1.3, 3.1.4 and 3.1.5. On the right hand side the steps that take part after the use of VMD are presented. These steps are utilized for both the complex leg and the solvent leg systems and they are explained in chapters 3.1.6 and 3.1.7.

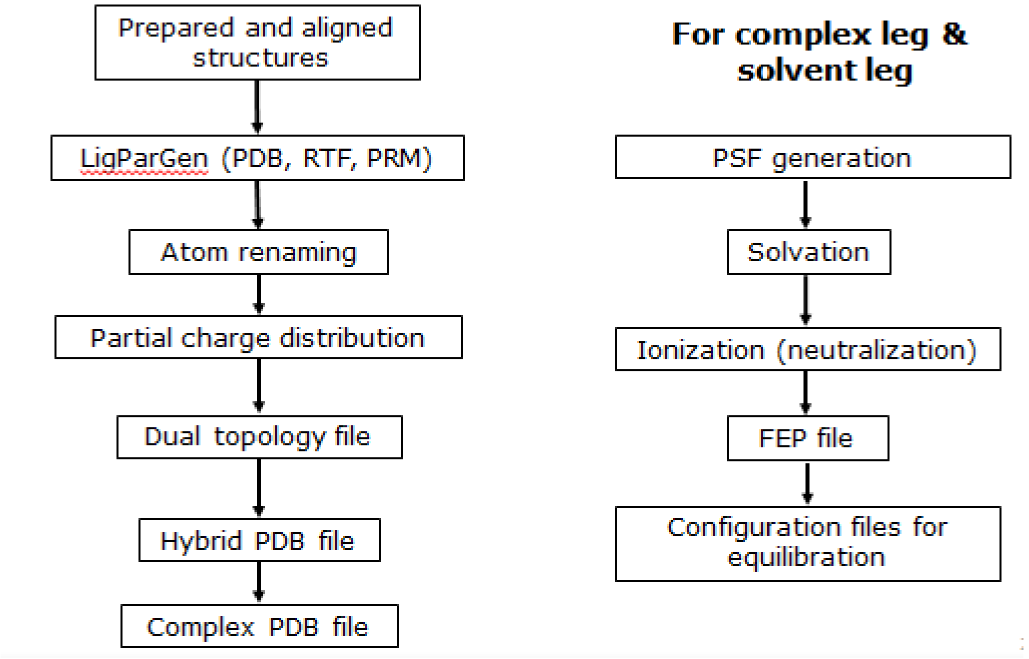

Prepared and aligned structures
LigParGen (PDB, RTF, PRM)
Atom renaming
Partial charge distribution
Dual topology file
Hybrid PDB file
Complex PDB file
For complex leg & solvent leg
PSF generation
Solvation
Ionization (neutralization)
FEP file
Configuration files for equilibration

Figure 16: Workflow of the algorithm.

3.1.1 Protein preparation, and ligand design, preparation & alignment

Here are presented the steps one has to perform in Maestro Schrödinger suite [75] to prepare the structures.

1. Prepare the protein structure (through Protein Prep Wizard [76, 77]).

In the first stage of preparation Protein Prep Wizard adds hydrogens, assigns correct bonding classes, creates disulfide bonds, removes water molecules, if necessary and completes side chains and loops if they are missing. The latter is done with the Schrödinger Prime software which recognizes which side chains are missing in relation to the protein sequence and completes them, while providing for the tertiary structure of the missing loops by Homologous Modeling. Homologous Modeling is a computational prediction technique of the tertiary structure of a protein, which uses known protein structures as models for modeling unknown structures.

In the second stage of preparation, it is possible to remove molecules co-crystallized with the protein as well as chains of the protein itself. At this stage, the different protonation and tautomeric states of the co-crystallized molecules, such as ligands are predicted. The predictions are made in a specific pH range with Schrödinger's Epik software.

In the third and final step of the preparation, optimization of the hydrogen bonding network is effected by changing the orientation of hydroxyl groups (-OH) and hydrothio groups (-SH), water molecules, Asparagine and Glutamine amides as well as the imidazole ring of Histidine. Also, with the PROPKA tool the protonation states of Histidine, Aspartic acid, Glutamic acid and the Histidine tautomers are contemplated. Finally, a limited minimization of the energy of the structure is achieved. It is limited because atoms other than hydrogen are not allowed to move more than 0.30 Å.

2. Split the prepared structure into protein, ligand, water molecules (right click on the entry), and use the reference ligand structure as a template to design the analogue, through "3D Builder". It is advised not to use "2D Sketcher", because "2D Sketcher" will miscalculate the coordinates, so it is best to utilize "3D Builder". This is done, so that the common part of the analogue and the reference ligand have the same coordinates (and so both molecules will exist in the area of the binding pocket). This is part of an assumption that will be mentioned in the ligand alignment section.

3. Prepare only the analogue, following the usual procedure (through LigPrep).

Keep the correct configurations, regarding chiralities and charges (these should be the same as in the reference ligand). The agreement in chiralities is part of the assumption that will be mentioned in the ligand alignment section. Charges should be the same between reference and target ligands in FEP, because the introduction of a charge, during the simulation, can lead to weak convergence or huge errors.

4. Select the reference ligand and the analogue, and align them (through Flexible Ligand Alignment):

- Common scaffold alignment
- Maximum common substructure

This will flexibly align the analogue to the reference ligand. This is done, based on the assumption that the common part of a mutant ligand will have the same binding mode as that of the reference. By implementing the above, we achieve a really good overlap of the potential energies of the two molecules in a transformation, so that the differences are minimized and the algorithm converges.

5. Export the reference and mutant ligands, as well as the prepared protein, as PDB files. In the end open the protein and delete all hydrogens.

3.1.2 Parameterization of the ligands, using the LigParGen server

Here is explained how to obtain the coordinate, parameter and topology files from LigParGen discussed in chapter 2.2.3.

1. Upload the ligands (one at a time) to the LigParGen server.

2. Select **0 optimization iterations** (this messes with the coordinates and the ligands won't be aligned after that), and choose the **1.14*CM1A** charge model (it has produced easier-to-deal with results in the past), and submit.

3. The files of interest are the PDB (coordinates), PRM (parameters) and RTF (topology).

3.1.3 Atom renaming

LigParGen introduces some inconsistencies with the atom names, which will cause NAMD to produce fatal errors. Specifically, some atoms that remain the same, during the transformation, will have different names assigned, and different atoms will have the same names. This causes NAMD to identify same atom names, take part in different angles, resulting in a fatal error. In Figures 17 and 18 one can see that the atom that is named ***HON*** in the reference ligand (Figure 17) does not have the same name in the mutant ligand (Figure 18), but it is instead named as ***HOQ***. In addition a different atom in the reference ligand has the name ***HOQ*** as well.

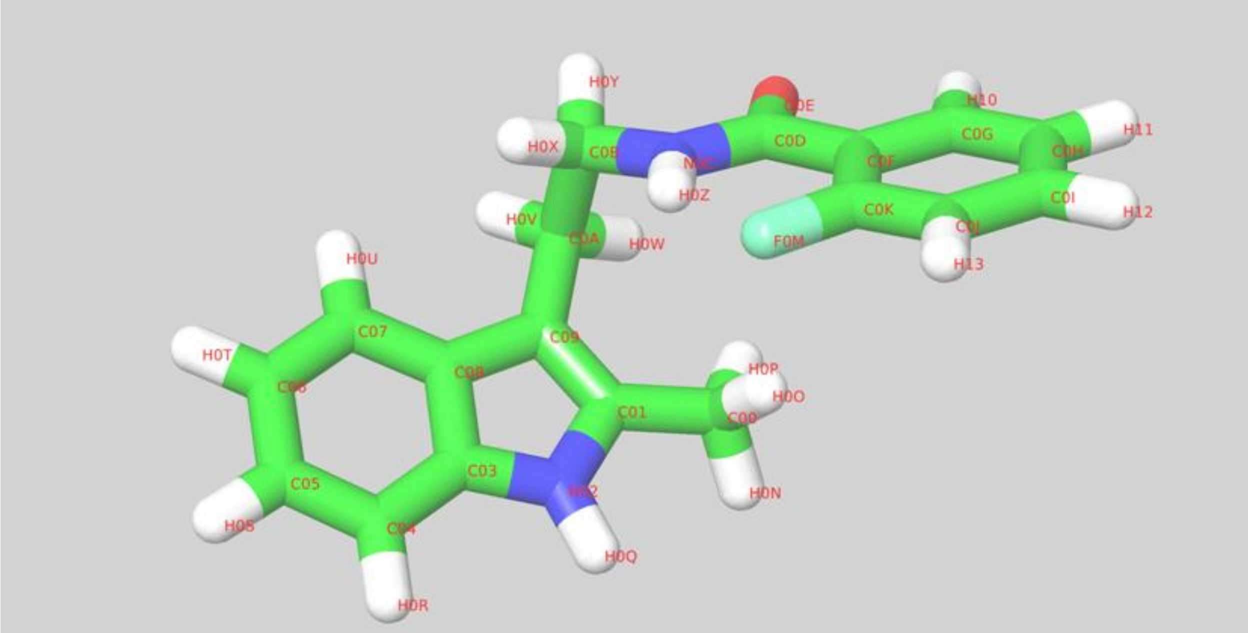


Figure 17: Reference ligand CK666, atoms named from LigParGen.

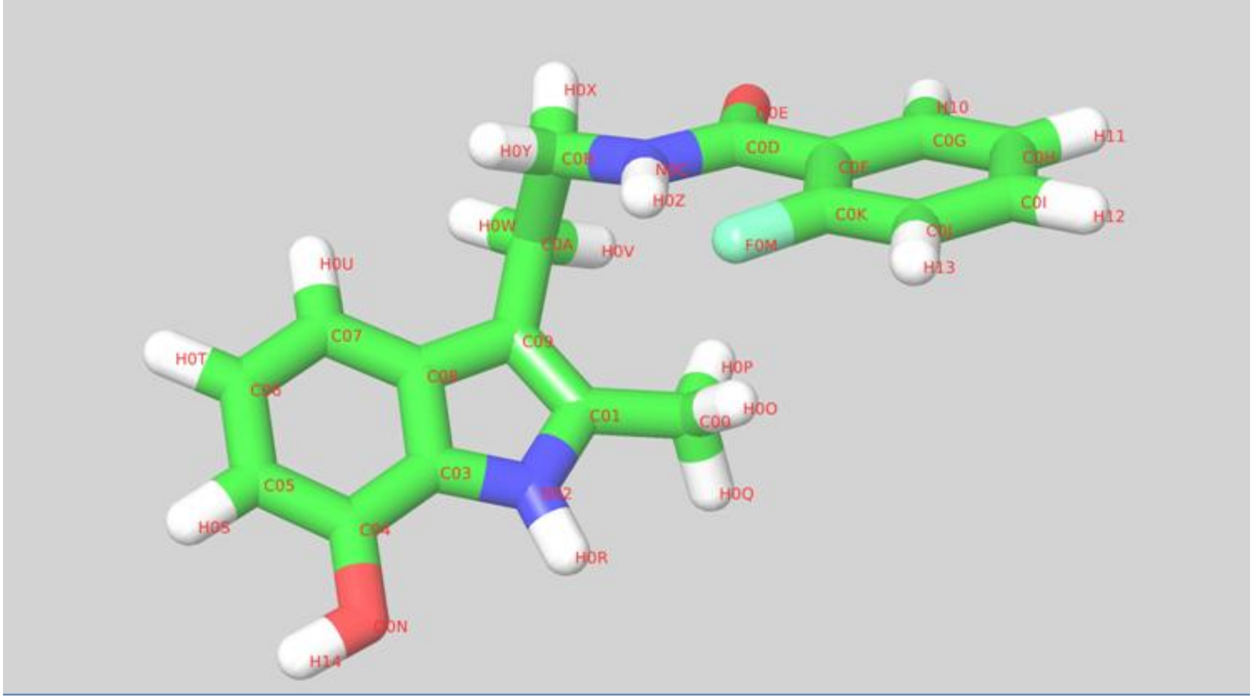


Figure 18: Mutant ligand AI003, atoms named from LigParGen.

In order to correct the inconsistency a script in Python was written, which renames all the atoms of the reference ligand giving them new names, according to their coordinates. Then, always according to their coordinates, the atoms of the mutant ligand are renamed (Figure 21). This procedure creates two new PDB files with the atoms being renamed, one for the reference ligand and one for the mutant ligand. Thus a new naming scheme needs to be applied to the RTF and PRM files of both the reference and the mutant ligand. In the end we have six new files, three for each ligand, all renamed properly. In Figures 19 and 20 one can now see that the same atoms have the same names in both reference and mutant ligands. The only atoms that have different names are the ones that are modified, in our case ***H14***, meaning that the area taking part in the mutation has been located.

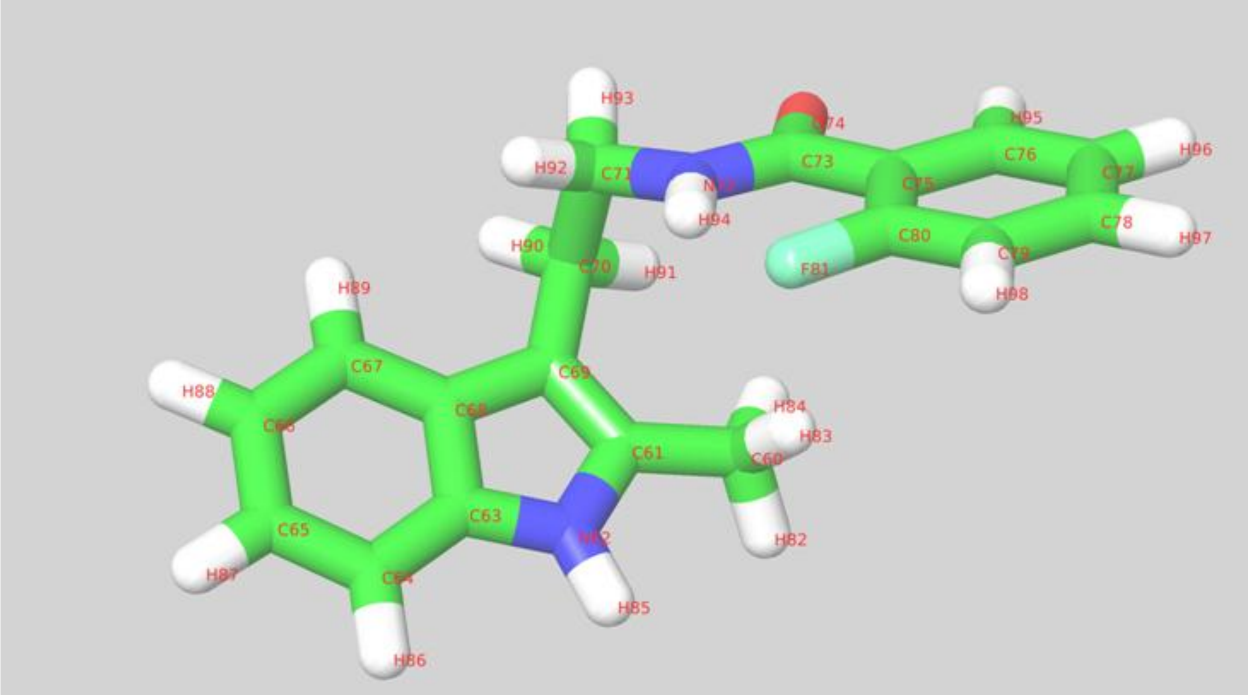


Figure 19: Reference ligand CK666, after the renaming.

