University of Southampton
School of Chemistry

Doctor of Philosophie Thesis

Modelling Protein Backbone Loops Using the Monte Carlo Method

Author:
Juan Fernandez Carmona

Supervisor:
Dr. Jonathan Essex

April 3, 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

SCHOOL OF CHEMISTRY

Doctor of Philosophy

MODELLING PROTEIN BACKBONE LOOP USING THE MONTE CARLO METHOD

by Juan Fernandez Carmona

Novel methods that perform local moves such as the gaussian bias or Concerted Rotation with Angles, increase the exploration of the conformational phase space. These methods have been applied successfully to small systems, and have proved to be more efficient than the classical Monte Carlo method.

The main aim of my work was to study and include backbone moves for proteins, such as the Concerted Rotation with Angle (CRA) and the gaussian bias in the ProtoMS package. The CRA was then applied to several systems of biological interest to compute relative binding free energies and conformational changes to obtain insights into the binding mode and system flexibility.

The CRA algorithm has been used to sample biological systems such as lysozyme L99A mutant, Bcr-Abl kinases and PDE5 phosphodiesterase and led to increased sampling of the backbone and more precise free energy results.

Acknowledgements

I would like to acknowledge my supervisor the Pr Jonanthan Essex for his trust and support during my PhD. I would also like to thank Pzifer and my industrial supervisor Willen Van Hoorn for funding my PhD.

This is probably the most difficult chapter of the thesis to write, as this chapter is the only chapter that everyone is going to read. Sadly the long list of name to follow is going to render this chapter as narcoleptic as the remainder of the thesis.

The Pr George Attard for being my advisor and also for teaching me that Pr or post-grad, gravity strikes in the same hard and predictable way deserves my gratitude. I can not emphasise enough the impact the doctors Christopher Woods and Julien Michel had on my work.

As I speak of computational tools I think it is only fair that to thank the developers of the numerous open source (and free) softwares I have been using during my PhD(in no particular order: LaTeX, VMD, Linux, GCC, glibc, g77, Perl (Huge thank!), Python, KDE, LATEX, PyMol, Xm-Grace, and many more without which things would have been more difficult).

I am in the most debt to my friends. The ones I have made here in Southampton and the old ones from France for their support. They have made the dark moments brighter. I specially want to thank Tanya without whom the outcome of the PhD may have been different.

To open the off-science ”thanks party” I would like to start with former group members Sarah and Esther, and the C CUBED (copyright Pr Attard) for getting me back into climbing and for coping with my lame jokes (and I have to admit that some of them were very lame).

All the members of the Southampton university Judo Club deserve my gratitude, particularly Nicola, James, Becky, Owen and Alun for the post training pub sessions, the movies, the DVD nights and the wedding where I get publicly humiliated playing crocket (I naively though the games was an invention from Lewis Carroll), but specially Gary for introducing me to single-speeding (the most British way of riding a bike), letting me use his workshop and organising the trip to Scotland so I could ride against the finest single-speeders in the world.

Speaking of bikes I would like to acknowledge my fellow mountain bikers who's sheer stupidity and lack of basic scientific understanding inspired me to work harder to increase the knowledge in the world. I live in the hope that by raising the overall average level of scientific understanding my fellow riders will be dragged above the level of ignorance that causes them to suffer near death experiences on a regular basis specially as they persist to build very hazardous riding material made of fallen bits of trees. Special mention for the mid-life crisis squad (frasher, phil and richard) and Tim for the numerous night rides that kept me sane (and skinny) through the winter, the tunnel hill trolls (Peter, Kirsty, Elaine, Jo and Dezb) and Nick whose sheer stupidity in crafting the mentioned riding material almost got me killed.

I cannot end this chapter without the deepest thoughts to my mother. She has always been supportive of all my choice even the ones she though were unwise (like purchasing a motorbike) and alway tried to help from the best she could. So for everything thanks mum.

Contents

Chapter 1

Introduction

1.1 Aims

The aim of this work is to implement novel methods to increase the sampling of the backbone of proteins using stochastic simulations. For these methods to be become widely used, they must be able to enhance the sampling of the protein backbone, to increase accuracy of relative binding free energy computations where the sampling is a limiting factor, and to perform reliably under the constraints of the pharmaceutical industry. The methods should be fast and require as little user intervention as possible.

1.2 Drug design

From a chemical point of view, the design of active substrates for a given protein is a difficult and expensive process. For a drug to be efficient and have little or no side effects it has to be very selective to its target. To test the efficiency and selectivity of a molecule towards a given biological target, one option is to do expensive experimental screening *via* automatic testing. Such testing makes the production of a drug very cost ineffective.

Long gone are the day of experimental automatic testing (although some virtual screening for lead optimisation still exists). Nowadays, our understanding of the mechanisms involved in the binding process is widely used to lead to the design of new drugs. The use of experimental crystal or NMR structures and modelling methods allows the drug design process to be more efficient. This process is called *rational drug design*[1]. Rational drug design aims to gather knowledge of the structure of the protein and existing ligand(s), and to study the interaction between these to lead to the design of new compounds. Having efficient and cheap computational methods should make the trial and error test obsolete and decrease the cost of bringing a new drug to the market.

Since the 1990s, computational chemistry has emerged as a technique of choice to investigate both protein folding and protein behaviour in vacuum or solvent[2]. Development of methods such as molecular modelling, scoring functions or free energy perturbation and a rise in hardware developments* have resulted in a major breakthrough in rational drug design[4–6].

The use of molecular modelling to investigate protein behaviour in solvent has become more and more reliable and faster as computing costs have been reduced. Nowadays, methods such as free energy perturbation can be applied to more systems to investigate protein flexibility or ligand selectivity. There is still some space for interesting challenges in the field of computational chemistry such as folding of proteins into their native structure or sampling the activation pathway leading to domain motions, since very little information is available from an experimental point of view.

Insights into the conformational changes related to the binding mechanism, would provide the knowledge to design selective compounds and reduce the cost of new drugs.

*Moore stated in 1965 that the number of processor would double every year[3]. CPU speed in 1990 was 25 MHz with a 30 MHz CPU project from Intel, whereas now most of the desktop have now 3.2 GHz dual or even quad core processors inside. The memory and storage capabilities of computers has increased by a factor of thousand (having several GBytes of RAM is nowadays common even on the cheapest desktop machines).

1.3 Rigorous methods for rational drug design

The primary techniques used to calculate the physical properties of models of proteins are molecular dynamics simulations (MD) and Monte Carlo (MC) methods which have become more accurate in recent years. Considering the wealth of other related methods to enhance sampling of protein that have been recently employed (including minimisation techniques[7], conformational space annealing[8,9], multi-canonical simulation[10,11], and more recently replica exchange methods[12], digital filtering[13,14], and ensemble dynamics[15,16]) the use of molecular modelling is now able to provide most of the information on a chosen system.

However, in spite of hardware and method developments, studying complete folding, or sampling large-scale activation pathways using traditional molecular mechanics methods such as molecular dynamics or Monte Carlo has, until recently, been beyond the computational possibilities for any but the smallest systems.

MD simulations use force fields and a time step to numerically integrate Newton's laws. They aim to explore the phase space by building up a time/conformation relationship. The ability to use explicit solvation and to obtain dynamic properties of a system is one of the advantages of MD. One of the main weaknesses is that a system can become trapped in a local energy minimum (in a computational accessible timescale), limiting exploration of the potential energy surface and leading to convergence problems. So far, average MD studies are no longer than approximately 100 ns, whereas most of the conformational biological processes such as folding, occur in the range of microsecond or millisecond.

MC simulations aim, on the other hand, to generate a trajectory through phase space which samples from a statistical ensemble. The step $n+1$ is chosen by randomly moving one or several atoms or degrees of freedom (dof). The energy of the new configuration has to satisfy the Metropolis criterion[17] in order to be accepted as a new configuration (see section 3.3). Through a judicious choice of moves, this method allows some energy barriers to be stepped over. However, the random generation of a new protein backbone conformation often leads to side chain clashes resulting in a high energy state and a rejected move. This problem has been addressed using specific methods and algorithms for the sampling of proteins by MC methods.

In theory, both MD and MC should lead to the same results, despite the fact they work in different ways. The time averaged properties (for MD), or the ensemble average properties (for MC) should be identical for the same system, provided the simulations are run for long enough. There are not absolute rules to decide which method to use. Systems, models, force fields and the properties to be measured lead to the choice of one method rather than the other. In MD, it is very difficult to explore all the potential energy surface, particularly when two states with similar potential energy are separated with a high energy barrier. Such phase space sampling problems are less likely to occur with the MC method, as ”jumps” over energy barriers are possible.

1.4 Concluding remarks

Modelling methods are able to give insights of protein conformational changes non accessible using experimental techniques. Such conformational changes are however very difficult to model using traditional molecular modelling. MD methods can be trapped in local energy minima. MC methods are able to jump over energy barriers, but sampling large backbone moves is difficult. For both MD and MC advanced sampling methods have been developed to address these issues.

The next chapter will give brief information on a protein structure, how do they fold and what are the mechanisms responsible for that. Then theory beyond the MC and MD methods will be describe. The last part of the background overview will be a review of the existing specific algorithms for the Monte Carlo method with a focus on the methods used during my PhD.

Chapter 2

Protein structure

A protein is a complex macromolecule, composed of polymeric amino-acid chains[18]. The three-dimensional structure of a protein is the consequence of several factors and interactions described below.

2.1 Amino-acids and protein structure

In biochemistry, amino acids refer to the general formula $H_2NCHRCOOH$, where R is an organic substituent (see figure 2.1). In the α-amino acids,

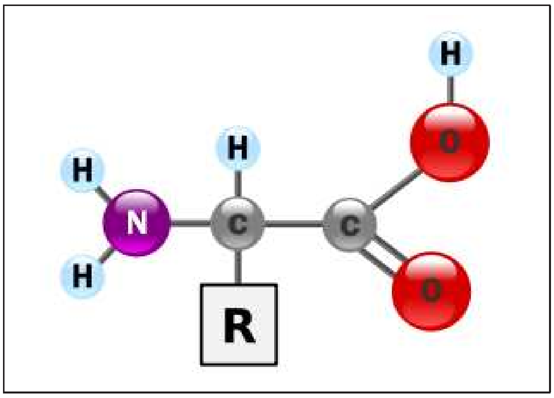


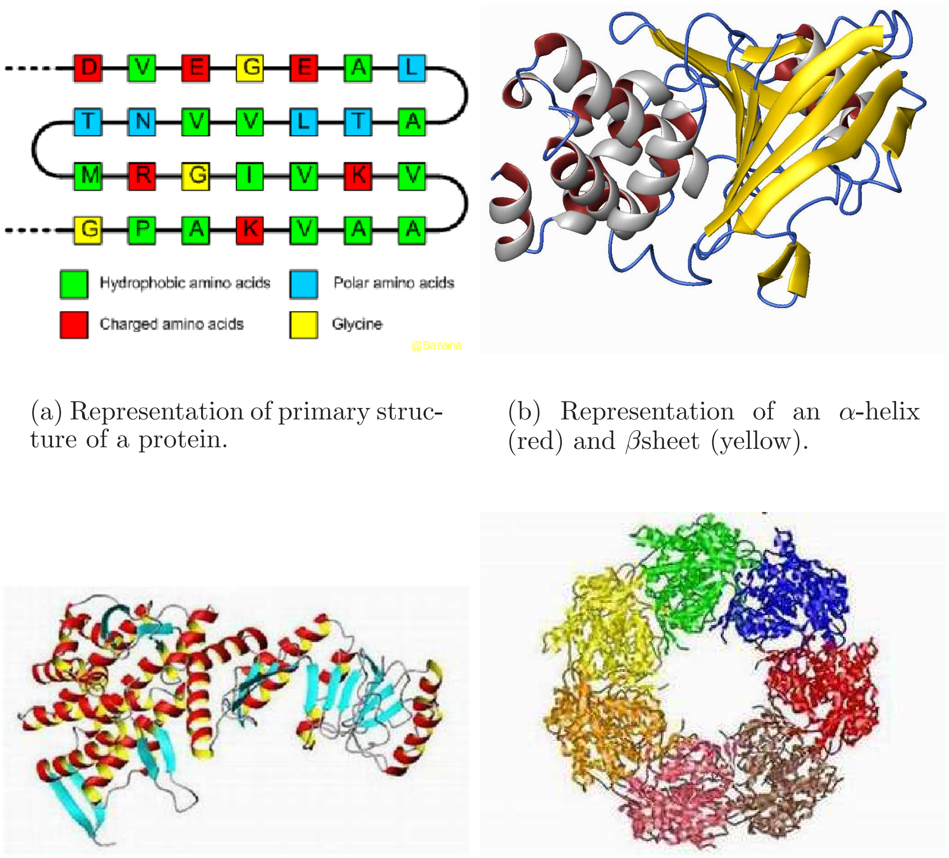
Figure 2.1: Representation of an amino-acid[18]. **R** represents the side chain of the amino acid.

the amino and carboxylate groups are attached to the same carbon, which is called the α carbon, the substituent R is referred to as the side chain. The various α amino acids differ in which side chain (R group) is attached to their α carbon. They can vary in size from just a hydrogen atom in glycine through a methyl group in alanine to a large heterocyclic group in tryptophan (see table 2.1 for the list of common amino acids). In a protein, the amide bond is referred as the peptide bond. In a peptide bond, the C, O, N, H atoms are in the same plane (thus forming a dihedral angle of 180 degree, or 0 degree for the proline 0).

Amino Acid	3-Letter	1-Letter	Polarity	Acidity or basicity
Alanine	Ala	A	non-polar	neutral
Arginine	Arg	R	polar	basic (strongly)
Asparagine	Asn	N	polar	neutral
Aspartic acid	Asp	D	polar	acidic
Cysteine	Cys	C	polar	neutral
Glutamic acid	Glu	E	polar	acidic
Glutamine	Gln	Q	polar	neutral
Glycine	Gly	G	non-polar	neutral
Histidine	His	H	polar	basic (weakly)
Isoleucine	Ile	I	non-polar	neutral
Leucine	Leu	L	non-polar	neutral
Lysine	Lys	K	polar	basic
Methionine	Met	M	non-polar	neutral
Phenylalanine	Phe	F	non-polar	neutral
Proline	Pro	P	non-polar	neutral
Serine	Ser	S	polar	neutral
Threonine	Thr	T	polar	neutral
Tryptophan	Trp	W	non-polar	neutral
Tyrosine	Tyr	Y	polar	neutral
Valine	Val	V	non-polar	neutral

Table 2.1: Amino acid nomenclature

Amino acids can be combined to form the structure of many different proteins in the same fashion letters can be combined to form many different words. This combination is known as the primary structure of the protein 2.2(a). Protein are not linear macro molecules and due to internal forces,



(a) Representation of primary structure of a protein.

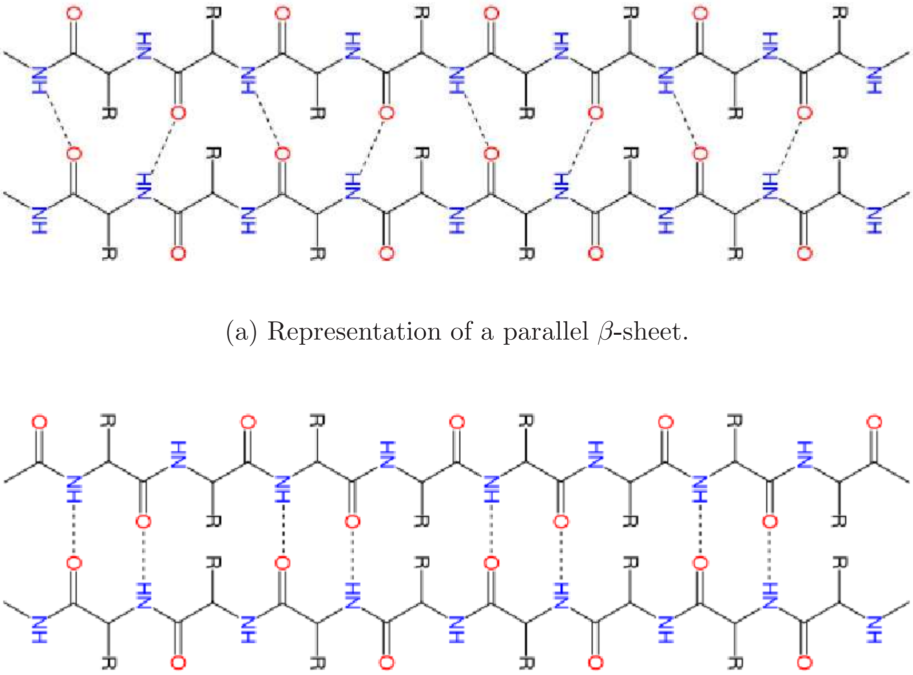
(b) Representation of an α-helix (red) and βsheet (yellow).

(c) Representation of tertiary structure.

(d) Representation of a quaternary structure.

Figure 2.2: Representation of protein structures.[19]

they adopt folded conformations. These conformations are different for each protein, and referred as secondary 2.2(b) and tertiary 2.2(c) structures. The secondary structure is partly the consequence of the H-bonding interactions between the oxygen of the carboxyl group of one amino acid and the hydrogen of the amide functions of another. The principal folds for secondary structure are the α-helix, and β-sheet. In the α-helix, the amino-acids roll in an anticlockwise direction and the side chains are on the outside of the helix. In the fully extended β strand, successive side chains point straight up, then straight down, then straight up, etc. In parallel β-sheet, sides chains point toward the same direction, whereas in anti parallel β-sheet, side chains point in opposite direction (see figure 2.1). However, other extended structures such



(a) Representation of a parallel β-sheet.

(b) Representation of an antiparrallel β-sheet.