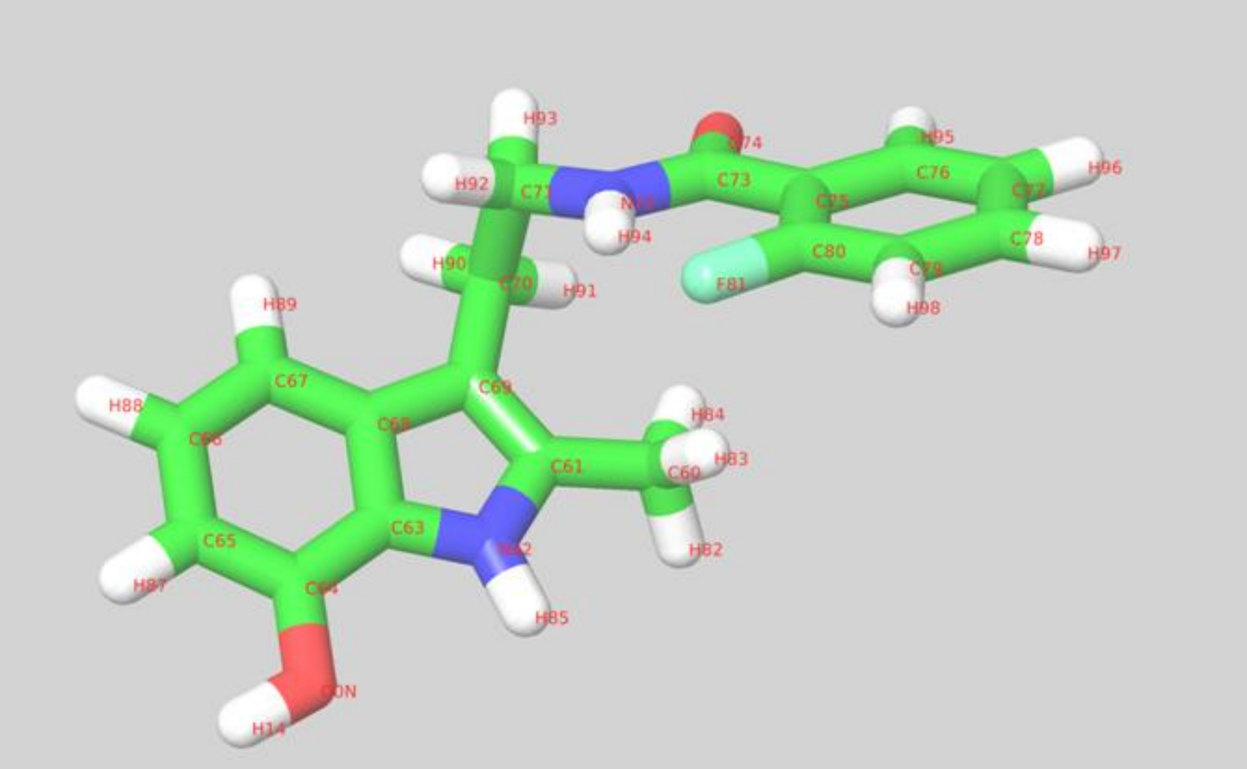
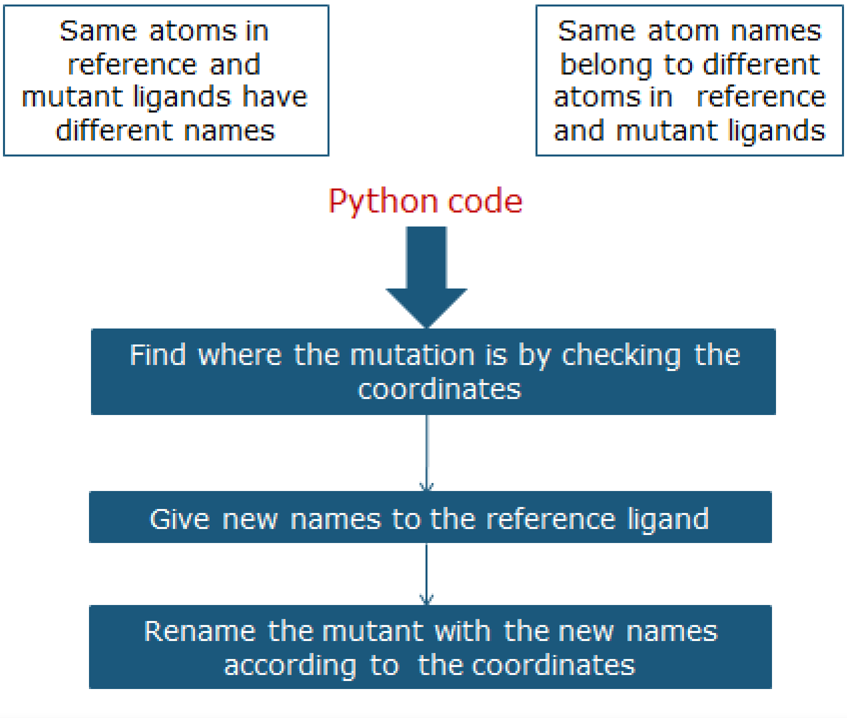


Figure 20: Mutant ligand AI003, after the renaming.


Same atoms in reference and mutant ligands have different names

Same atom names belong to different atoms in reference and mutant ligands

Python code

Find where the mutation is by checking the coordinates

Give new names to the reference ligand

Rename the mutant with the new names according to the coordinates

Figure 21: Atom renaming algorithm.

3.1.4 Partial charge re-distribution

As explained in chapter 2.1.4, NAMD uses the dual topology technique. During which both the reference and the mutant ligand exist in the same topology file (a hybrid RTF).

Creating a hybrid RTF file can be done by just including all the atoms of the two ligands in the same file. However, to reduce the perturbations during the transformation (and help the algorithm converge), a technique is applied, where the common part between the reference and the mutant ligand stays the same (charge-wise), during the simulation, and only the mutation area changes. This is based upon the assumption that, in a large-enough ligand, the partial charges of the atoms away from the mutation area are going to change only by a small amount that can be neglected. The workflow can be seen in Figure 22.

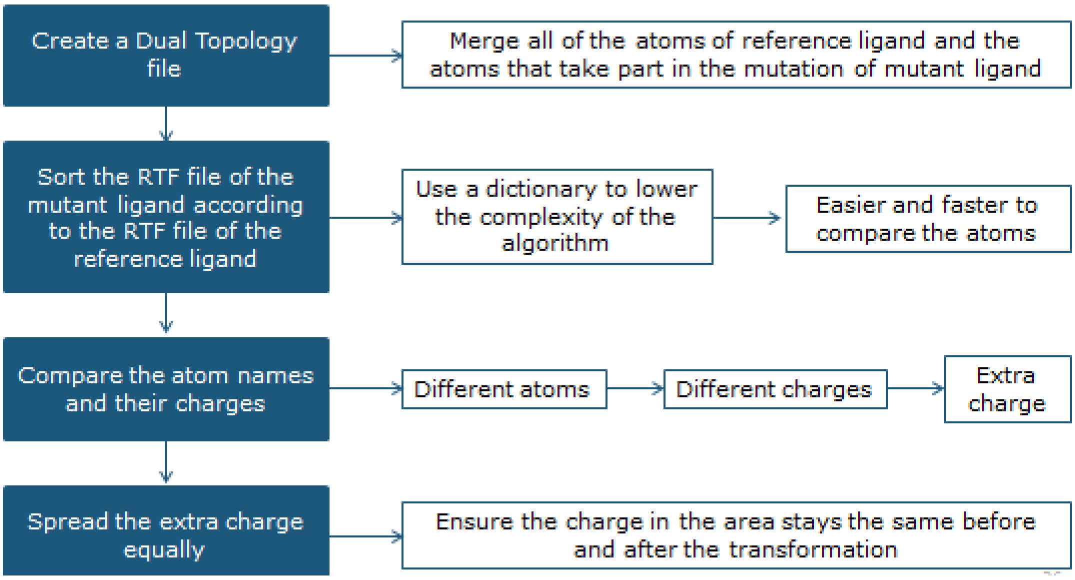

Create a Dual Topology file
Merge all of the atoms of reference ligand and the atoms that take part in the mutation of mutant ligand
Sort the RTF file of the mutant ligand according to the RTF file of the reference ligand
Use a dictionary to lower the complexity of the algorithm
Easier and faster to compare the atoms
Compare the atom names and their charges
Different atoms
Different charges
Extra charge
Spread the extra charge equally
Ensure the charge in the area stays the same before and after the transformation

Figure 22: Dual Topology workflow.

If the above is utilized, then the mutation area in the reference ligand and the mutation area in the mutant ligand need to have the same charge (since the charge of the common part stays the same), in order for the total charge of the two molecules to be the same. For this to be achieved, a distribution of the extra charge that is introduced in the mutation area by the mutation needs to occur. The extra charge is distributed to a number of neighbouring-to-the-mutation atoms, in the mutant ligand. The amount of atoms included is chosen aiming that the charge that gets added to each atom is below or equal to a limit.

The partial charge redistribution is explained in Figure 23. Let Sum A be the sum of the partial charges of the atoms that are being modified in the reference ligand, and Sum B, be the sum of the partial charges of the atoms that are being modified in the mutant ligand. These two sums should be equal, as it has been explained above, but they are not. This difference is the extra charge that needs to be distributed to the neighbouring-to-the-mutation atoms, in the mutant ligand. Let that charge be value X. The only way to distribute this value X, is to divide it with the number of atoms that take part in the modification. Let the number of the atoms be N and the quotient of the deviation be DIV. This DIV number actually is the very-very small charge we want to distribute. So we need it to be smaller than 0.02, for the FEP algorithm to converge. If DIV is larger than 0.02, we have to include more atoms in our calculation.

Now the extra charge has been distributed equally, and so the atoms of the mutant ligand that have taken part in our calculation eventually have new partial charges. The most important thing is that these differences in the changes are so small that do not affect the whole ligand, but only the mutated area, (now Sum A = Sum B). This is how it is confirmed that the area that takes part in the mutation keeps its charge and therefore the procedure may continue.

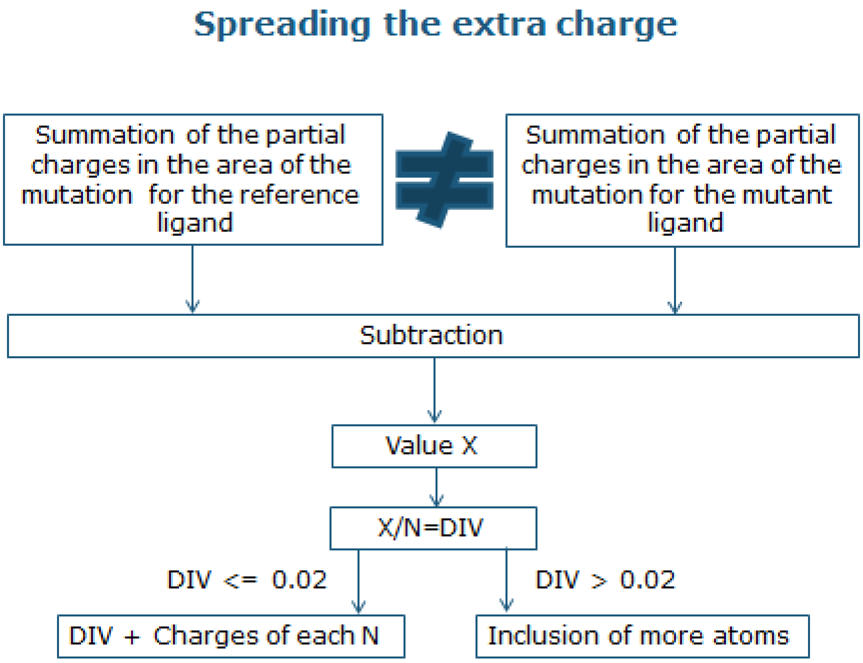

Spreading the extra charge
Summation of the partial charges in the area of the mutation for the reference ligand
≠
Summation of the partial charges in the area of the mutation for the mutant ligand
Subtraction
Value X
X/N=DIV
DIV <= 0.02
DIV > 0.02
DIV + Charges of each N
Inclusion of more atoms

Figure 23: Partial Charge re-distribution.

These atoms will be included in the hybrid RTF but, in addition to their charges, their names must be different from their reference counterparts as well, in order for NAMD to recognise them as different atoms.

The hybrid RTF produced will include the common part of the two ligands, the mutation area before the transformation, and the mutation area after the transformation. So, essentially, it will include the whole reference ligand and, from the mutant ligand, only the mutant atoms that were affected by the partial charge distribution procedure (the newly appearing atoms of the mutation and the neighbouring-to-the-mutation atoms that had their charges changed, but with their new charges and names).

Later, it will be signified, which atoms are the ones that will disappear during the simulation (the disappearing, due to the mutation, reference atoms and the atoms that will have their charges changed), which ones will stay the same (the common part of the two ligands minus the atoms that will have their charges changed), and which ones will appear (the newly appearing, due to the mutation, atoms and the atoms that had their charges changed, but with their new charges).

3.1.5 Hybrid RTF & PDB creation, and complex formation

The two ligands need to be combined in a hybrid RTF, as discussed above. In addition to a hybrid RTF, a hybrid PDB file needs to be produced, accordingly.

Because the mutant atoms that had their charges changed had also their names changed (with respect to their reference counterparts), the PRM file or the mutant ligand needs to be updated, to account for the new names.

For the complex leg of the simulation, it is necessary to have a complex PDB file (Figure 24). So, the protein PDB and the hybrid PDB files need to be combined, simply by concatenating the two files and fixing the atom numbering of the ligand at the end of the new file.