Figure 2.3: Representation of antiparallel and parallel β-sheet.

as the polyproline helix and alpha sheet are rare in native state proteins but are often hypothesised as important protein folding intermediates[18]. Other types of helices exist such as 3_{10}-helix or the π-helix[20–22]. Tight turns and lose, flexible loops link the more "regular" secondary structure elements. The random coil is not a true secondary structure, but is the class that indicates an absence of regular secondary structure. The overall 3D structure of the polypeptide chain is referred to as the protein tertiary structure. The tertiary structure of a protein describes the way the secondary structure folds into a more compact conformation using a variety of turns and shapes (figure 2.2(c)). Tertiary structure is stabilised by H-bonding, ionic effects, non polar interactions, or sometimes by disulphide bridges. For some proteins with an important number of residues, peculiar reorganisation can occur: several motifs pack together to form compact, local, semi-independent units called domains. A structural domain is an element of the protein's overall structure that is self-stabilising and often folds independently of the rest of the protein chain. Each domain contains an individual hydrophobic core built from secondary structure units connected by loop regions.

Many proteins are actually assemblies of more than one polypeptide chain, which in the context of the larger assembly are known as protein subunits. The quaternary protein structure involves the clustering of several subunits into a final specific shape(figure 2.2(d)). There are two major categories of proteins with quaternary structure - fibrous and globular.

2.2 Protein flexibility

The understanding of protein 3D structure is one of the most important keys in the synthesis of inhibitors and medical drugs (for more details on protein structures see reference[18]). Proteins are not fixed structures and due to internal and external forces, their shape changes by contracting or relaxing with time (often called protein breathing). The lock and key model (see figure 2.4) for a protein-ligand interaction is now known to be incomplete due to the protein dynamics[23].

Being able to investigate structure-function relationships and obtain insights of protein behaviour is a key of modern computational chemistry, and could lead to major breakthrough in understanding binding processes. As a protein breathes, internal degrees of freedom change, and binding features evolve. Getting information on how these features change and how the ligand binding mode evolves can lead to better drug design and an increase in the efficiency of a drug. There are several possible fluctuations for proteins. The simplest is side chain motion, where internal degrees of freedom along the side chain move according to internal or external forces. For example, a protein bound to different ligand with different rotamers to accommodate the change of volume[25].

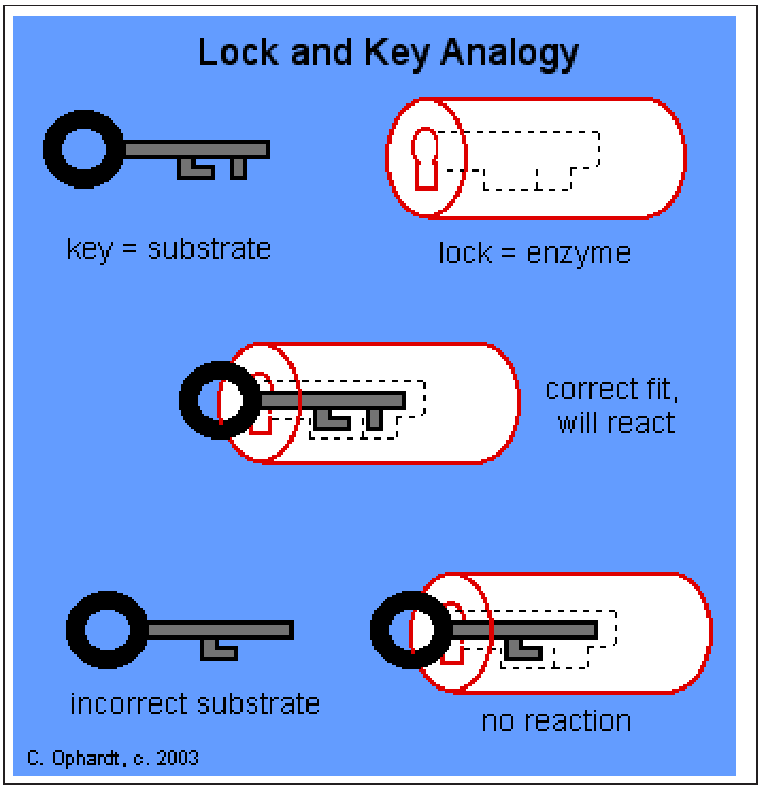


Figure 2.4: Lock and key model for protein[24].

Then backbone motions are involved. Such moves can be simple changes in the Ramachadran angles[26] or bond angles to make a section of the protein wriggle, or larger moves such as loop conformation changes and domain motions.

Figure 2.5 shows the CDK-2 kinase in both active and inactive forms. The key loop to the binding site (flat in the picture) sees its conformation changed during the activation process.

The presence of multiple domains in proteins gives rise to a great deal of flexibility and mobility[27]. Several domain motions can occur to change the conformation of a protein[27,28](see figure 2.6). Most of the time, when domain motion occurs, the internal conformation of the domain remains the same, whereas the conformation of the protein is changed. Large moves are part of the activation process of most cellular proteins. However such reor-

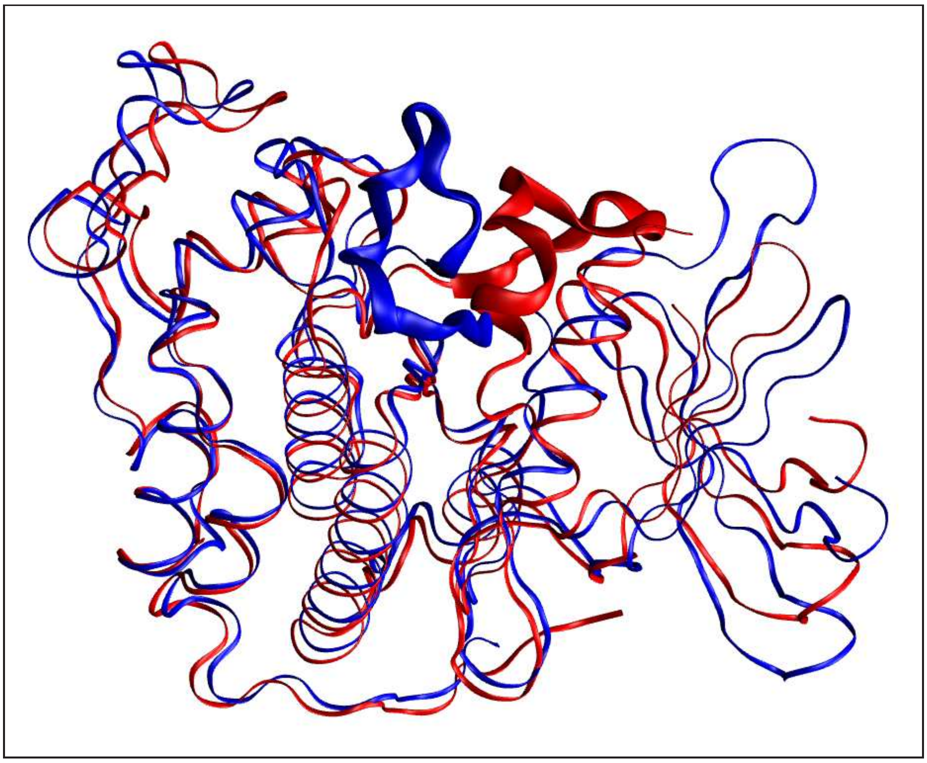


Figure 2.5: Superimposition of active (blue PDB code 2C5P) and inactive (red PDB code 1PXM) structures of the CDK2 kinase.

ganisations occur mostly on the μ-second time scale and undergo significant conformational rearrangement (more information on loop and domain reorganisation is in chapters 6 and 7). Investigating such reorganisations is of important biological interest and could lead to an increase of the efficiency of targeting specific conformational states. To be able to design a drug as selective as possible to bind its target, a perfect understanding of the activation pathway is needed. However this knowledge is actually one of the main challenges in molecular modelling. Classical methods fail to reach such aims. Owing to time scale problems, the MD simulation is not capable of sampling such large scale motions. However, the MD technique is usually chosen over MC to simulate proteins even though there are no absolute rules(see references[30] and[31] for examples of studies using the Monte Carlo method). The Monte Carlo method fails to sample such changes too. However, specific al-

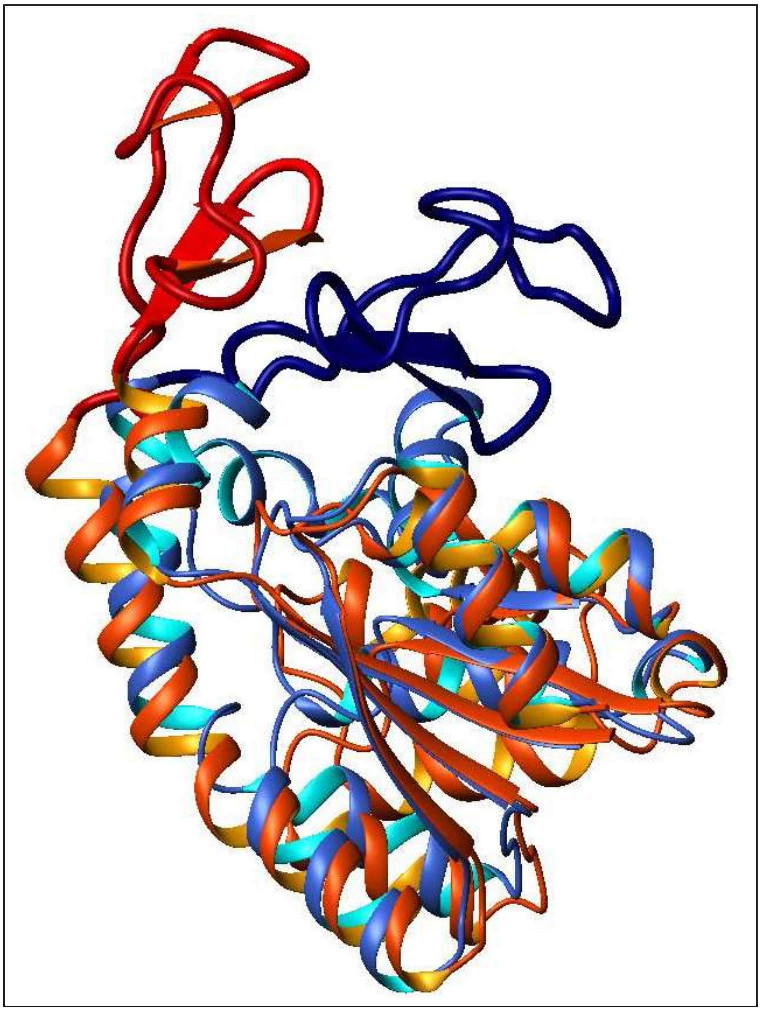


Figure 2.6: The LID domain of the Adenylate Kinase is in an open conformation, if no ATP is bound to the active site (red). The LID domain closes (dark blue) when an ATP molecule binds to the active site[29].

gorithms to model large backbone moves will enable us to sample large scale displacements and increase the backbone sampling. However, is the sampling provided by these novel methods for MC simulations enough to sample large scale reorganisation?

2.3 More to protein-ligand binding

Ligand binding is not just a matter of change in the shape of the protein structure. The whole process of computing the binding free energy of a ligand involves several enthalpic and entropic contributions from the ligand, the protein and the solvent[32–34]. All these terms represent the work necessary to move a ligand from the bulk (solvent) into the solvated binding pocket (including desolvating the binding pocket). Such terms are represented in table 2.2. Depending upon the nature of the ligand and the residues involved in the binding mechanism, the enthalpic or entropic contributions can have great influence upon the binding. Binding processes can be enthalpy driven or entropy driven and there is no absolute rules to predict *a priori* the binding affinity between a receptor and a ligand. However, the use of computational methods can approximate the estimation of the binding free energy.

Enthalpic terms (ΔH)	Entropic terms (ΔS)
New solute-ligand interaction	Protein degrees of freedom
Change in ligand/protein structure	Ligand degrees of freedom
Ligand desolvation	Ligand desolvation
Protein/complex desolvation	Protein/complex desolvation

Table 2.2: Enthalpic and entropic contribution to the protein/ligand binding.

The next section will give details on the theory behind molecular dynamics and the Monte Carlo method and the thermodynamics beyond the estimation of absolute and relative binding free energy.

Chapter 3

Standard methods for molecular modelling

This chapter briefly overviews MD and MC theory. For further interest, references [17,35–38] can be consulted. Other methods such as scoring functions and docking will be briefly described, and theories and equations beyond implicit solvation and free energy perturbation will be detailed.

Molecular modelling simulation is a technique for computing the equilibrium and transport properties of many body systems. The nuclear constituents of the system, are modelled to obey to the law of classical mechanics in terms of forces and energy (hence the name of molecular modelling).

3.1 Potentials and force fields

To model the behaviour of a biological system using the law of classical mechanics, a set of parameters and equations used to model the real system has to be built. This set of parameters and equations is referred to force field. The basic functional form of a force field encapsulates both bonded terms relating to atoms that are linked by covalent bonds, and non-bonded (also called "non-covalent") terms describing the long-range electrostatic and van der Waals forces. Force field parameters are derived from experiment and/or high-level quantum mechanical calculations.

The most popular forcefields in biological simulations are the AMBER[39–41] (developed to model DNA and protein), CHARMM[42,43] (developed to model proteins), GROMOS[44] (developed to model condensed phase of alkanes) and OPLS[45] (developed to model physical properties of liquids) forcefield. They are all-atom force fields, where every atom including the hydrogen is represented, but some can use the united atom model. The specific decomposition of the terms depends on the force field, but a general form for the total energy in an additive force field can be written as:

$$E_{total} = E_{bond} + E_{angle} + E_{dihedral} + E_{electrostatic} + E_{vanderWaals} \tag{3.1}$$

For the AMBER[39–41] force field the individual constituents can be expressed as follow:

$$E_{bond} = \sum_{bonds} K_r(r - r_{eq})^2 \tag{3.2}$$

$$E_{angle} = \sum_{angles} K_\theta(\theta - \theta_{eq})^2 \tag{3.3}$$

$$E_{dihedral} = \sum_{dihedrals} \frac{V_N}{2}[1 + \cos(n\phi - \gamma)] \tag{3.4}$$

$$E_{electrostatic} = \sum_{pairs} \frac{q_i q_j}{4\pi\epsilon_0 r} \tag{3.5}$$

$$E_{vanderWaals} = \sum_{pairs} 4\epsilon\Big[\Big(\frac{\sigma}{r}\Big)^{12} - \Big(\frac{\sigma}{r}\Big)^{6}\Big] \tag{3.6}$$

Bond and angle parameters are described as simple harmonic oscillators with a force constant and an equilibrium position, dihedral parameters by a Fourier series with coefficients (V_N), dihedral angle (ϕ) and a phase (γ). Non bonded interaction are treated through the use of a Coulombic potential (Equation 3.5) depending on the atomic charges and the distance between the two atoms of a pair, and through a Lennard-Jones (LJ) potential for the van der Waals interactions (Equation 3.6). Force fields are parametrised to reproduce experimental results (such as hydration absolute free energies) and quantum results.

3.2 Molecular Dynamics

In molecular dynamics simulations, we choose a system with N particles and we solve Newton's equations of motion for this system until the properties of the system no longer present a drift with time (equilibration period). Then after equilibration, measurement of the physical properties is performed. Newton's laws postulate that:

- A body continues to move in a straight line at constant velocity unless a force acts upon it.
- Force equals the rate of change of momentum.
- To every action there is an equal and opposite reaction.

Solving the differential equation embodied in Newton's second law ($F = ma$) gives us the trajectory:

$$\frac{d^2 x_i}{dt^2} = \frac{F_{x_i}}{m_i} \tag{3.7}$$

Equation 3.7 describes the motion of a particle of mass m_i along one coordinate (x_i) with F_{x_i} being the force applied on the particle in that direction. The first molecular dynamics simulation was performed in 1957 using a hard sphere model for the pair potential[46]. A more realistic approach consists of using a continuous potential. The force of each particle will change whenever the particle changes its position or whenever a particle with which it interacts changes position. The problem is that the continuous potential for a multiple body system makes the integration analytically impossible for system with more than two bodies. To solve this, the integration is broken into small steps each separated in time by a time step δt. The total force on each particle in the system at the time t is calculated as the vector sum of its interactions with the other particles. Newton's second law is used to calculate the acceleration from the forces. Accelerations are combined with position and velocities at the time t to compute the change in the configuration and to obtain the coordinates and the velocities at the time $t + \delta t$. This process is repeated iteratively until the end of the simulation. This value of δt captures all the changes in the degrees of freedom of the system, and forces or potential are conserved. The force $\boldsymbol{F}_i$ applied at a particle i at the time t depends on the potential energy $\boldsymbol{V}_i$ of this particle:

$$\boldsymbol{F}_i \;=\; -\nabla_{r_i}\boldsymbol{V}_i \tag{3.8}$$

So a classical MD algorithm could be written:

- Get the coordinates and the velocities of all the particles of the system.
- Compute the potential energy and get the force for each particle.
- Use the coordinates, velocities and the force of each atom to get the new sets of coordinates and velocities.
- Repeat.

At each step all the interactions, velocities and positions have to be recomputed which makes this method very expensive in CPU time. However the use of specific algorithms such as the velocity Verlet[47], described in the equations 3.9 to 3.11 enables faster computations.

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t) + \frac{\delta}{2}t^2\mathbf{a}(t) \tag{3.9}$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t+\frac{\delta t}{2}) + \frac{\delta t}{2}\mathbf{a}(t+\delta t) \tag{3.10}$$

$$\mathbf{v}(t+\frac{\delta t}{2}) = \mathbf{v}(t) + \frac{\delta t}{2}\mathbf{a}(t) \tag{3.11}$$

The velocity Verlet algorithm[47] manages the explicit velocities of all the constituent of the system. This algorithm is time reversible. To conserve energy during the integration, the time step has to be used in such way that the forces remain approximately constant. To keep the forces constant, the correct time step δt to use for a protein system is 1 fs, so the fast vibration of the bonds involving hydrogens can be sampled accurately. Computational time can be gained by using the SHAKE[48] algorithm to constrain bonds involving hydrogen, allowing the time step to be increased from 1 to 2 fs and hence halving the time of the computation. But, despite the use of such algorithms, simulations for more than 1 ms on a large protein are not tractable in a human time frame due to the cost in computer time.

3.3 Metropolis Monte-Carlo Method

The Monte Carlo method was developed at the end of the second world war. This statistical method is based on the generation of an important quantity of random numbers like in the casinos (hence the name from the Principality in the south of France famous for its casino) to solve conformational problems.