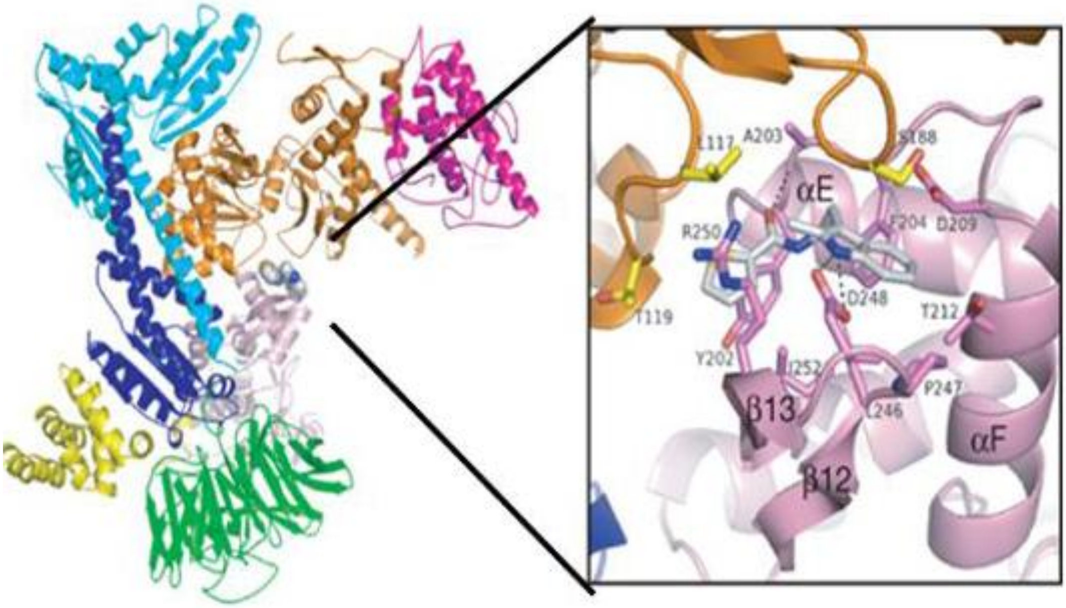


Figure 24: Left: Binding of the CK-666 inhibitor to the Arp2 (pink ribbon) and Arp3 (orange ribbon) subunits. Right: CK-666 mooring cavity enlarged [78].

3.1.6 Complex leg & Solvent leg system PSF generation, solvation and ionization, using VMD

The preparation of the system can be performed through the VMD GUI, but the server uses a tcl script that calls the VMD packages *psfgen*, *solvate* and *autoionize*, in order to create the PSF, solvate the system in a water box with limits that are in a 10 A distance from the atom with the greatest coordinate in each direction, and insert counterions to electrically neutralize it, respectively. In addition, the script will measure the values of the minimum, the maximum and the centre of the box, which will be needed later.

The final PSF and PDB files from the preparation that VMD did are *ionized.psf* and *ionized.pdb*, respectively.

The procedure for the solvent leg is the same as in the preparation of the complex leg system.

In Figures 25 and 26 one can see the VMD's representation on ionized.pdb file for the complex and the solvent systems respectively.

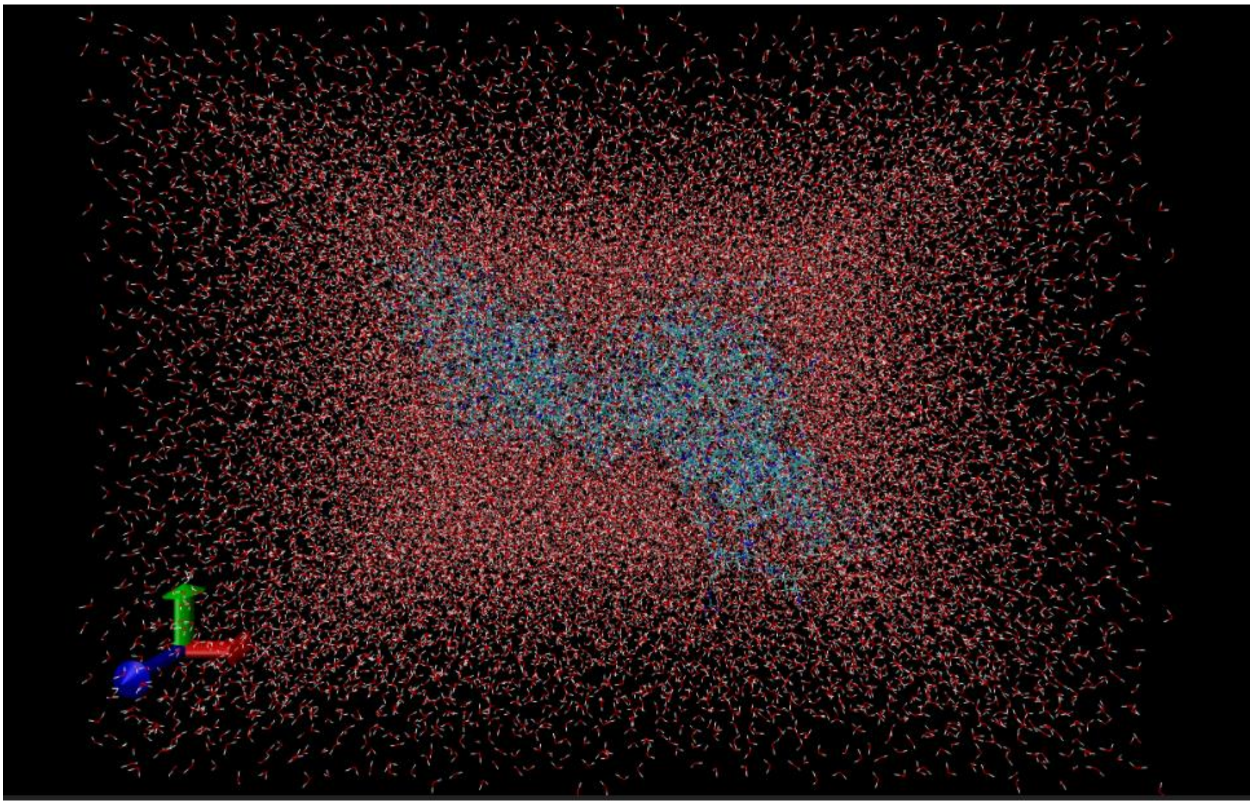


Figure 25: Ionized.pdb file for complex system.

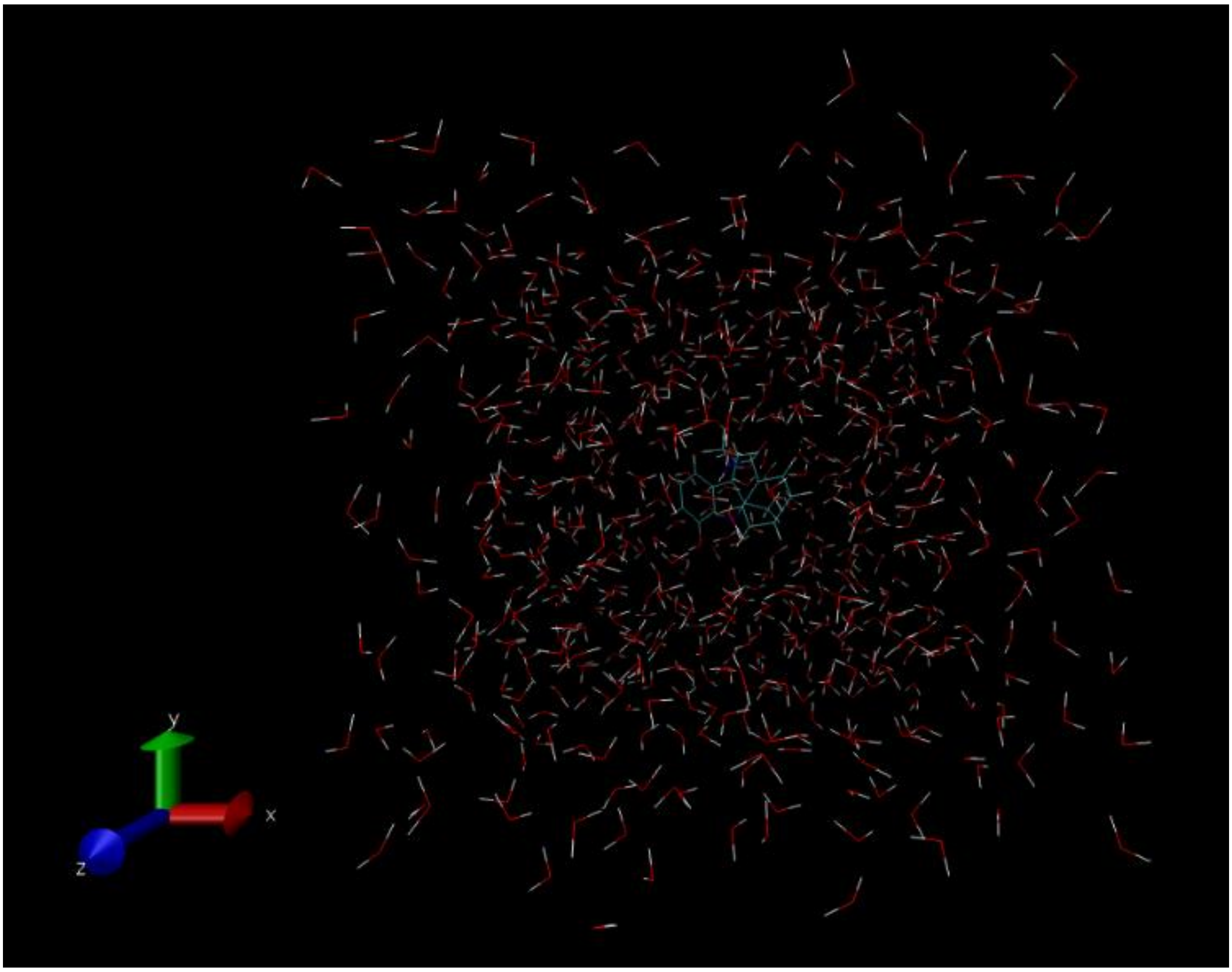


Figure 26: Ionized.pdb file for solvent system.

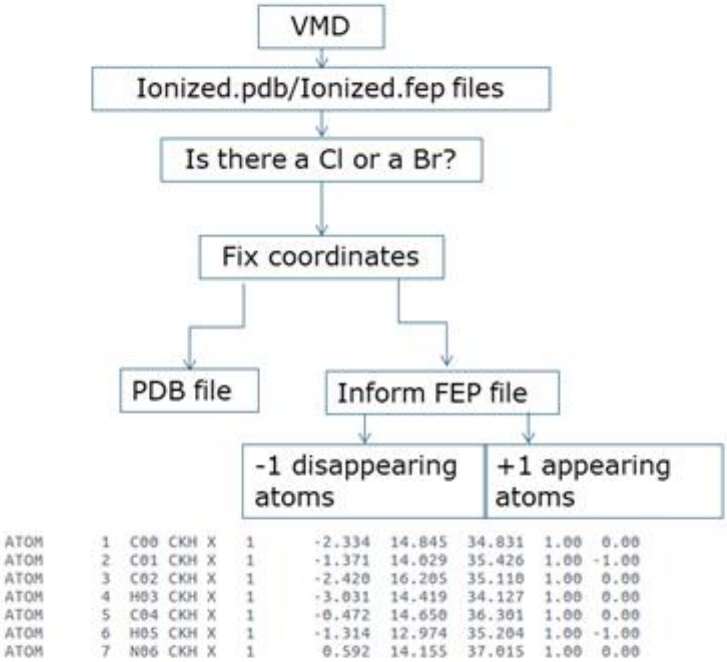
3.1.7 FEP files update and correction of two-character-named chemical elements error in PDB files

As discussed before, the FEP files are used by NAMD to identify, which atoms disappear, which atoms stay the same, and which atoms appear, during the simulation.

These are the atoms selected in the partial charge distribution step. The FEP file is simply a copy of *ionized.pdb* with a slight modification. Inside this FEP file, we will use the columns dedicated for the B-factor values, in order to specify, which atoms disappear, which atoms stay the same, and which atoms appear, during the simulation. This is done by setting the following values:

- A value of -1.00 signifies that the atom will disappear.
- A value of 0.00 signifies that the atom will stay the same.
- A value of 1.00 signifies that the atom will disappear.

The algorithm will also correct an error VMD produces. If there is an atom, the name of which consists of two letters (such as Cl or Br), then VMD will fail to read its coordinates, during the PSF generation, and instead will print 0.000 in all x, y and z, in *ionized.pdb*. The algorithm corrects that, simply by reading the correct coordinates from the initial PDB files and updates the ionized.pdb file. The above are explained in a diagram in Figure 27.



```
ATOM      1  C00 CKH X   1      -2.334  14.845  34.831  1.00  0.00
ATOM      2  C01 CKH X   1      -1.371  14.029  35.426  1.00 -1.00
ATOM      3  C02 CKH X   1      -2.420  16.205  35.110  1.00  0.00
ATOM      4  H03 CKH X   1      -3.031  14.419  34.127  1.00  0.00
ATOM      5  C04 CKH X   1      -0.472  14.650  36.301  1.00  0.00
ATOM      6  H05 CKH X   1      -1.314  12.974  35.204  1.00 -1.00
ATOM      7  N06 CKH X   1       0.592  14.155  37.015  1.00  0.00
```

Figure 27: VMD output files.

3.1.8 Input files

The inputs that the algorithm needs are seven files:

- The prepared protein structure from Maestro.
- Two PDB files, one for each ligand, obtained by LigParGen.
- Two RTF files, one for each ligand, obtained by LigParGen.
- Two PRM files, one for each ligand, obtained by LigParGen.

3.1.9 Implementation of the algorithm

The algorithm is written in Python. All of the steps above are implemented in different scripts, since they are used for different purposes at different times (Figure 28).

The first script takes care of LigParGen's inconsistency and renames the atoms of the reference ligand and the mutant ligand for all files. It is called names.py.

Then sort.py is called in order to sort the mutant ligand, according to the reference ligand. This step is only important because the next script dual.py needs the atoms of the mutant ligand to be in the same sequence as the atoms of the reference ligand in order to compare their names and partial charges. Also the dual.py creates the hybrid.pdb, the hybrid.rtf and updates the mutant's ligand PRM file.

In the next a complex.pdb file needs to be created which is the merge of the PDB file of the protein and the hybrid.pdb file. Complex.py implements that.

In order to use VMD without the GUI and automate the whole procedure, split_chains.py is called. It reads how many chains the protein has and creates the scripts that VMD needs to run, according to the number of chains it recognizes. After VMD has run successfully min-max.py prints the values for the minimum, the maximum and the centre of the box to a text file, in order to later provide this information to the user.

Last, but not least, fep.py is called to update the FEP files about the atoms that have disappeared, and the atoms that have appeared. Also it corrects VMD's inconsistency that occurs with atoms that their names have two characters as explained in chapter 3.1.7.

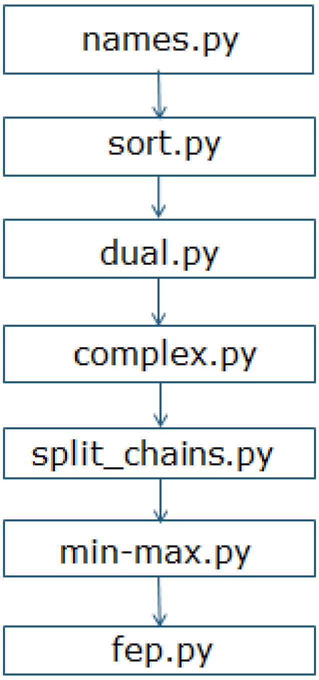


Figure 28: A workflow for the scripts.

3.1.10 Output

The algorithm gives as an output a zip file that contains all the files that have been created after it has completed its run.