Statistical mechanics aims to explain thermodynamics of an ensemble (macroscopic properties *e.g.* temperature, pressure etc) by collecting the mechanical properties of the constituent of the ensemble (microscopic properties such as atomic positions or velocities). It all started with the law of gas, $PV = nRT$ from Boyle in 1661, but during the nineteenth century an uneasy feeling was growing among the scientific community as to whether or not the model would be able to explain individual atomistic properties[*].

Collecting a set of data for all the constituents of a macroscopic ensemble is usually very costly due to the curse of the dimensionality. This can be explained very simply by the following analogy.

Considering a unit sphere of dimension k (hypercube). The volume of the

[*] Gibbs stated in the introduction of his book *Elementary Principles in Statistical Mechanics.*[49]: The laws of thermodynamics, as empirically determined, express the approximate and probable behaviour of systems of a great number of particles, or, more precisely, they express the laws of mechanics for such systems as they appear to beings who have not the fineness of perception to enable them to appreciate quantities of the order of magnitude of those which relate to single particles, and who cannot repeat their experiments often enough to obtain any but the most probable results.

sphere is given by the formula:

$$V = \int_S dx_1 ... dx_k \tag{3.12}$$

whose solution is:

$$V = \frac{\pi^{k/2}}{\Gamma(\frac{k}{2}+1)} \tag{3.13}$$

Γ being the gamma function.

A solution of equation 3.12 using quadrature methods can be obtained by computing the ratio of the sphere and its bounding cube. It leads to the reformulation of equation 3.13 as:

$$I = \frac{V}{V_R} = \frac{\int_{[-1,1]^k} I_S(x_1, ..., x_k) dx_1, ..., dx_k}{\int_{[-1,1]^k} dx_1, ..., dx_k} \tag{3.14}$$

The function $I_S(X)$ takes the value 1 if X belong to the sphere or 0 if X belongs to the cube but not the sphere.

To approximate the solution of 3.14, a uniform lattice of point spread over $[-1,1]^k$ is built. Then the integrand over the $[-1,1]^k$ interval is averaged. For a lattice of m points per dimension a total number of m^k points have to be sampled. The number of points required to compute the average of the integral, increases exponentially with the number of dimension of the hypercube. For the unit cell a lattice with a 0.01 mesh will require 100 points, for a circle ($k = 2$) 10000 points are needed, and for a sphere one million points are needed. Now to obtain the same 0.01 lattice spacing for a 10^{10} hypercube, 10^{20} sampling points will be required.

Rather than using quadrature, one way to estimate the quantity I would be to use the Monte Carlo method[38] where instead of m points for each dimension k, a total of N point are randomly spread across the hyper cube ($[-1,1]$). The solution to the equation 3.14 can be estimated by the average of all the points:

$$I_{est} = \frac{1}{N}\sum_{i=1}^{N} I_S(X_i) \tag{3.15}$$

Where for an ergodic system, the limit of I_{est} when $N \to \infty$ is I. We can calculate the volume of the $[-1, 1]$ box for the dimension k as 2^k. The value of the ratio of the sphere and its bounding cube I becomes then:

$$I = \frac{V}{V_R} = \frac{\pi^{k/2}}{\Gamma(\frac{k}{2}+1)2^k} \tag{3.16}$$

The ratio I is now easily computed and values are plotted table 3.1

k	$\frac{V}{V_R}$
1	1.00×10^{0}
2	7.85×10^{-1}
3	5.24×10^{-1}
10	2.49×10^{-3}
100	1.87×10^{-69}

Table 3.1: Ratio of the sphere to its bounding cube $\frac{V}{V_R}$ for different dimensions

We can see the problem of the Monte Carlo method with a large number of degrees of freedom (typically sampling a protein). Most of the point are taken outside the sphere of interest (in this particular example, sphere having both a practical and metaphorical meaning). If we want to use the Monte Carlo method to sample a general property A of a given system ε of N particles, we are likely to experience the same limitations:

$$\langle A \rangle_\varepsilon = \int A(r^N)\rho_\epsilon(r^N)dr^N \tag{3.17}$$

where $\rho_\epsilon(r^N)$ is the probability of the system being in the configuration r^N.

According to the Boltzmann distribution, this probability in the canonical ensemble (NVT) can be expressed as:

$$\rho_{NVT} = \frac{exp(-\beta\mathcal{U}(\boldsymbol{r}^N))}{\int exp(-\beta\mathcal{U}(\boldsymbol{r}^N))d\boldsymbol{r}^N} = Z^{-1}exp(-\beta\mathcal{U}(\boldsymbol{r}^N)) \tag{3.18}$$

where $\beta = 1/k_\beta T$ and Z is the configuration integral over all the ensemble $\int exp(-\beta\mathcal{U}(\boldsymbol{r}^N))dr^N$. The term $\mathcal{U}(\boldsymbol{r}^N))$ is the energy of the system in the state r^N. Using a Boltzmann distribution, the ratio of high energy states over the low energy states is such that most of the configurations generated at random are located in the region of the phase space where the system has high energy configurations (corresponding to non-physical configurations) and thus contributes near to zero to the integral Z (in the case of the hypercube most of the sampling was performed outside the sphere).

To be able to use the Monte Carlo method to solve chemical problems, the method has to be adapted. A method developed by Metropolis et al.[17,38] called Metropolis Monte Carlo, biases the generation of configurations towards those that make the most important contributions to the configuration integral, those being the lower energy configurations.

The Metropolis Monte Carlo method uses an importance sampling technique in which the use of a distribution function ρ(x) allows function evaluation to be concentrated in the region of space that makes important contributions to the integral (*i.e.* low energy configurations). In the simple Monte Carlo integration method, states with both high and low energy are generated with equal probability and then a weight of $\exp(-\mathcal{U}(\boldsymbol{r}^N)/k_\beta T)$ is assigned to them for the calculation of properties in the canonical ensemble. In the Metropolis scheme, the states are generated with a probability of $\exp(-\mathcal{U}(\boldsymbol{r}^N)/k_\beta T)$ and each is counted equally. The Metropolis algorithm generates a Markov chain of states which satisfies the two following conditions:

- Each outcome depends only on the previous one.

- Each trial belongs to a finite set of possible outcomes.

Suppose that the system is in state m, the possibility of jumping to the state n is the $N \times M$ transition matrix π_{mn}. The probability of a system being in a particular state is represented by the vector $\boldsymbol{\rho}$:

$$\boldsymbol{\rho} = (\rho_1, \rho_2, ..., \rho_m, \rho_n, ..., \rho_N) \tag{3.19}$$

Thus, the probability for an initial randomly chosen configuration $\boldsymbol{\rho}(1)$ to jump into a second state $\boldsymbol{\rho}(2)$ is given by:

$$\boldsymbol{\rho}(2) = \boldsymbol{\rho}(1)\pi \tag{3.20}$$

The probability of the n^{th} state is:

$$\boldsymbol{\rho}(n) = \boldsymbol{\rho}(n-1)\pi = ... = \boldsymbol{\rho}(2)\pi^{(n-1)} = \boldsymbol{\rho}(1)\pi^n \tag{3.21}$$

and the limiting distribution for a Markov chain is given by:

$$\boldsymbol{\rho}_{limit} = \lim_{n\to\infty} \boldsymbol{\rho}(1)\pi^N. \tag{3.22}$$

When this limit is reached, we can now write the reverse distribution condition: $\boldsymbol{\rho}_{limit} = \boldsymbol{\rho}_{limit}\boldsymbol{\pi}$. This means that for an equilibrium ensemble, each element of the probability vector must satisfy the following condition:

$$\sum_m \rho_m \pi_{mn} = \rho_n \tag{3.23}$$

The transition matrix π gives the probability of jumping from one configuration to another ($n \to m$). This probability can be given by multiplying the probability of making a move from a state n to state m (α_{nm}) by the probability of accepting this trial move ($acc(n \to m)$). The matrix A (called the underlying matrix) is directly related to the new trial configuration pathway. Assuming that the stochastic matrix A is symmetrical (i.e. the probability of a jump from n to m is the same as that from m to n), owing to the condition above, we can now write:

$$\rho \times \pi_{nm} = \rho \times \pi_{mn} \tag{3.24}$$

$$\rho_n \times \alpha_{nm} \times acc(n \rightarrow m) = \rho_m \times \alpha_{mn} \times acc(m \rightarrow n) \tag{3.25}$$

$$\frac{\alpha_{nm} \times acc(n \rightarrow m)}{\alpha_{mn} \times acc(m \rightarrow n)} = \frac{\rho_m}{\rho_n} \tag{3.26}$$

Using the Boltzmann equation for canonical ensemble, we can now express the famous Metropolis criterion as:

$$acc(n \rightarrow m) = min(1, exp(-\beta(\mathcal{U}(m) - \mathcal{U}(n)))) \tag{3.27}$$

So a typical Monte Carlo algorithm would be described as follows:

- Collect the structural information and compute the energy of the system in state n
- Perform random moves on degrees of freedom to get the new configuration m
- Collect the structural information and compute the energy of the system in state m
- Perform the acceptance test:
  - If $(\mathcal{U}(m))$ is lower than $(\mathcal{U}(n))$ accept the move
  - If $(\mathcal{U}(m))$ is greater than $(\mathcal{U}(n))$ accept the move according to 3.21. Chose a random number between 0 and 1. If the random number is smaller than $exp(-\beta(\mathcal{U}(m) - \mathcal{U}(n)))$ reject the move and keep the conformation n. If the random number is greater than $exp(-\beta(\mathcal{U}(m) - \mathcal{U}(n)))$ accept the move and keep the conformation m
- Go back to the step one with the new conformation (n or m).

In Metropolis Monte Carlo, moving a single atom is not really a problem. Using a cartesian frame of reference, a small change in the coordinates can give the new position of the atom:

$$\begin{aligned} x_{i'} &= x_i + \delta_{lx} \\ y_{i'} &= y_i + \delta_{ly} \\ z_{i'} &= z_i + \delta_{lz} \end{aligned} \tag{3.28}$$

where $\delta_{lx,ly,lz}$ are randomly chosen in the range of δ_{max} (adjustable parameter) and the energy of the new configuration is then calculated. For a small system it is easy to generate a random configuration, but for such systems as proteins, owing to their complex structures, a special implementation must be used for sampling a judicious phase space area.

3.3.1 Standard Protein sampling method

To sample proteins, specific methods have to be used. One cannot hope that randomly moving cartesian coordinates will lead to a conformation that is acceptable from an energetic point of view. Sampling protein can be separated into sampling the side chains, or sampling the backbone.

Sampling the side chains is not very challenging. In both the widely used MC package MCPRO[50] and ProtoMS[51] the thrashing method is used to sample the side chains. This is done by small changes in the internal *degrees of freedom (dof)* along the side chain. The values of bond angles and dihedrals are changed in the Z matrix, and the cartesian coordinates are rebuilt. The new conformation is accepted or rejected according to the Metropolis test[17]. In most cases the χ angle is in this case considered as part of the side chain. The thrashing method applied to side chains is fast and efficient, as the number of moving atoms is generally small.

However, sampling accurately the backbone is not as easy. In MCPRO[50] the thrashing method is used (for more details about the possibilities of MCPRO see reference[52]) to sample protein backbones, as well as translations and rotations of the cartesian coordinates. A Z-matrix is used to store bond length, angle and dihedrals to be sampled. The value of one of the previous *dof* is changed in the Z-matrix, the cartesian coordinates are rebuilt and the Metropolis test[17] is performed using the potential energy of the new conformation. This method is less computationally demanding than MD simulations. The only changes in the potential energy are resulting from the change in the *dof*, so the energy of only one length, one angle, or one dihedral, has to be recomputed, as well as the moving non bonded interactions. However, this method possesses some weaknesses. The most important is a poor acceptance rate. If a *dof* is moved even by a small amount, atom clashes may occur in a region far away as large displacements due to the protein geometry occur. The other weakness is that most of the non bonded interactions have to be recalculated after the move.

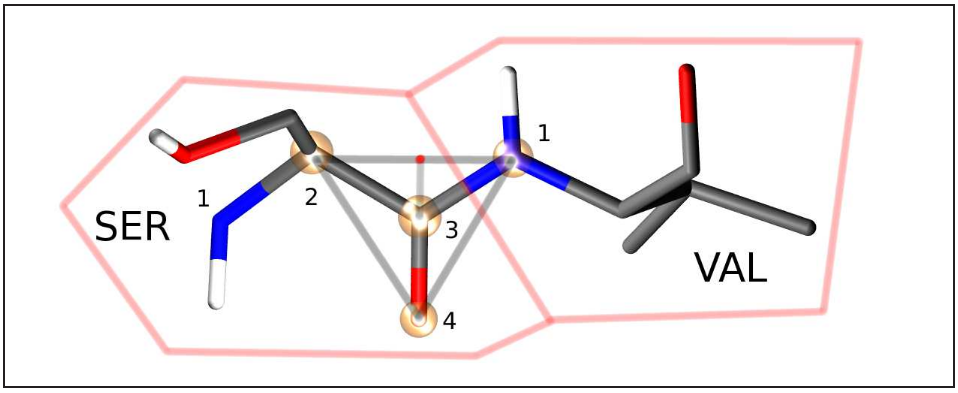


Figure 3.1: The four backbone atoms for two neighbouring residues are shown above. The protein backbone-move moves the last three backbone atoms *bbatoms* of one residue and the first bbatom of the next residue. This is because the moves assumes that these four *bbatoms* form a rigid triangle (as is shown by the grey lines).

On the other hand, the ProtoMS package[51] uses a rigid unit backbone model for the protein. The rigid unit backbone is defined by the rigid unit made of the atoms C, Cα, and O of the residue i and the atom N of the residue *i+1*. Moves assume this unit to be a rigid triangle, with the atom C at its centre. The rigid unit can be rotated, translated, and every atom attached to this unit will be rotated and translated as well. The rigid unit backbone is presented in figure 3.1.

In ProtoMS[51], the backbone and side chains can be moved independently as well as the whole residue (backbone plus side chain). When a backbone move is performed, the internal *dof* are kept fixed.

Moves as designed in ProtoMS[51] are really localised. Very few internal coordinates change at each step, so very few interactions have to be recomputed, which gives a noticeable gain in efficiency. These moves stretch bond lengths and change the bond and dihedral angles of two residue. These moves are accepted providing that the changes in bond length are not too important. This is one of the main weakness of the method: the moves have to be close to the previous conditions, and therefore, poor sampling of the conformational phase space occurs.

3.4 Free energy method

The term free energy refers to the thermodynamic quantity of perhaps the greatest importance for the chemist. This is because the value of the free energy gives direct knowledge of the direction of a reaction. The binding free energy for a host-guest system can be related to the strength (and the direction) of the binding process. A negative binding energy will refer to a favourable interaction, whereas a positive energy will refer to a non-favourable interaction. The bigger the absolute value of the binding free energy, the more favourable (or non-favourable) the interaction is. In the canonical ensemble, the Helmholtz free energy can be computed using the equation 3.29:

$$G = -k_B T ln Q \tag{3.29}$$

If the partition function is the NPT ensemble rather the the canonical ensemble, G would be the Gibbs free energy. The partition function necessary to compute the free energy is a function of the exponential energy of all the possible configuration Γ of the system[35]:

$$Q = \sum_{\Gamma} exp\Big(\frac{-E(\Gamma)}{k_B T}\Big) \tag{3.30}$$

This equation adopts at the limit the form:

$$Q = \frac{1}{N!}\frac{1}{h^{3N}}\int exp\Big(\frac{-E(\Gamma)}{k_B T}\Big)d\Gamma \tag{3.31}$$

This equation is valid as a classical limit of the partition function. N is the number of atom of the system and h is Planck's constant. The total energy $E(\Gamma)$ can be express as the sum of the potential energy E_p function of the coordinates q and the kinetic energy E_k function of the momentum p. Thus the equation 3.31 can be rewritten:

$$Q = \frac{1}{N!}\frac{1}{h^{3N}}\int_q\int_p exp - \Big(\frac{E_p(q) + E_k(p)}{k_B T}\Big)d\boldsymbol{p}d\boldsymbol{q} \tag{3.32}$$

Momenta and coordinates of a system are independent so the kinetic and potential part of the partition function can be separated, and the partition function can be express as the product of both energies.

$$Q = \frac{1}{N!}\int_p exp(\frac{-E_k(p)}{k_B T})d\boldsymbol{p}\frac{1}{h^{3N}}\int_q exp(\frac{-E_p(q)}{k_B T})d\boldsymbol{q} \tag{3.33}$$

$$= Q_k Q_p \tag{3.34}$$

The potential energy partition function cannot be solved analytically due to the large number of internal and external energy terms that needs to be computed. The evaluation of Q_p can be performed analytically (analytical solution for the kinetic partition function can be found using the particle in the box model[35]). Most of the time, the factor $\frac{1}{h^{3N}}$ is dropped, and the *configuration integral* Z is defined instead as:

$$Z = \int_q exp(\frac{-E_p(q)}{k_B T})d\boldsymbol{q} \tag{3.35}$$

the integral function Z is still very difficult to compute. Owing to the high dimension of Z (N) the numerical integration converges slowly for system such as a protein[35].