The file that the user can download as "files.zip" should contain:

- Two subfolders, “complex” and “solvent”.
- The six renamed files (PDB, RTF, PRM) for both ligands.
- The two hybrid files named as “ligand.pdb” and “ligand.rtf”.
- The renamed PRM file after the creation of the two hybrid files, “updated.prm”.
- The “complex.pdb” file.
- The “fep.tcl” script that NAMD needs to run the simulation.
- A file with the OPLS-AA parameters of proteins (par_opls_aam.inp).
- A file with the OPLS-AA topology of proteins (top_opls_aam.inp).

Into the “complex” file the user should be able to see the following:

- The PDB files for each of the protein’s chains and the ligand.
- Files named “ionized.pdb”, “ionized.fep”, “ionized.psf”.
- Files named “ionized_new.pdb” and “ionized_new.fep” which contain the updated ionized files, after the “fep.py” script.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen”, and “VMD_prepare_complex_after_gui_autopsf”.
- The files that VMD gives as an output, “complex_wb.log”, “complex_wb.pdb”, “complex_wb.psf”, “psf-complex.psf”, “psf-complex.pdb”.
- The NAMD configuration files for complex in the file “conf_comlex”.

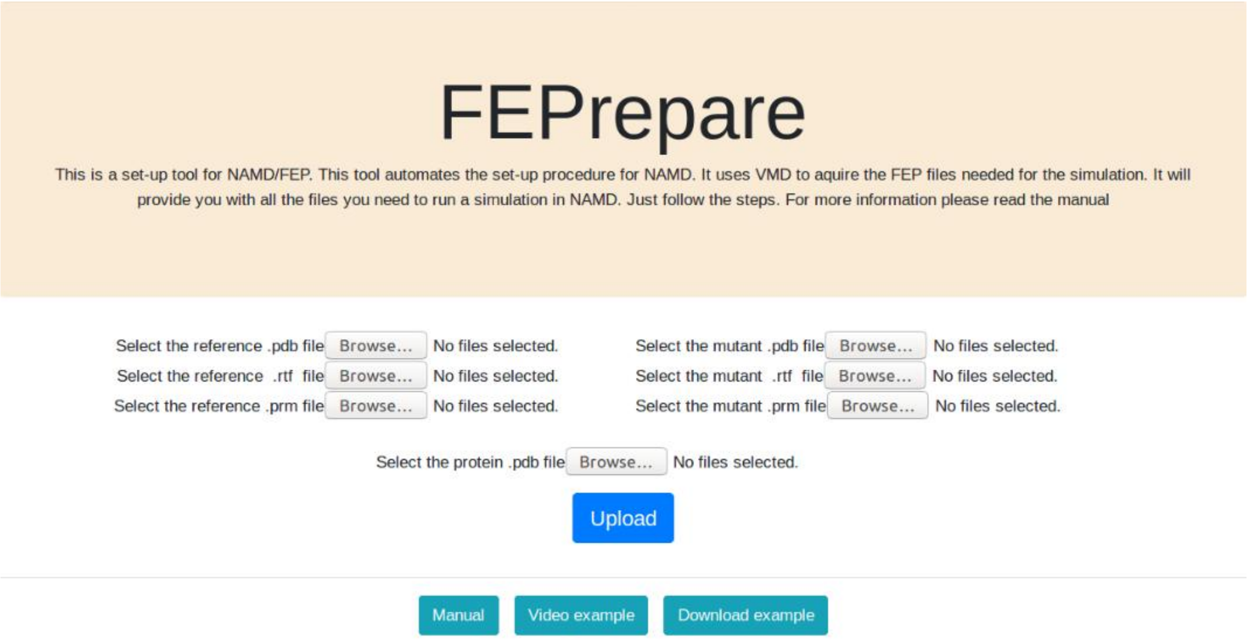
Into the “solvent” file the user should be able to see the following:

- Files named “ionized.pdb”, “ionized.fep”, “ionized.psf”.
- Files named “ionized_new.pdb” and “ionized_new.fep” which contain the updated ionized files, after the “fep.py” script.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen_solv”, and “VMD_prepare_ligand_after_gui_autopsf”.
- The files that VMD gives as an output, “ligand_wb.log”, “ligand_wb.pdb”, “ligand_wb.psf”, “psf-solvated.psf”, “psf-solvated.pdb”.
- The NAMD configuration files for complex in the file “conf_solvent”.

3.2 Implementation of the Web-server

As mention in chapter 2.7, the automation of the set-up procedure for NAMD/FEP is important to be implemented into a code which can be used from people with no programming skills. In order to achieve that a web-based server has been created (Figure 29) and can be accessed through:

http://feprepare.vi-seem.eu/



FEPrepare

This is a set-up tool for NAMD/FEP. This tool automates the set-up procedure for NAMD. It uses VMD to aquire the FEP files needed for the simulation. It will provide you with all the files you need to run a simulation in NAMD. Just follow the steps. For more information please read the manual

Select the reference .pdb file Browse… No files selected.
Select the reference .rtf file Browse… No files selected.
Select the reference .prm file Browse… No files selected.

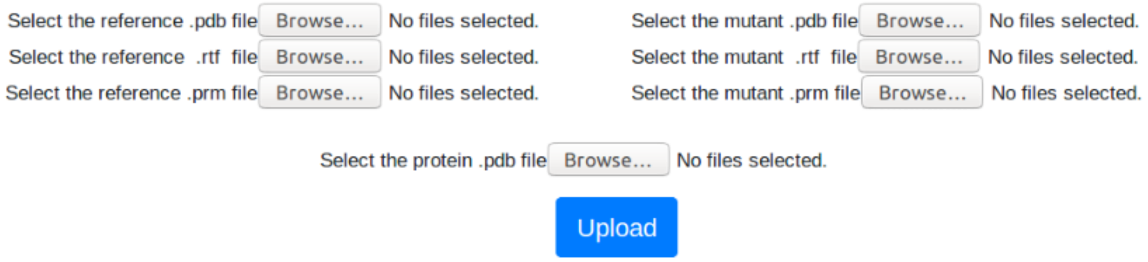
Select the mutant .pdb file Browse… No files selected.
Select the mutant .rtf file Browse… No files selected.
Select the mutant .prm file Browse… No files selected.

Select the protein .pdb file Browse… No files selected.

Figure 29: FEPrepare.

The web server receives as input the seven files discussed above, in chapter 3.1.8. It executes the scripts that are written in Python, runs VMD and produces all the files discussed in chapter 3.1.10. The web server is written in PHP and uses HTML.

The user can select the files needed to be uploaded, and then hit "Upload". PHP will upload the files to the server and download them in text files for the Python scripts to process them (Figure 30).



Select the reference .pdb file Browse… No files selected.
Select the reference .rtf file Browse… No files selected.
Select the reference .prm file Browse… No files selected.

Select the mutant .pdb file Browse… No files selected.
Select the mutant .rtf file Browse… No files selected.
Select the mutant .prm file Browse… No files selected.

Select the protein .pdb file Browse… No files selected.

Figure 30: Input for the web-server.

After the python scripts have run, PHP uploads the files to the web-server, and the user can then download them and save them to his personal computer (Figure 31).

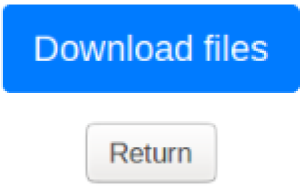


Figure 31: Download all files as zip.

The user can download the manual, by hitting “Manual”. A video example is available for all users by hitting “Video example”, and last but not least an example can be downloaded by hitting “Download example” (Figure 32).

Figure 32: Manual- Video example- Download example.

3.3 Test Cases

In this chapter, some of the use cases are shown, in detail, in order to better understand the outputs the web-server provides along with their validation. All our tests were conducted on Arp2/3 protein and one of its know inhibitors CK666 and its mutants.

3.3.1 Arp2/3

The inhibition of Arp2/3 has shown to lead to the control of plasticity of nerve synapses [79], contribute to the predominance of the helper with regard to lamellipodiums [80], regulate the shape and movement of the endosomes [81] and changes the mechanism of regulation of ion channels by the cortactin protein [82].

Despite the fact that, since the discovery of Arp2/3 in 1994 [83] to date, significant knowledge has been gained on its role in cytoskeleton dynamics, its involvement in cell mobility, the formation of actin precursors and ultimately metastasis of cancer has not been fully investigated.

In Figure 33a a representation of the Arp2/3 cluster organization in subunits is presented. In 33b the iosolated crystalline structure of the Arp2/3 protein complex from the bos taurus organism (code from the Protein Data Bank (PDB): 1A8K) can be seen. The modules appear in a ribbon representation of different colors. In 33c the ribbon representation of the activated conformation of the Arp2/3 protein complex as predicted are presented. The colors are the same as in 33b [84].

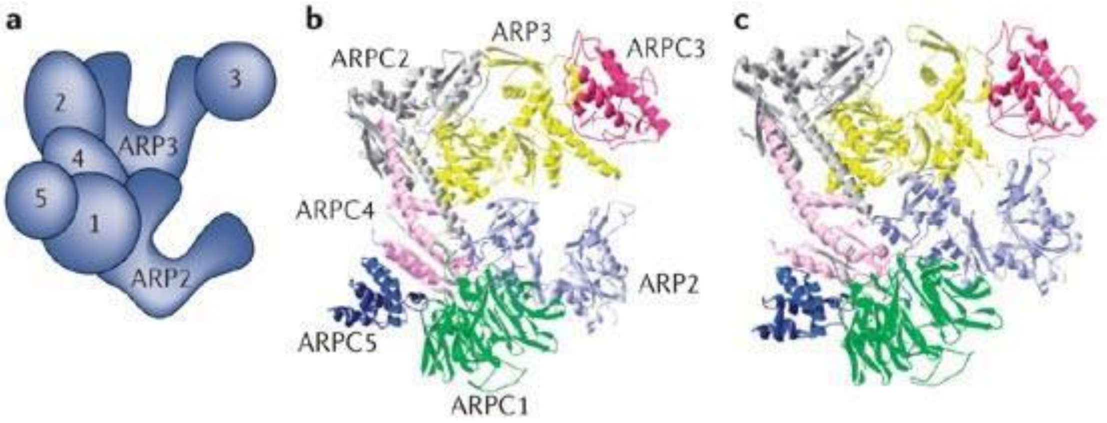


Figure 33: Arp2/3 [84].

Inhibition of an enzyme can be accomplished by binding a molecule to it, causing the activity of the protein to decrease. Since many diseases are associated with specific protein functions, many drugs are enzyme inhibitors. There are two main types of suspension: competitive and non-competitive suspension. During competitive inhibition, the receptor substrate and the inhibitor both bind to the active site of the protein by binding one to antagonize the binding of the other. In non-competitive inhibition, the inhibitor binds allosterically, i.e. in a protein cavity other than the active site. In this case, the binding of the substrate to the active site is not inhibited by binding of the inhibitor but may be affected by conformational changes in protein caused by binding of the inhibitor.

A know inhibitor for Arp2/3 is CK666 and it is visualized in Figure 34.

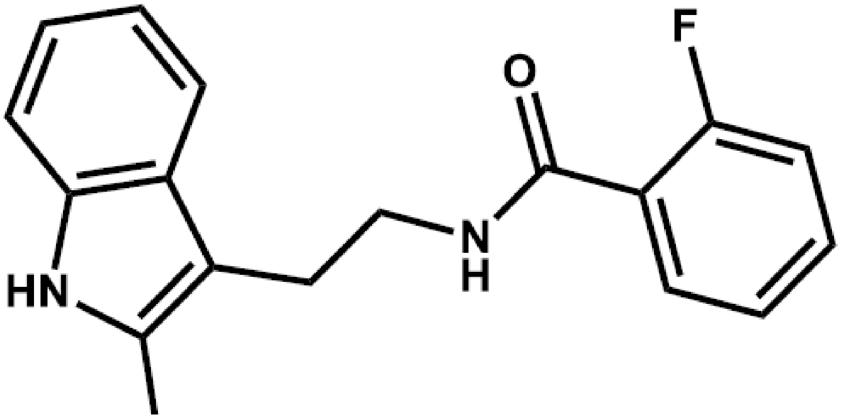


Figure 34: CK666 inhibitor

In this project CK666 was tested against 15 other mutant ligands. AI003, AI007, AI015, AI062, AI064, AI065, AI066, AI067, AI068, AI071, AI078, AI079, AI086, AI093, AI094.

3.3.2 Use Case, CK666-AI003

After the user has uploaded the files needed to FEPrepare, the algorithm downloads the files, reads them and starts renaming the atoms as explained in chapter 3.1.3. After the first script "names.py" has run the two ligands will look like Figures 35 and 36.

In Figures 35 and 36 it is shown that all the atoms in both ligands are the same, apart from atoms ***H86*** (in CK666), and ***O0N*** and ***H14*** (in AI003)***.*** This is the area that takes part in the mutation. A hydrogen atom is mutated into oxygen and hydrogen.

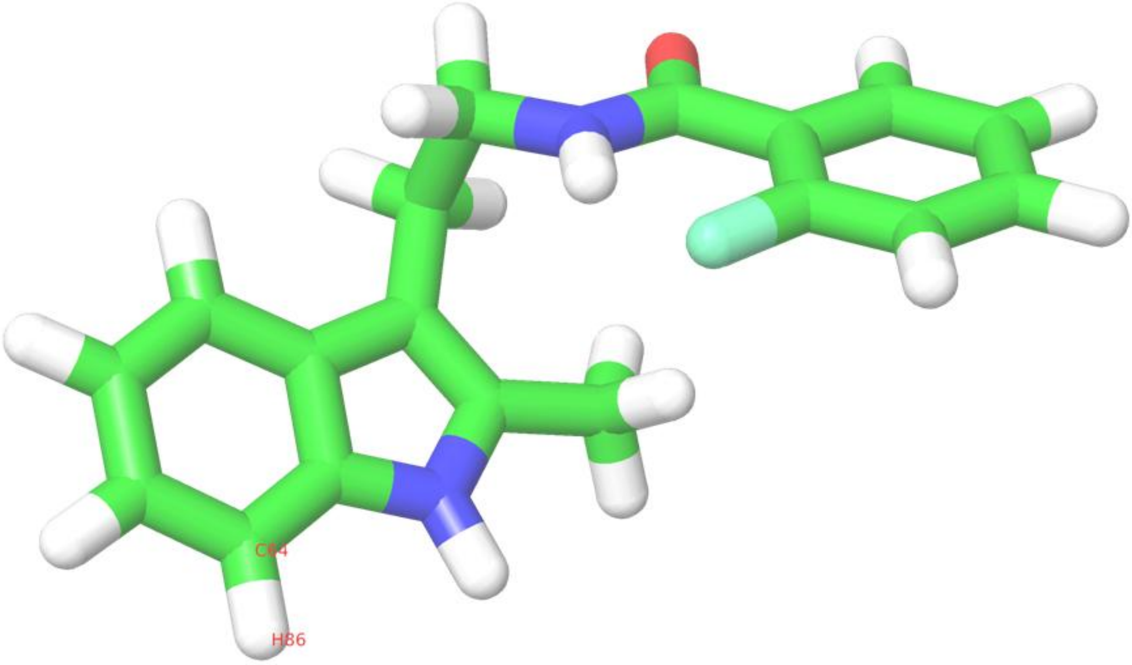


Figure 35 : Reference ligand CK666.

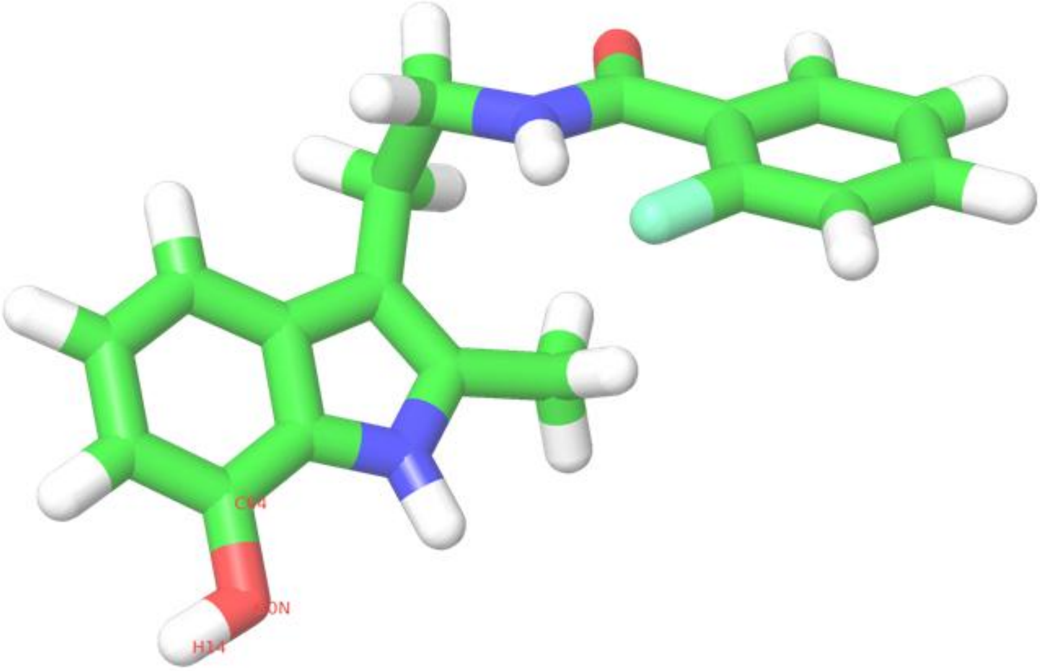


Figure 36: Mutant ligand AI003.

Although there is a small modification in the atoms, an extra charge in the area is introduced, which cannot be neglected. As explained in chapter 3.1.4 this extra charge needs to be distributed. The algorithm runs “dual.py”, the code takes care of the extra charge and merges the atoms of reference ligand (CK666) with the atoms that take part in the mutation of the mutant ligand (AI003), in our case ***O0N***, ***H14***. So far the outcome of the algorithm is the same as if the calculations we performed manually.

Now the hybrid.rtf and hybrid.pdb files have been created and the PRM file of AI003 has been updated. The next step is to combine the protein with the hybrid.pdb file in order to create the complex.pdb, the script “complex.py” performs that correctly as well.

Since the purpose of the project is to totally automate the procedure the user does not have to use the VMD GUI. In this direction a script that reads the protein and splits it into the correct numbers of chains has been implemented. Also the same script “split_chains.py” generates correctly the files that VMD needs to run its calculations.

In the end it is validated that the algorithm has produced the correct output, by visualising the two ionized.pdb files that are created after “fep.py” has run, as it is explained in chapter 3.1.6.

These outputs are totally compatible with NAMD since the simulation finishes correctly and provides the same results as when the simulation was run with manually created inputs.

3.3.3 Use Case, CK666-AI007

As explained above, after the user has uploaded the files needed to FEPrepare, the algorithm downloads the files, and reads them. After the first script “names.py” has run, the two ligands will look like Figures 37 and 38, in these figures the licorice representation from VMD has been chosen to better visualize the ligands.

In Figures 37 and 38 it is shown that all the atoms in both ligands are the same, apart from atoms ***H86*** (in CK666), and ***Cl0*** (in AI007)***.*** This is the area that takes part in the mutation. In other words a hydrogen atom is mutated into chloride.

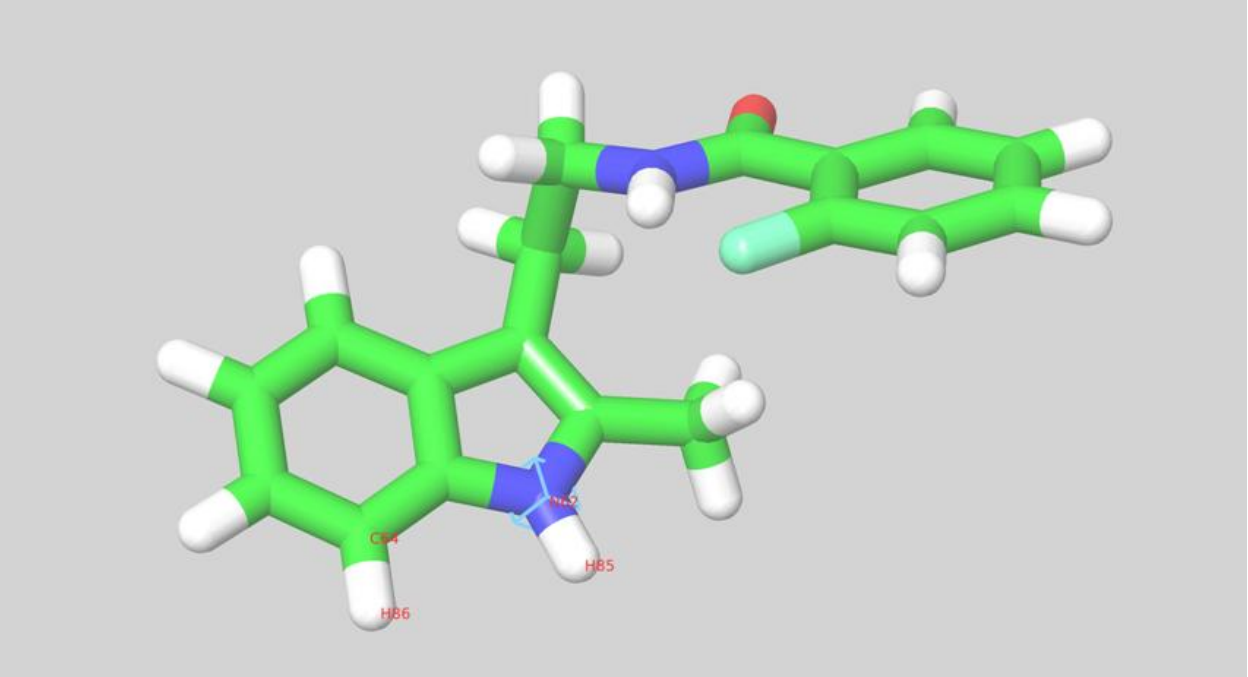


Figure 37: Reference ligand CK666.

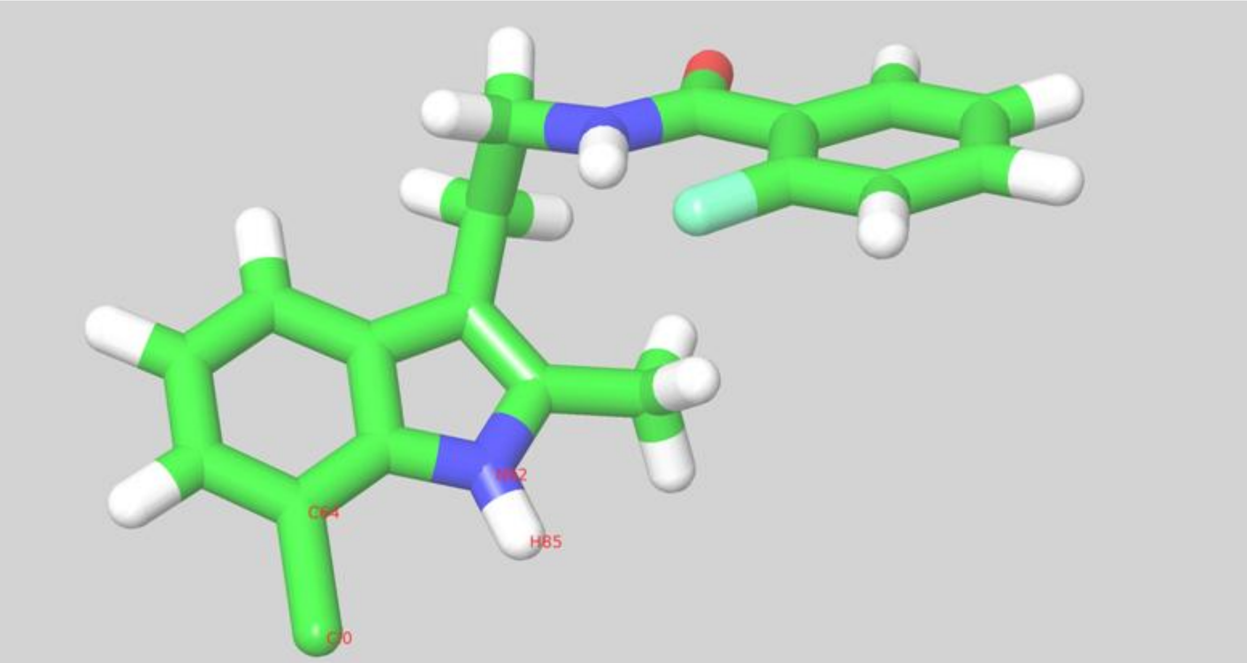


Figure 38: Mutant ligand AI007.

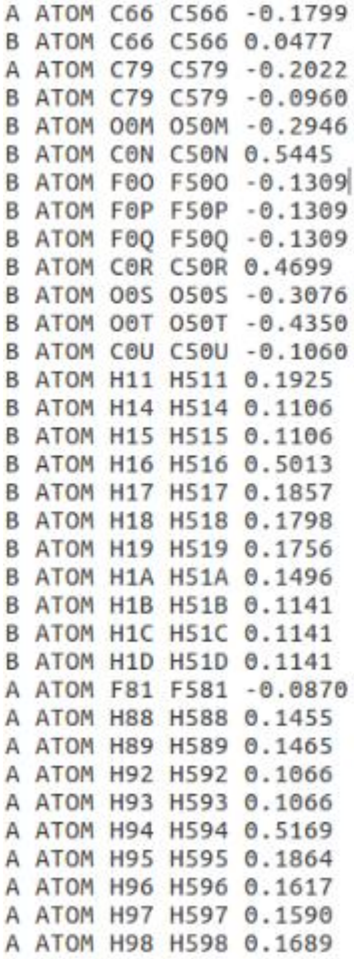
Besides the small modification in the atoms, an extra charge is introduced in the area. The algorithm runs the “dual.py” script and the code minds of the extra charge and merges the atoms of reference ligand (CK666) with the atoms that take part in the mutation of the mutant ligand (AI007), in our case ***Cl0, H85 C64*** and ***N62***. Apart from the chloride atom that takes part in the mutation, this time we have to include more atoms to our calculation. These atoms are around the area that takes part in the mutation, therefore their charges change significantly during the transformation. The manual calculations agree that the output is correct.

Now the hybrid.rtf and hybrid.pdb files have been created and the PRM file of AI007 has been updated. So the next step is to combine the protein PDB file with the hybrid.pdb file in order to create the complex.pdb. The script “complex.py” implements that correctly as well. The script “split_chains.py” generates correctly the files that VMD needs to run its calculations.

In the end it is validated that our algorithm has produced the correct output, by visualising the two ionized.pdb files that are created after “fep.py” has run. These outputs are totally compatible with NAMD since the simulation finishes correctly and provides the same results as when the simulation was run with manually created inputs.

3.3.4 Use Case, CK666-AI066

The mutant ligand AI066 is a far more complicated ligand compared to the ones described above in chapters 3.3.2 and 3.3.3. This is because different atoms are modified in different areas. Nevertheless the algorithm did not fail to produce correct results. Figure 39 shows the atoms that have been chosen to take part in the mutation after the scripts “names.py” and “dual.py” have run.



```
A ATOM C66 C566 -0.1799
B ATOM C66 C566 0.0477
A ATOM C79 C579 -0.2022
B ATOM C79 C579 -0.0960
B ATOM O0M O50M -0.2946
B ATOM C0N C50N 0.5445
B ATOM F0O F50O -0.1309
B ATOM F0P F50P -0.1309
B ATOM F0Q F50Q -0.1309
B ATOM C0R C50R 0.4699
B ATOM O0S O50S -0.3076
B ATOM O0T O50T -0.4350
B ATOM C0U C50U -0.1060
B ATOM H11 H511 0.1925
B ATOM H14 H514 0.1106
B ATOM H15 H515 0.1106
B ATOM H16 H516 0.5013
B ATOM H17 H517 0.1857
B ATOM H18 H518 0.1798
B ATOM H19 H519 0.1756
B ATOM H1A H51A 0.1496
B ATOM H1B H51B 0.1141
B ATOM H1C H51C 0.1141
B ATOM H1D H51D 0.1141
A ATOM F81 F581 -0.0870
A ATOM H88 H588 0.1455
A ATOM H89 H589 0.1465
A ATOM H92 H592 0.1066
A ATOM H93 H593 0.1066
A ATOM H94 H594 0.5169
A ATOM H95 H595 0.1864
A ATOM H96 H596 0.1617
A ATOM H97 H597 0.1590
A ATOM H98 H598 0.1689
```

Figure 39: The atoms that take part in the mutation. The atoms from the reference ligand are marked with an A, and the atoms from the mutant ligand are marked with a B.

In Figures 40 and 41 it is shown that only the atoms presented in Figure 39 are different. These are the areas that take part in the mutation. In this case, the algorithm managed to successfully include in the calculation only the atoms that participate in the mutation and only a few neighbouring atoms. The manual calculations agree that the output is correct.