3.4.1 Free energy perturbation

Absolute free energies, are most of the time used in the context to compare two different system, typically answering the question: does a molecule A interact better with our receptor P than the molecule B? So rather than computing the two different absolute free energies, it is easier to compute the relative free energy $\Delta G_{A\rightarrow B}$ between the two systems. This was first performed by Zwanzig in 1954[53].

$$\begin{aligned}
\Delta G_{A\rightarrow B} &= G_B - G_A \\
&= (-\frac{1}{\beta}lnQ_B) - (-\frac{1}{\beta}lnQ_A) \\
&= -\frac{1}{\beta}ln\Big[\frac{Q_B}{Q_A}\Big] \\
&= -\frac{1}{\beta}ln\Big[\frac{\int exp(-\beta U_B(r^N))dr^N}{\int exp(-\beta U_A(r^N))dr^N}\Big]
\end{aligned}$$

multiply by $1 = exp(-\beta U_A(r^N))exp(\beta U_A(r^N))$ gives:

$$
\begin{aligned}
&= -\frac{1}{\beta}ln\Big[\frac{\int exp(-\beta U_B(r^N)) \times exp(-\beta U_A(r^N))exp(\beta U_A(r^N))dr^N}{\int exp(-\beta U_A(r^N))dr^N}\Big]\\
&= -\frac{1}{\beta}ln\Big[\frac{\int exp(-\beta U_A(r^N)) \times exp(-\beta(U_B(r^N)) - U_A(r^N))dr^N}{\int exp(-\beta U_A(r^N))dr^N}\Big]\\
&= -\frac{1}{\beta}ln\Big[\int \frac{exp(-\beta U_A(r^N))}{Q_A} \times exp(-\beta(U_B - U_A)(r^N))dr^N\Big]\\
&= -\frac{1}{\beta}ln\Big[\int \pi_A(r^N) \times exp(-\beta\Delta U_{AB}(r^N))dr^N\Big]\\
&= -\frac{1}{\beta}ln\Big\langle exp(-\beta\Delta U_{AB}(r^N))\Big\rangle_A
\end{aligned}
\tag{3.36}
$$

So the relative free energy is the ensemble average of the exponential of the Boltzmann weighted difference between the two potential U_A and U_B. A method called Free Energy Perturbation is used in computer simulation to solve the Zwanzig equation. At each step i (or t in the case of MD) the value of the of the quantity $\exp(-\Delta U_{AB}(i)/k_BT)$ is accumulated, and averaged at the end of the simulation. The problem with solving equation 3.36 is that the two potentials have to be located in a region of the phase space close to each other. Problems occur when the two configurations are located in two different regions of the phase space. For example, if the phase space of low energy states for B are located in the region of high energy states for A, then the relative free energy $\Delta G_{A\to B}$ is likely to be over estimated, as the potential U_A will not generate enough configurations corresponding to the potential U_B. If the potentials are switched, the relative free energy $\Delta G_{B\to A}$ will be over estimated as well. The difference between the two values of the free energy is referred to as hysteresis. The larger the hysteresis, the more inaccurate the calculation of the energy will be.

However, the relative free energy is a state function and thus only depends on the two states A and B. Different pathways to join both states do not change the value of $\Delta G_{B\to A}$. So a simple solution is to imagine a pathway linking the two states A and B in such way that the hysteresis is minimised. Generally a multi-stage calculation is implemented using the coupling parameter λ to define intermediate states (potentials) $U_{P(\lambda)}$ between the potentials U_A and U_B, see figure 3.2. So the relative free energy $\Delta G_{B\rightarrow A}$

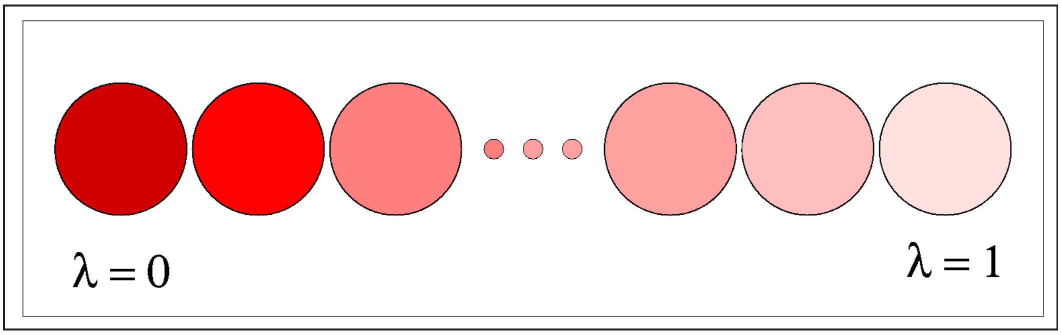


Figure 3.2: New pathway using a multi-stage calculation process.

can be rewritten as the sum of the differences:

$$G_B - G_A = \Delta G = \sum_{\lambda=0}^{1} -k_B T ln \langle exp(-\Delta U')/k_B T\rangle_{\lambda_k} \tag{3.37}$$

where $\Delta U' = U_{P(\lambda_{k+1})} - U_{P(\lambda_k)}$.

3.4.2 Thermodynamic integration

Another way to access to the relative free energy is to compute the numerical integral of the free energy gradient ($\frac{\partial G}{\partial \lambda}$). This method is called thermodynamic integration (TI)[35]. The gradient $(\frac{\partial G}{\partial \lambda})_\lambda$ is estimated (numerically or analytically) for each λ during a set of simulation run at different λ. Once known, the free energy gradient is integrated to yield to the relative free energy along the λ coordinate:

$$G_{\lambda=1} - G_{\lambda=0} = \int_0^1 \left(\frac{\partial G}{\partial \lambda}\right)_\lambda d\lambda \tag{3.38}$$

The trapezium rule is often use to evaluate the integral and access the relative free energy[35]. The free energy gradient is equal to the ensemble average of the potential:

$$\int_0^1 \Big(\frac{\partial G}{\partial \lambda}\Big)_\lambda d\lambda = \int_0^1 \Big\langle \frac{\partial U}{\partial \lambda} \Big\rangle_\lambda d\lambda \tag{3.39}$$

For a forcefield the gradient can be evaluated by calculating the gradient of each term directly with respect to λ. The finite difference $(\frac{\Delta G}{\Delta \lambda})_\lambda$ can be calculated as an alternative of the gradient. For each lambda, the evaluation of the Zwanzig equation for a reference state λ should lead to the same energy for both the forward and backward estimates (respectively $\lambda + \Delta\lambda$ $\lambda - \Delta\lambda$), provided $\Delta\lambda$ is small enough and the number of steps is such that the Zwanzig energy has converged.

Both free energy perturbation and thermodynamic integration are known to reproduce accurately some experimental results on a broad range of systems[30,54–56].

3.4.3 Replica Exchange Thermodynamic Interaction

Novel methods have been implemented to enhance the accuracy of the thermodynamic integration method, inspired by generalised ensembles and called Replica Exchange Thermodynamic Integration[57,58] (RETI). RETI considers the Hamiltonians of the system for different coupling parameters λ to be part of the same generalised ensemble. Hence it is possible to connect to different λ in a free energy simulation. During a RETI simulation, a set of replicas that cover the range of λ are run, and periodically, moves between the replicas i and j of the Hamiltonians H_A and H_B are performed. Moves are accepted according to the test

$$exp\Big[\beta(E_B(j) - E_B(i) - E_A(j) + E_A(i))\Big] \geq rand(0,1) \tag{3.40}$$

where $E_B(j)$ and $E_B(i)$ are the Hamiltonian of the state B for the replicas i and j, and $E_A(j)$ and $E_A(i)$ are the Hamiltonian of the state A.

The RETI simulation has little extra-cost over a standard thermodynamic integration or free energy perturbation simulation, as all the replicas already exist for the simulation. RETI provides enhanced sampling, as the method allows the different trajectories to access regions of the phase space that would otherwise be in-accessible. For example when one λ_i exchange with a λ_j located in a region of the phase space separated by a high energy barrier, performs some local sampling and then "jump back" to the original side of the energy barrier the RETI simulations allow all the replica to sample the high energy configuration thus enhancing the sampling.

3.5 Temperature replica exchange

Owing the nature of the Metropolis test[17,38], an increase of temperature is likely to lead in an increase of the acceptance rate and hence an increase in the exploration of the energy surface.

Ideally the same level of sampling would benefit simulation run at standard temperature (298K) but due to the ruggedness of the potential energy surface, systems can get trapped into a local energy minimum. A simple and efficient way to achieve efficient sampling is to run parallel tempering (PT) simulations[59,60]. The idea of PT is to perform several concurrent simulations of different replicas of the same system at different temperatures and to exchange replicas between simulations i and j with probability:

$$p = min(1, exp(-(\beta_j - \beta_i)(E_i - E_j))) \tag{3.41}$$

where $\beta_i = 1/k_b T_i$ and E_i are the inverse temperatures and energies of the conformations respectively.

3.6 Modelling solvent

Most biological systems exist in an aqueous environment. To be realistic, computer simulations have to reproduce the effect of the solvent. The most obvious representation is an explicit solvation where each molecule of solvent is represented and interact with the system in a discrete fashion[61–64]. Using an explicit model of solvent, although probably accurate, means that most of the time thousands and thousands of new molecules (and their relative interactions) need to be computed, and most of the CPU time is used to re-compute solvent interactions:

- at each step all solute/solvent and solvent/solvent non bonded interactions for moving atoms needs to be re-computed.
- after a solute move, the solvent need to be reorganised around the solute.
- presence of the solvent may render large conformational changes difficult if not almost impossible.