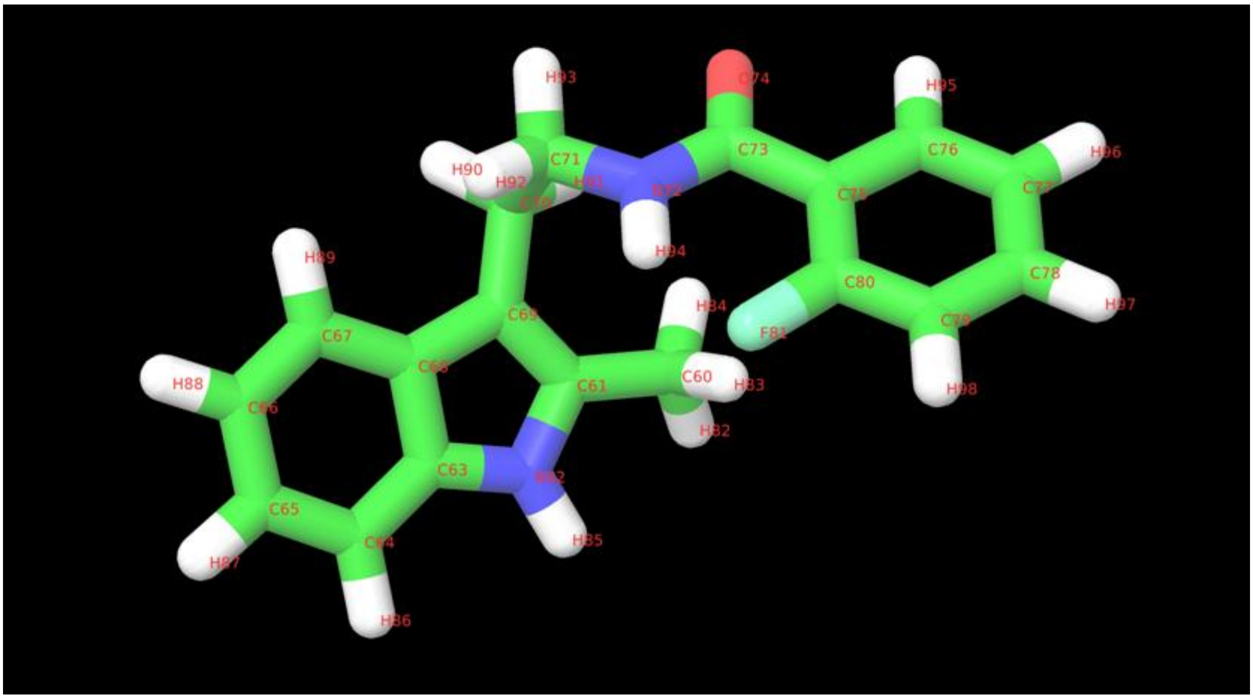


Figure 40: Reference ligand CK666.

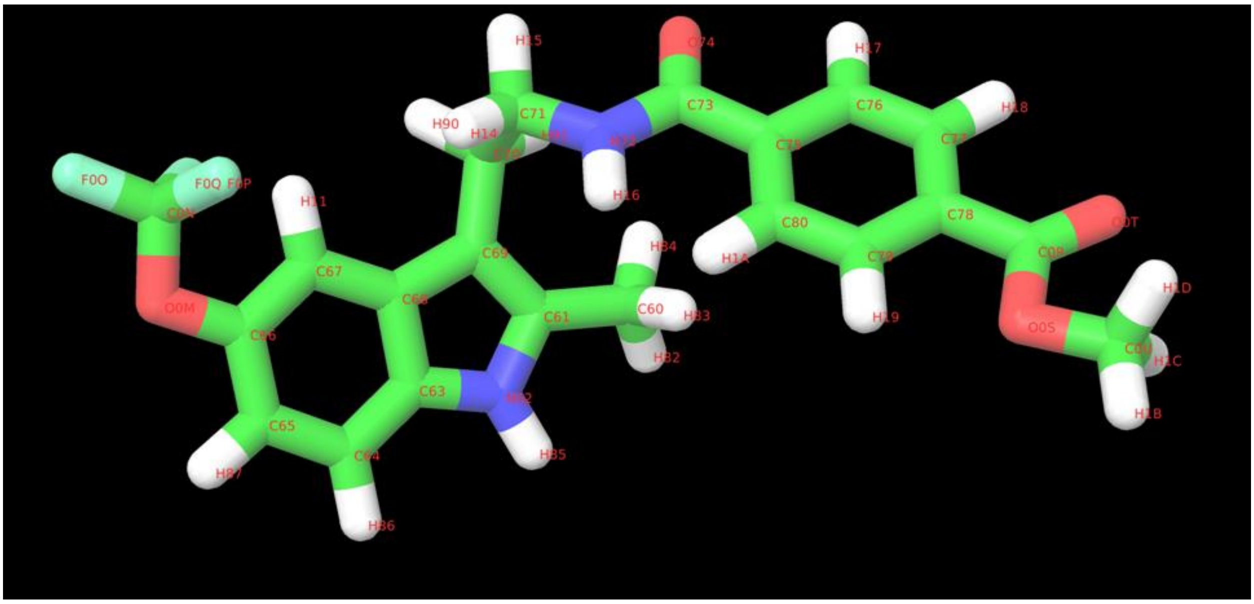


Figure 41: Mutant ligand AI066.

After the above calculations have run, the hybrid.rtf and hybrid.pdb files have been created and the PRM file of AI066 has been updated. The next step is to combine the protein with the hybrid.pdb file in order to create the complex.pdb. The script “complex.py” implements that correctly as well. The script “split_chains.py” generates correctly the files that VMD needs to run its calculations.

In the end it is validated that our algorithm has produced the correct output, by visualising the two ionized.pdb files that are created after “fep.py” has run. These outputs are totally compatible with NAMD since the simulation finishes correctly and provides the same results as when the simulation was run with manually created inputs.

4. CONCLUSIONS

The most fundamental goal in drug design is to predict whether a given molecule will bind to a target and if so how strongly. MD is most often used to estimate the strength of the intermolecular interaction between the small molecule and its biological target. These methods are also used to predict the conformation of the small molecule and to model conformational changes in the target that may occur when the small molecule binds to it [85].

FEP is not only the oldest but also one of the most useful, general purpose strategies for calculating free energy differences. Today, it is used for some of the most challenging applications, such as protein–ligand interactions and in silico protein engineering. It can also be applied to examine the effect of force fields on the computed free energies.

There are several simulation packages that perform FEP simulations, such as AMBER, GROMCS and NAMD. Although NAMD is a very well built tool, its set-up procedure has not yet been streamlined. This is what motivated us to create a web based server that automates the whole set-up procedure.

Creating all the different files needed to run a NAMD/FEP simulation is a rather tedious and time consuming process, an experienced user needed a full day to do everything manually. FEPrepare automates the set-up procedure and creates all the necessary inputs in seconds. Time waste was not the only problem. So far the Dual Topology approach introduced that the creation of the hybrid.rtf file would be done by merging the reference ligand and the mutant ligand into one. But we are interested in reducing the perturbations; this is why we came up with our own algorithm to do so. As a result it would be fair to say that FEPrepare is not only an implementation of a given workflow, but rather an important research project conducted in Cournia lab, able to perform work that other tools cannot.

The code is dynamic and it uses different functions for the different needs. This way any future implementations can be added easily and there will not be any need to rewrite parts of the code.

Having tested the algorithm with one reference ligand and 15 mutant ligands, we can be certain that it works for ligands with at least 100 atoms. Also since the code is highly dynamic we know it can work for any protein.

It is clear that FEPrepare is best and only tool one can use to set-up a NAMD/FEP simulation. The user is responsible for preparing and aligning the structures. He is also responsible for acquiring the PDB, RTF and PRM file from LigParGen. He then uploads these files to our web-server and he can download a zip file with everything he needs to run his simulation in NAMD. The website can be reached at the following address:

http://feprepare.vi-seem.eu/

5. FUTURE PERSPECTIVES

Although the work that the tool performs is totally automated, and with a single upload the user can download everything needed to run a NAMD simulation, there are several things that could fabricate FEPrepare into the ideal preparation tool not only for NAMD but for AMBER and GROMACS as well.

This tool can be the groundwork for a bigger application. First of all now our server assumes that the user has prepared the protein and aligned the ligands before uploading the structures in LigParGen. It would be ideal if FEPrepare could perform these two tasks and there was no need for using any other OPLS.

Secondly, our server handles input only from the LigParGen web-server. But not everyone is familiar with LigParGen, for this reason it would be ideal if our server could handle input of any format. Our first attempt will be to incorporate FEPrepare with CHARMM General Force Field (C-Gen-FF) format [86].

C-Gen-FF program performs atom typing and assignment of parameters and charges by analogy. Atom typing is done by a deterministic programmable decision tree. Assignment of bonded parameters is based on substituting atom types in the definition of the desired parameter. Charges are assigned using an extended bond-charge increment scheme that is able to capture short and medium-range inductive and mesomeric effects. C-Gen-FF produces one single file, a “stream file” while LigParGen produces three different outputs for each ligand.

Additionally FEPrepare should be generalized to take input for all commonly used force fields, like AMBER and GROMOS.

Now the program works only for one pair of ligands. An algorithm is needed to create a map of all the required mutations and create the input files. So instead of testing if a mutant ligand is better than the reference each time we would identify all the possible ligand pairs. Such a tool already exists. It’s LOMAP that reduces the graph of all possible ligand pairs to a minimum set based on a definition of similarity used to weight the graph's edges to solve the shortest path tree problem. As input, it takes a set of potential ligands (not only a pair), and outputs a map of planned free energy calculations spanning the set with relatively few transformations which are designed to be relatively efficient, based on the number of atomic insertions and deletions required. LOMAP also keeps the overall distance across structural clusters that are below a specified threshold. By building in closed cycles of mutations it provides consistency and information in case the calculations perform poorly. Supposing FEPrepare could support such a tool, the calculations would be even faster.

FEPrepare, uses VMD in order to create the PSF files, solvate the system, and insert counterions to electrically neutralize it. Since VMD is already used by scientists and labs all over the world, FEPrepare algorithm could be implemented as plug-in tool to VMD. The only thing that would have to change would be to rewrite the code in tcl, since it’s the only programming language VMD uses.

Last but not least, as it has already been explained FESetup is a powerful tool which handles software of different tools and combines them to prepare a simulation for AMBER and GROMACS. Since FESetup does not fully support NAMD, and our web-server does, by incorporating the two codes we could achieve having a complete set-up tool for FEP calculations.

Finally, although the tool is very fast, the algorithm could be improved to lower computational complexity and also fix bugs that might appear in the future.

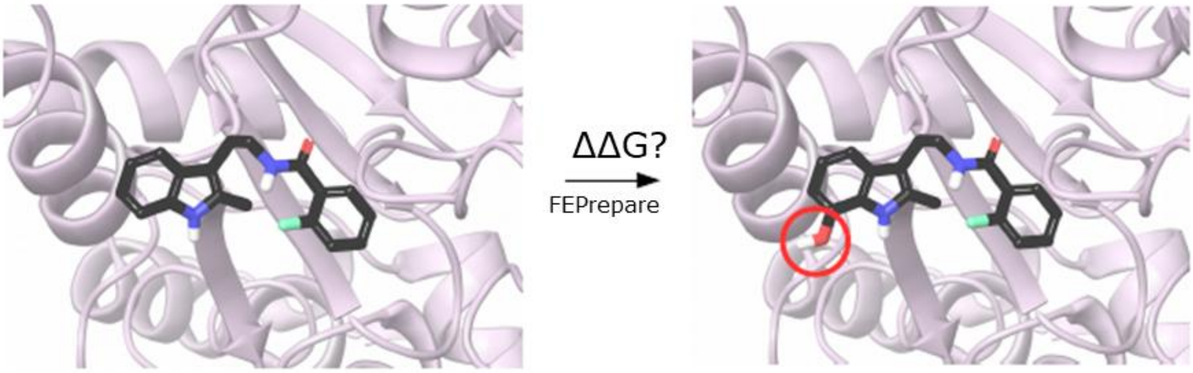
ABBREVIATIONS - ACRONYMS

BRFAA	Biomedical Research Foundation of the Academy of Athens
BAC	Binding Affinity Calculator
BAR	Bennett Acceptance Radio
CSS	Cascading Style Sheets
DIT	Department of Informatics and Telecommunications
FEP	Free Energy Perturbation
FEW	Free Energy Workflow
GB	Generalized Born
HPC	High performance computing
HTML	Hypertext Markup Language
ITMB	Information Technologies in Medicine and Biology
LIE	Linear Interaction Energies
MC	Monte Carlo
MCSS	Maximum Common Substructure Search
MD	Molecular Dynamics
MM/PBSA	Molecular Mechanics Poisson Boltzmann Surface Area
MM/GBSA	Molecular Mechanics Generalized Born Surface Area
NAMD	Nanoscale Molecular Dynamics
NHRF	National Hellenic Research Foundation
NKUA	National and Kapodistrian University of Athens
OPLS	Optimized Potentials for Liquid Simulations
PDB	Protein Data Bank file format
PES	Potential Energy Surface
PHP	Hypertext Preprocessor
PRM	Parameter file
PSF	Protein Structure File
RTF	Topology file
TCL	Tool Command Language
TI	Thermodynamic Integration

ANNEX I

The manual of the website is the following:

FEPrepare: A set-up tool for NAMD/FEP



Stamatia Zavitsanou, Alexandros Tsegenes & Zoe Cournia
Biomedical Research Foundation
Academy of Athens

http://feprepare.vi-seem.eu/

I.1. INTRODUCTION

One of the most important tasks in drug design is to predict, among a series of lead candidates, which ones will bind more strongly to the therapeutic target. In this direction, relative binding free energy methodologies have been developed, which rely on physics-based molecular simulations and rigorous statistical mechanics to calculate the differences in the free energy of binding between a parent candidate drug and analogues. For example, Free Energy Perturbation (FEP) calculations calculate the free energy difference between an initial (reference) and a final (target) molecule to an average of a function of their energy difference evaluated by sampling for the initial state [1].

Automating free energy perturbation calculations is a step forward to delivering high throughput calculations for accurate predictions of relative binding affinities before a compound is synthesized, and consequently save enormous time and cost.

NAMD [2] is a free parallel molecular dynamics code, designed for high-performance simulations of large biomolecular systems. Although FEP calculations are possible with NAMD, no automated tool has been developed to streamline the process, making the calculations tedious and unfeasible for a large number of molecules. That gave us the motivation to provide an easily accessible web based preparation tool which can produce all the files needed to run a NAMD simulation.

I.2. METHODOLOGY

In order to run a NAMD/FEP simulation, several inputs need to be created and no algorithm that does so exists. In order to create those files one has to prepare and align the structures from Maestro (or any OPLS that he prefers). Then upload these structures to LigParGen [3] in order to download the topology and the parameter files of the two ligands. Because of an inconsistency in the files that LigParGen provides, PDB, RTF, PRM, new atom names need to be given to all the atoms of both reference and mutant ligand. This is a very time consuming process, this is why the algorithm takes care of it, with a script.

The most tedious file to create, but at the same time most important, is the Dual-Topology file. In the dual-topology approach, both reference and target state atoms exist at the same time, reference state atoms disappear and target state atoms appear. In order to reduce the amount of perturbations during the transformation, we do not just merge the two ligands into one, but rather merge the two ligands into one, keeping the reference ligand the same and adding only the atoms that are being mutated from the mutant ligand. As a result the common part of the two ligands stays the same. The difficult thing is to decide which atoms are being mutated and therefore need to be merged with the atoms of the reference ligand. The algorithm takes into account the difference in the names of the atoms, as well as the difference in their partial charges, in order to figure out which of the atoms should be included in the calculation. Because of the modifications that take place around the area; the summation of the area's charge changes. In relative binding free energy calculations we cannot afford to have different charges before and after the transformation. To avoid this from happening we distribute the difference of the charges before and after the transformation equally, to all the atoms that take part in our calculation (Figure 1).

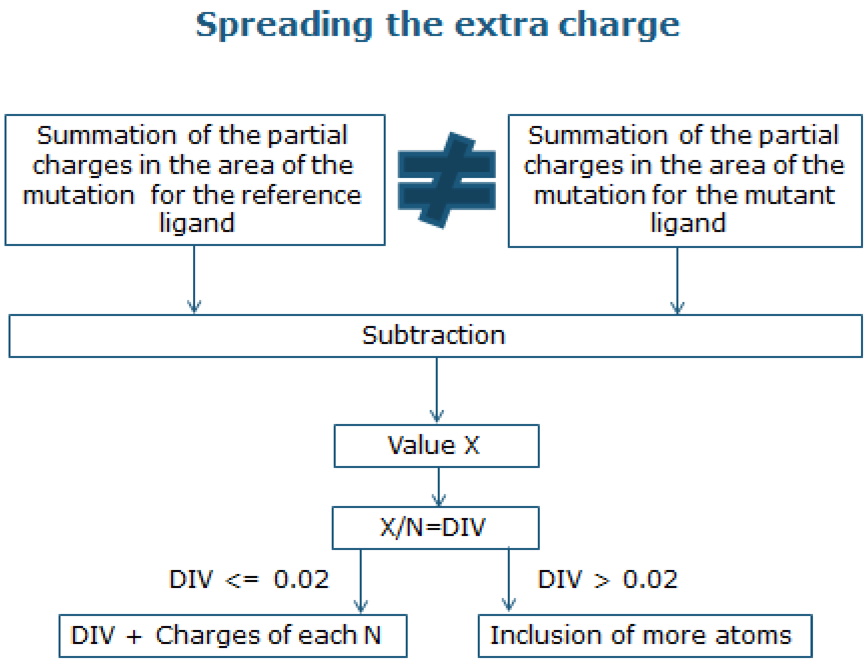

Spreading the extra charge
Summation of the partial charges in the area of the mutation for the reference ligand
≠
Summation of the partial charges in the area of the mutation for the mutant ligand
Subtraction
Value X
X/N=DIV
DIV <= 0.02
DIV > 0.02
DIV + Charges of each N
Inclusion of more atoms

Figure 42: Workflow of the algorithm that distributes the partial charges of the atoms that take part in the mutation.

After we have created the Dual-Topology file, or as some call hybrid.rtf file, we need to merge the atoms of the reference ligand with the atoms of the mutant ligand, that take part in the calculation, in the hybrid.pdb file, and update the PRM file as well. A very important file is the complex.pdb file. In order to create the complex.pdb file the algorithm merges the hybrid.pdb file with the PDB file of the protein. This file is used as an input to VMD. Since we have automated the whole procedure, there is no reason for the user to use the VMD GUI. The algorithm will do so, and generate the PSF, solvate the system in a water box with limits that are in a 10 A distance from the atom with the greatest coordinate in each direction, and insert counterions to electrically neutralize it, respectively. In addition, the script will measure the values of the minimum, the maximum and the centre of the box. The final PSF and PDB files from the preparation that VMD did are the ionized.psf file and the ionized.pdb file, respectively.

Now, we need to create a FEP file, in which we specify, which atoms disappear, which atoms stay the same, and which atoms appear, during the simulation. These are the atoms selected in the partial charge distribution step. We call these files "ionized_new.pdb" and "ionized_new.fep". The FEP file is simply a copy of *ionized.pdb* with a slight modification. We need to do the same things for the solvent as well. Of course the algorithm does that too. In the end the user can download all these files, as well as all the input files he needs to run his simulation in NAMD, such as "par_opls_aam.inp" (OPLS-AA parameters of proteins), "top_opls_aam.inp" (OPLS-AA topology of proteins), "fep.tcl" (iterative tcl script needed for the equilibration runs).

In Figure 2, one can see a workflow of the methodology we have used as described above.

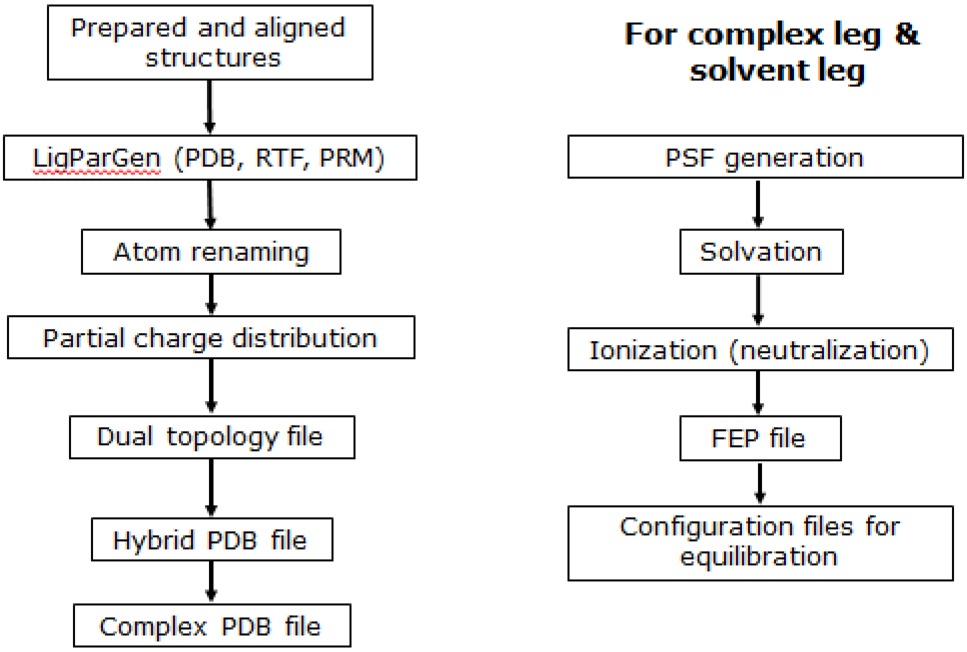

Prepared and aligned structures
LigParGen (PDB, RTF, PRM)
Atom renaming
Partial charge distribution
Dual topology file
Hybrid PDB file
Complex PDB file
For complex leg & solvent leg
PSF generation
Solvation
Ionization (neutralization)
FEP file
Configuration files for equilibration

Figure 43: FEPrepare workflow.

I.3. DESCRIPTION OF THE PROGRAM

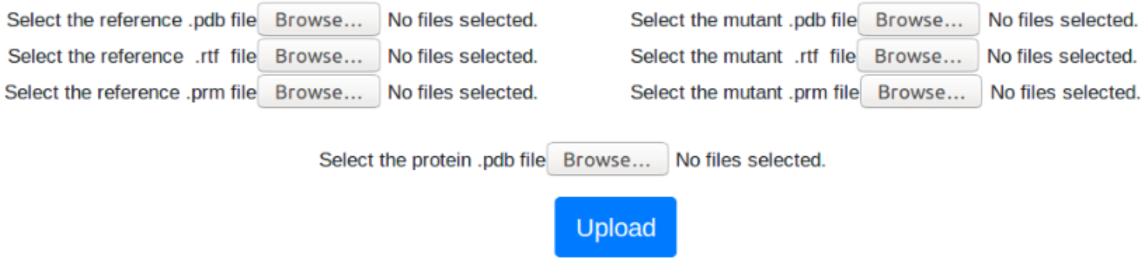
This tool creates all the files needed to run a NAMD/FEP simulation. It has been implemented as a web-server using Python and PHP and can be accessed at:

http://feprepare.vi-seem.eu/.

I.3.1 Input

The topology and parameter files

The inputs that FEPrepare needs are the coordinate, the topology and the parameter files for both ligands (reference and mutant) given as .PDB, .RTF and PRM files, and the .PDB file of the prepared and aligned protein.



Select the reference .pdb file No files selected.

Select the reference .rtf file No files selected.

Select the reference .prm file No files selected.

Select the mutant .pdb file No files selected.

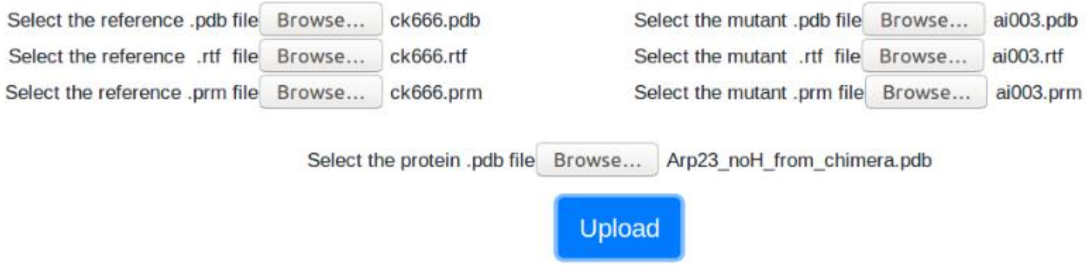
Select the mutant .rtf file No files selected.

Select the mutant .prm file No files selected.

Select the protein .pdb file No files selected.

Figure 44: File selection.

For example we have chosen CK666 as a reference ligand and AI003 as a mutant ligand. Our protein is Arp2/3.



Select the reference .pdb file ck666.pdb

Select the reference .rtf file ck666.rtf

Select the reference .prm file ck666.prm

Select the mutant .pdb file ai003.pdb

Select the mutant .rtf file ai003.rtf

Select the mutant .prm file ai003.prm

Select the protein .pdb file Arp23_noH_from_chimera.pdb

Figure 45: Files selected.

After all these required inputs are fulfilled then hit the Upload button.

All these files can be downloaded from: http://feprepare.vi-seem.eu/example. In case one needs to see how the example works.

I.3.2 Output

As a result you can download the files as a .zip file.

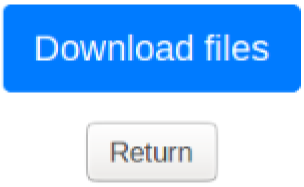


Figure 46: Files needed for NAMD/FEP simulation.

The file that the user can download as “files.zip” should contain:

- Two subfolders, “complex” and “solvent”.
- The six renamed files (PDB, RTF, PRM) for both ligands (in our case CK666.pdb, CK666.rtf, CK666.prm, AI003.pdb, AI003.rtf, AI003.prm).
- The two hybrid files named as “ligand.pdb” and “ligand.rtf”.
- The renamed PRM file after the creation of the two hybrid files, “updated.prm”.
- The “complex.pdb” file.
- The “fep.tcl” script that NAMD needs to run the simulation.
- A file with the OPLS-AA parameters of proteins (par_opls_aam.inp).
- A file with the OPLS-AA topology of proteins (top_opls_aam.inp).

Into the “complex” file the user should be able to see the following:

- The PDB files for each of the protein’s chains and the ligand (in our case “chainA.pdb” and “chainB.pdb” for the protein and “chainX.pdb” for the ligand).
- Files named “ionized.pdb”, “ionized.fep”, “ionized.psf”.
- Files named “ionized_new.pdb” and “ionized_new.fep” which contain the updated ionized files, after the “fep.py” script.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen”, and “VMD_prepare_complex_after_gui_autopsf”.
- The files that VMD gives as an output, “complex_wb.log”, “complex_wb.pdb”, “complex_wb.psf”, “psf-complex.psf”, “psf-complex.pdb”.

Into the “solvent” file the user should be able to see the following:

- Files named “ionized.pdb”, “ionized.fep”, “ionized.psf”.
- Files named “ionized_new.pdb” and “ionized_new.fep” which contain the updated ionized files, after the “fep.py” script.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen_solv”, and “VMD_prepare_ligand_after_gui_autopsf”.
- The files that VMD gives as an output, “ligand_wb.log”, “ligand_wb.pdb”, “ligand_wb.psf”, “psf-solvated.psf”, “psf-solvated.pdb”.

I.4. BIBLIOGRAPHY

[1] Athanasiou C, Vasilakaki S, Dellis D, Cournia Z. “Using Physics-based pose predictions and Free Energy Perturbation calculations to predict binding poses and relative binding affinities for FXR ligands in the D3R Grand Challenge 2” Journal of Computer-Aided Molecular Design, 2017, in press

[2] Phillips, J C, et al. “Scalable molecular dynamics with NAMD”. Journal of Computational Chemistry 2005, 26, 1781-1802.

[3] L. S. Dodda, I. Cabeza de Vaca, J. Tirado-Rives, and W. L. Jorgensen, "LigParGen web server: an automatic OPLS-AA parameter generator for organic ligands," Nucleic acids research, vol. 45, no. W1, pp. W331-W336, 2017.

REFERENCES

[1] D. S. Wishart, "Emerging applications of metabolomics in drug discovery and precision medicine," *Nature Reviews Drug Discovery,* vol. 15, no. 7, p. 473, 2016.

[2] D. Brown and G. Superti-Furga, "Rediscovering the sweet spot in drug discovery," *Drug discovery today,* vol. 8, no. 23, pp. 1067-1077, 2003.

[3] R. Macarron *et al.*, "Impact of high-throughput screening in biomedical research," *Nature reviews Drug discovery,* vol. 10, no. 3, p. 188, 2011.

[4] S. J. Y. Macalino, V. Gosu, S. Hong, and S. Choi, "Role of computer-aided drug design in modern drug discovery," *Archives of pharmacal research,* vol. 38, no. 9, pp. 1686-1701, 2015.

[5] S. Kazachenko, M. Giovinazzo, K. W. Hall, and N. M. Cann, "Algorithms for GPU-based molecular dynamics simulations of complex fluids: Applications to water, mixtures, and liquid crystals," *Journal of computational chemistry,* vol. 36, no. 24, pp. 1787-1804, 2015.

[6] D. L. Mobley and M. K. Gilson, "Predicting binding free energies: frontiers and benchmarks," *Annual review of biophysics,* vol. 46, pp. 531-558, 2017.

[7] M. De Vivo, "Bridging quantum mechanics and structure-based drug design," *optimization,* vol. 7, p. 8, 2011.

[8] J. Yin *et al.*, "Overview of the SAMPL5 host–guest challenge: Are we doing better?," *Journal of computer-aided molecular design,* vol. 31, no. 1, pp. 1-19, 2017.

[9] J. A. Lemkul, J. Huang, B. Roux, and A. D. MacKerell Jr, "An empirical polarizable force field based on the classical drude oscillator model: development history and recent applications," *Chemical reviews,* vol. 116, no. 9, pp. 4983-5013, 2016.

[10] C. Athanasiou, S. Vasilakaki, D. Dellis, and Z. Cournia, "Using physics-based pose predictions and free energy perturbation calculations to predict binding poses and relative binding affinities for FXR ligands in the D3R Grand Challenge 2," *Journal of computer-aided molecular design,* vol. 32, no. 1, pp. 21-44, 2018.

[11] A. Jakalian, B. L. Bush, D. B. Jack, and C. I. Bayly, "Fast, efficient generation of high-quality atomic Charges. AM1-BCC model: I. Method," *Journal of computational chemistry,* vol. 21, no. 2, pp. 132-146, 2000.

[12] G. Saladino and F. L. Gervasio, "Modeling the effect of pathogenic mutations on the conformational landscape of protein kinases," *Current opinion in structural biology,* vol. 37, pp. 108-114, 2016.

[13] T. J. Harpole and L. Delemotte, "Conformational landscapes of membrane proteins delineated by enhanced sampling molecular dynamics simulations," *Biochimica et Biophysica Acta (BBA)-Biomembranes,* 2017.

[14] A. K. Patidar *et al.*, "Lead Discovery and Lead Optimization: A Useful Strategy in Molecular Modification of Lead Compound in Analog Design."

[15] J. P. Hughes, S. Rees, S. B. Kalindjian, and K. L. Philpott, "Principles of early drug discovery," *British journal of pharmacology,* vol. 162, no. 6, pp. 1239-1249, 2011.

[16] A. Zhang, Y. Zhang, A. R. Branfman, R. J. Baldessarini, and J. L. Neumeyer, "Advances in development of dopaminergic aporphinoids," *Journal of medicinal chemistry,* vol. 50, no. 2, pp. 171-181, 2007.

[17] D. C. Rees, M. Congreve, C. W. Murray, and R. Carr, "Fragment-based lead discovery," *Nature Reviews Drug Discovery,* vol. 3, no. 8, p. 660, 2004.

[18] E. Harder *et al.*, "OPLS3: a force field providing broad coverage of drug-like small molecules and proteins," *Journal of chemical theory and computation,* vol. 12, no. 1, pp. 281-296, 2015.

[19] S. M. Paul *et al.*, "How to improve R&D productivity: the pharmaceutical industry's grand challenge," *Nature reviews Drug discovery,* vol. 9, no. 3, p. 203, 2010.

[20] C. Chipot and A. Pohorille, *Free energy calculations*. Springer, 2007.

[21] J. G. Kirkwood, "Statistical mechanics of fluid mixtures," *The Journal of Chemical Physics,* vol. 3, no. 5, pp. 300-313, 1935.

[22] A. Moll, A. Hildebrandt, H.-P. Lenhof, and O. Kohlbacher, "BALLView: an object-oriented molecular visualization and modeling framework," *Journal of computer-aided molecular design,* vol. 19, no. 11, pp. 791-800, 2005.

[23] N. Homeyer and H. Gohlke, "Free energy calculations by the molecular mechanics Poisson−Boltzmann surface area method," *Molecular Informatics,* vol. 31, no. 2, pp. 114-122, 2012.

[24] J. Wang, T. Hou, and X. Xu, "Recent advances in free energy calculations with a combination of molecular mechanics and continuum models," *Current Computer-Aided Drug Design,* vol. 2, no. 3, pp. 287-306, 2006.

[25] B. Honig and A. Nicholls, "Classical electrostatics in biology and chemistry," *Science,* vol. 268, no. 5214, pp. 1144-1149, 1995.

[26] M. K. Gilson, K. A. Sharp, and B. H. Honig, "Calculating the electrostatic potential of molecules in solution: method and error assessment," *Journal of computational chemistry,* vol. 9, no. 4, pp. 327-335, 1988.

[27] H. Gohlke, C. Kiel, and D. A. Case, "Insights into protein–protein binding by binding free energy calculation and free energy decomposition for the Ras–Raf and Ras–RalGDS complexes," *Journal of molecular biology,* vol. 330, no. 4, pp. 891-913, 2003.

[28] H. Gohlke and D. A. Case, "Converging free energy estimates: MM-PB (GB) SA studies on the protein–protein complex Ras–Raf," *Journal of computational chemistry,* vol. 25, no. 2, pp. 238-250, 2004.

[29] C. Tan, Y.-H. Tan, and R. Luo, "Implicit nonpolar solvent models," *The Journal of Physical Chemistry B,* vol. 111, no. 42, pp. 12263-12274, 2007.

[30] J. Srinivasan, T. E. Cheatham, P. Cieplak, P. A. Kollman, and D. A. Case, "Continuum solvent studies of the stability of DNA, RNA, and phosphoramidate− DNA helices," *Journal of the American Chemical Society,* vol. 120, no. 37, pp. 9401-9409, 1998.

[31] M. Karplus and J. N. Kushick, "Method for estimating the configurational entropy of macromolecules," *Macromolecules,* vol. 14, no. 2, pp. 325-332, 1981.

[32] H. Gutiérrez-de-Terán and J. Åqvist, "Linear interaction energy: method and applications in drug design," in *Computational drug discovery and design*: Springer, 2012, pp. 305-323.

[33] D. A. Case *et al.*, "The Amber biomolecular simulation programs," *Journal of computational chemistry,* vol. 26, no. 16, pp. 1668-1688, 2005.

[34] K. Lindorff-Larsen *et al.*, "Improved side-chain torsion potentials for the Amber ff99SB protein force field," *Proteins: Structure, Function, and Bioinformatics,* vol. 78, no. 8, pp. 1950-1958, 2010.

[35] V. Gapsys, S. Michielssens, D. Seeliger, and B. L. de Groot, "pmx: Automated protein structure and topology generation for alchemical perturbations," *Journal of computational chemistry,* vol. 36, no. 5, pp. 348-354, 2015.

[36] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, "GROMACS: fast, flexible, and free," *Journal of computational chemistry,* vol. 26, no. 16, pp. 1701-1718, 2005.

[37] M. Lundborg and E. Lindahl, "Automatic GROMACS topology generation and comparisons of force fields for solvation free energy calculations," *The Journal of Physical Chemistry B,* vol. 119, no. 3, pp. 810-823, 2014.

[38] S. Liu *et al.*, "Lead optimization mapper: automating free energy calculations for lead optimization," *Journal of computer-aided molecular design,* vol. 27, no. 9, pp. 755-770, 2013.

[39] S. K. Sadiq, D. Wright, S. J. Watson, S. J. Zasada, I. Stoica, and P. V. Coveney, "Automated molecular simulation based binding affinity calculator for ligand-bound HIV-1 proteases," ed: ACS Publications, 2008.

[40] H. H. Loeffler, J. Michel, and C. Woods, "FESetup: automating setup for alchemical free energy simulations," ed: ACS Publications, 2015.

[41] J. C. Phillips *et al.*, "Scalable molecular dynamics with NAMD," *Journal of computational chemistry,* vol. 26, no. 16, pp. 1781-1802, 2005.

[42] J. W. Raymond and P. Willett, "Maximum common subgraph isomorphism algorithms for the matching of chemical structures," *Journal of computer-aided molecular design,* vol. 16, no. 7, pp. 521-533, 2002.

[43] A. Dalke and J. Hastings, "FMCS: a novel algorithm for the multiple MCS problem," *Journal of cheminformatics,* vol. 5, no. S1, p. O6, 2013.

[44] A. R. Leach, *Molecular modelling: principles and applications*. Pearson education, 2001.

[45] A. Hinchliffe, *Molecular modelling for beginners*. John Wiley & Sons, 2005.

[46] P. Atkins and J. De Paula, "Atkins' physical chemistry," *New York,* p. 77, 2006.

[47] S. Paschalis and G. Anagnostatos, "Ground State of 4-7H Considering Internal Collective Rotation," *Journal of Modern Physics,* vol. 4, no. 05, p. 66, 2013.

[48] E. Schrödinger, "An undulatory theory of the mechanics of atoms and molecules," *Physical review,* vol. 28, no. 6, p. 1049, 1926.

[49] M. Born and R. Oppenheimer, "Zur quantentheorie der molekeln," *Annalen der physik,* vol. 389, no. 20, pp. 457-484, 1927.

[50] M. Born and R. Oppenheimer, "On the quantum theory of molecules," in *Quantum Chemistry: Classic Scientific Papers*: World Scientific, 2000, pp. 1-24.

[51] D. C. Young, *Computational drug design: a guide for computational and medicinal chemists*. John Wiley & Sons, 2009.

[52] C. L. Verlinde and W. G. Hol, "Structure-based drug design: progress, results and challenges," *Structure,* vol. 2, no. 7, pp. 577-587, 1994.

[53] H. Gohlke and G. Klebe, "Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors," *Angewandte Chemie International Edition,* vol. 41, no. 15, pp. 2644-2676, 2002.

[54] R. W. Zwanzig, "High-temperature equation of state by a perturbation method. I. nonpolar gases," *The Journal of Chemical Physics,* vol. 22, no. 8, pp. 1420-1426, 1954.

[55] L. Wang, B. Berne, and R. A. Friesner, "On achieving high accuracy and reliability in the calculation of relative protein–ligand binding affinities," *Proceedings of the National Academy of Sciences,* vol. 109, no. 6, pp. 1937-1942, 2012.

[56] K. Vanommeslaeghe *et al.*, "CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields," *Journal of computational chemistry,* vol. 31, no. 4, pp. 671-690, 2010.

[57] S. Boresch and M. Karplus, "The role of bonded terms in free energy simulations: 1. Theoretical analysis," *The Journal of Physical Chemistry A,* vol. 103, no. 1, pp. 103-118, 1999.

[58] J. Michel and J. W. Essex, "Prediction of protein–ligand binding affinity by free energy simulations: assumptions, pitfalls and expectations," *Journal of computer-aided molecular design,* vol. 24, no. 8, pp. 639-658, 2010.

[59] J. C. Gumbart, B. Roux, and C. Chipot, "Standard binding free energies from computer simulations: What is the best strategy?," *Journal of chemical theory and computation,* vol. 9, no. 1, pp. 794-802, 2012.

[60] M. R. Shirts and D. L. Mobley, "An introduction to best practices in free energy calculations," in *Biomolecular Simulations*: Springer, 2013, pp. 271-311.

[61] R. K. Brunner and L. V. Kalé, "Handling application-induced load imbalance using parallel objects," *Parallel and Distributed Computing for Symbolic and Irregular Applications,* pp. 167-181, 2000.

[62] M. Nelson *et al.*, "MDScope—A visual computing environment for structural biology," *Computer physics communications,* vol. 91, no. 1-3, pp. 111-133, 1995.

[63] M. T. Nelson *et al.*, "NAMD: a parallel, object-oriented molecular dynamics program," *The International Journal of Supercomputer Applications and High Performance Computing,* vol. 10, no. 4, pp. 251-268, 1996.

[64] D. Kosztin, T. C. Bishop, and K. Schulten, "Binding of the estrogen receptor to DNA. The role of waters," *Biophysical journal,* vol. 73, no. 2, pp. 557-570, 1997.

[65] E. Villa, A. Balaeff, and K. Schulten, "Structural dynamics of the lac repressor–DNA complex revealed by a multiscale simulation," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 102, no. 19, pp. 6783-6788, 2005.

[66] R. Kubo, M. Toda, and N. Hashitsume, *Statistical physics II: nonequilibrium statistical mechanics*. Springer Science & Business Media, 2012.

[67] W. Humphrey, A. Dalke, and K. Schulten, "VMD: visual molecular dynamics," *Journal of molecular graphics,* vol. 14, no. 1, pp. 33-38, 1996.

[68] R. Salomon-Ferrer, D. A. Case, and R. C. Walker, "An overview of the Amber biomolecular simulation package," *Wiley Interdisciplinary Reviews: Computational Molecular Science,* vol. 3, no. 2, pp. 198-210, 2013.

[69] J. I. Intermolecular, "Surface Forces Academic Press," *San Diego,* 1992.

[70] C. N. Schutz and A. Warshel, "What are the dielectric “constants” of proteins and how to validate electrostatic models?," *Proteins: Structure, Function, and Bioinformatics,* vol. 44, no. 4, pp. 400-417, 2001.

[71] L. S. Dodda, I. Cabeza de Vaca, J. Tirado-Rives, and W. L. Jorgensen, "LigParGen web server: an automatic OPLS-AA parameter generator for organic ligands," *Nucleic acids research,* vol. 45, no. W1, pp. W331-W336, 2017.

[72] G. Schlossnagle, *Advanced PHP programming: a practical guide to developing large-scale web sites and applications with PHP 5*. Pearson Education, 2004.

[73] R. Sacks-Davis, T. Arnold-Moore, and J. Zobel, "Database systems for structured documents," *IEICE TRANSACTIONS on Information and Systems,* vol. 78, no. 11, pp. 1335-1342, 1995.

[74] H. W. Lie and B. Bos, *Cascading style sheets: Designing for the web*. Addison-Wesley Professional, 2005.

[75] S. Maestro, "Version 9.2," *LLC, New York,* 2011.

[76] G. M. Sastry, M. Adzhigirey, T. Day, R. Annabhimoju, and W. Sherman, "Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments," *Journal of computer-aided molecular design,* vol. 27, no. 3, pp. 221-234, 2013.

[77] T. Pantsar *et al.*, "Design, synthesis, and biological evaluation of 2, 4-dihydropyrano [2, 3-c] pyrazole derivatives as autotaxin inhibitors," *European Journal of Pharmaceutical Sciences,* vol. 107, pp. 97-111, 2017.

[78] B. Nolen *et al.*, "Characterization of two classes of small molecule inhibitors of Arp2/3 complex," *Nature,* vol. 460, no. 7258, p. 1031, 2009.

[79] Y. Nakamura *et al.*, "PICK1 inhibition of the Arp2/3 complex controls dendritic spine size and synaptic plasticity," *The EMBO journal,* vol. 30, no. 4, pp. 719-730, 2011.

[80] M. Spillane *et al.*, "The actin nucleating Arp2/3 complex contributes to the formation of axonal filopodia and branches through the regulation of actin patch precursors to filopodia," *Developmental neurobiology,* vol. 71, no. 9, pp. 747-758, 2011.

[81] S. N. Duleh and M. D. Welch, "WASH and the Arp2/3 complex regulate endosome shape and trafficking," *Cytoskeleton,* vol. 67, no. 3, pp. 193-206, 2010.

[82] D. V. Ilatovskaya, T. S. Pavlov, V. Levchenko, Y. A. Negulyaev, and A. Staruschenko, "Cortical actin binding protein cortactin mediates ENaC activity via Arp2/3 complex," *The FASEB Journal,* vol. 25, no. 8, pp. 2688-2699, 2011.

[83] L. M. Machesky, S. J. Atkinson, C. Ampe, J. Vandekerckhove, and T. D. Pollard, "Purification of a cortical complex containing two unconventional actins from Acanthamoeba by affinity chromatography on profilin-agarose," *The Journal of cell biology,* vol. 127, no. 1, pp. 107-115, 1994.

[84] R. C. Robinson *et al.*, "Crystal structure of Arp2/3 complex," *Science,* vol. 294, no. 5547, pp. 1679-1684, 2001.

[85] R. A. Lewis, "The development of molecular modelling programs: the use and limitations of physical models," in *Drug Design Strategies*, 2011, pp. 88-107.

[86] K. Vanommeslaeghe and A. D. MacKerell Jr, "Automation of the CHARMM General Force Field (CGenFF) I: bond perception and atom typing," *Journal of chemical information and modeling,* vol. 52, no. 12, pp. 3144-3154, 2012.