The Generalised Born (GB) model is used to model a continuum dielectric potential to represent the solvent[65]. The electrostatics for a charged sphere q, dielectric constant ϵ_{vac} and a radius α can be expressed as:

$$G_{vac} = \frac{q^2}{2\epsilon_{vac}\alpha} \tag{3.42}$$

In a dielectric medium with a dielectric constant of ϵ_{solv}, the total electrostatic energy is shown to be:

$$G_{solv} = \frac{q^2}{2\epsilon_{solv}\alpha} \tag{3.43}$$

The difference between 3.43 and 3.42 expresses the electrostatic energy needed to transfer a spherical charged ion of radius α from a medium with a dielectric constant ϵ_{vac} to another with a dielectric constant ϵ_{solv}. This is known as the Born equation[65]:

$$\Delta G_{Born} = (\frac{1}{2\epsilon_{solv}} - \frac{1}{2\epsilon_{vac}})\frac{q^2}{\alpha} \tag{3.44}$$

If we assume the protein to be composed of charged spheres with a charge q_i, a radius α_i and an interior dielectric of ϵ_i, then providing we can assume each atom to be distant enough to the other, then the sum of the coulombic interaction and the Born solvation energy can be written:

$$\Delta G_{tot} = \frac{1}{2}\sum_{i}\sum_{i\neq j}\frac{q_i q_j}{\epsilon_{solv} r_{ij}} - \frac{1}{2}(\frac{1}{\epsilon_{vac}} - \frac{1}{\epsilon_{solv}})\sum_{i}\frac{q_i^2}{\alpha_i} \tag{3.45}$$

Unfortunately, equation 3.45 is not valid for pairs where the radius α_i and the distance r_{ij} are too close. The Coulombic interactions can be split in two[66]:

$$\Delta G_{tot} = \frac{1}{2}\sum_{i}\sum_{i\neq j}\frac{q_i q_j}{\epsilon_{vac} r_{ij}} - \frac{1}{2}(\frac{1}{\epsilon_{vac}} - \frac{1}{\epsilon_{solv}})\sum_{i}\sum_{i\neq j}\frac{q_i q_j}{r_{ij}} - \frac{1}{2}(\frac{1}{\epsilon_{vac}} - \frac{1}{\epsilon_{solv}})\sum_{i}\frac{q_i^2}{\alpha_i} \tag{3.46}$$

This equation can be rewritten:

$$\Delta G_{tot} = \frac{1}{2}\sum_{i}\sum_{i\neq j}\frac{q_i q_j}{\epsilon_{vac} r_{ij}} + \Delta G_{GENBORN} \tag{3.47}$$

Where $\Delta G_{GENBORN}$ is:

$$\Delta G_{GENBORN} = -\frac{1}{2}(\frac{1}{\epsilon_{vac}} - \frac{1}{\epsilon_{solv}})\sum_{i}\sum_{j}\frac{q_i q_j}{\sqrt{r_{ij}^2 B_i B_j e^{\frac{-r_{ij}^2}{4B_i B_j}}}} \tag{3.48}$$

The quantity α_i of the equation 3.44 is replaced by the values B_i and B_j. The difficulty of equation 3.48 lies in computing the value of the Born radii B_i. Its value is not α_i and it is influenced by its surroundings. The original work from Still[66] uses a numerical method to compute the value of the Born Radii B_i:

- Consider a shell of thickness T_k surrounding the van der Waals surface of atom k.
- Weight the interior radius $(r_k - 0.5T_k)$ of this shell using the ratio of solvent accessible surface area A_k to the actual surface area.

- Repeat the weight for the exterior radius ($r_k + 0.5T_k$) and calculate the difference between weighted interior and exterior radii.
- Sum the difference between weighted interior and exterior radii for a series of concentric shells up to shell M which encompasses the whole of the van der Waals surface of the molecule.
- For shell M no weight is applied and the radius is simply added to the previous summation term, to obtain an effective Born radius, which is then used in equation 3.48

This method is very costly. The use of an analytical method such as the Pairwise De-screening Approximation (PDA) developed by Hawkins *et al*[67,68] makes the computation of the Born radii quicker.

$$\frac{1}{B_i} = -\frac{1}{2\alpha_i}\sum_{j\neq i}[\frac{1}{L_{ij}} - \frac{1}{U_{ij}} + \frac{R_{ij}}{4}(\frac{1}{U_{ij}^2} - \frac{1}{L_{ij}^2}) + \frac{1}{2R_{ij}} ln \frac{L_{ij}}{U_{ij}} + \frac{S_{ij}^2\alpha_j^2}{4R_{ij}}(\frac{1}{L_{ij}^2} - \frac{1}{U_{ij}^2})] \tag{3.49}$$

$$\begin{aligned} L_{ij} &= 1 \text{ if } R_{ij} + S_{ij}\alpha_j \leq \alpha_i \\ L_{ij} &= \alpha_i \text{ if } R_{ij} - S_{ij}\alpha_j \leq \alpha_i < R_{ij} + S_{ij}\alpha_j \\ L_{ij} &= R_{ij} - \alpha_j \text{ if } \alpha_i \leq R_{ij} - S_{ij}\alpha_j \\ U_{ij} &= 1 \text{ if } R_{ij} + S_{ij}\alpha_j \leq \alpha_i \\ U_{ij} &= R_{ij} + S_{ij}\alpha_j \text{ if } \alpha_i < R_{ij} + S_{ij}\alpha_j \end{aligned}$$

R_{ij} is the distance between the two spheres centred on atoms i and j and α_i the intrinsic born radius of the atom i. The PDA approximation tends to overestimate the Born radius. So the screening factor S_{ij} is introduced to correct for the over-estimate by scaling the Born radius. This means the scaling factor should have a value between 0 and 1.

However it would be wrong to only consider the GB equations, as definite "answers" to the solvation problem. Solvation not only deals with charges, but also volumes. So to solvate a solute, a cavity has to be formed (disturbing the hydrogen bonding network in the case of water) and solvent molecules have to reorganise around the solute. Solute atoms interact with solvent atoms, thus forming repulsive or attractive van der Waals interactions (owing to the solute-solvent distance, such interactions are mainly attractive).

Both effect are taken into account using a solvent accessible surface area (SASA) term for the solute[66,69].

$$G_{nonpol} = Gcav + G_{vdW} = \sum_{k=1}^{N} \sigma_k.SASA_k \tag{3.50}$$

The SASA is the surface "filled" by the solute that is non-accessible for the solvent. Water molecules are approximated to spheres with a 1.4Å radius, and such a sphere is rolled over the van der Waals surface of the solute to approximate the SASA. One of the drawbacks of the method is that the water sphere can only roll on the solute atoms on the outside. Thus buried atoms are not taken into account to build the SASA whereas they do interact with explicit solvent.

Combining both methods is referred as Generalized Born Surface Area (GBSA)[69]. Parametrisation of an accurate GBSA model is obtain by reproducing the experimental absolute hydration energy of ions and small molecules[69].

3.7 Virtual screening in computational chemistry

In the constraints of the pharmaceutical world, one would like to be able to virtually screen several thousand of compounds per day. However a such task is not feasible using rigorous methods.

Usually, to be able to sample several thousands of compounds a day some level of precision has to be sacrificed to the benefit of speed. The use of docking and scoring functions to rank the affinity of a broad set of ligands to a known structure is widely used in the pharmaceutical world[70].

Numerous number of docking algorithms are available for free or a nominal fee (in a review from Taylor *et al.* from 2002, 127 algorithms are mentioned[70]) each having its strenghts and weaknesses. Popular algorithms are Autodock[71], Gold[72] or Flexx2 [73,74]. Rather than describe each algorithm, the general principles of docking and scoring functions will be described in the following sections.

3.7.1 Docking

Docking is a computational method used to rank the affinity of ligands towards a specific 3D structure of a receptor. To be able to dock a ligand to a protein the structure of the receptor has to be suggested, and then the different ligands are docked into the receptor.

The docking process aims to explore translational and rotational degrees of freedom of a given ligand within the receptor. An ensemble of ligand conformations is generated as the docking proceed. The receptor is usually considered rigid. To perform the generation of the different conformations, Monte Carlo methods, genetic algorithms or incremental construction can be used.

The energy of the different conformations of the protein-ligand system is then approximated using a scoring function. The section below will describe how to approximate the energy.

3.7.2 Scoring functions

Scoring functions are computed using the sum of empirical terms associated to the different degrees of freedom:

$$\begin{aligned}\Delta G_{binding} = \Delta G_{solvent} + \Delta G_{conformation} + \Delta G_{intermolecular} + \Delta G_{rotation} \\ + \Delta G_{rotation/translation} + \Delta G_{vibration}\end{aligned}$$

However, the use of empirical terms to approximate the different energetic terms does not yield to exact ranking. Terms such as the entropic penalty of desolvation are usually badly represented or even neglected in the use of a scoring function. A study from Michel *et al.* compares the results of ranking a set of ligands using various docking algorithms and RETI and shows that scoring function methods do not yield a ranking as good as thermodynamic methods[57].

3.8 Concluding remarks

There are no strict rules regarding which method is to be applied to sample the conformational phase space of a system. Most of the time, common sense and experience leads to the choice of one method.

Owing to the time scales, sampling phenomena such as large conformational changes in proteins using MD is non tractable in human time. However, enhanced MD techniques could lead to a good sampling of such moves. The other possibility is to use equilibrium techniques such as MC. Sampling large moves using standard MC and explicit solvation is inefficient, so the use of an implicit solvation and specific algorithm to enhance the sampling are needed.

The following chapter will review several sophisticated implementations used in MC simulations to sample polymers and proteins.

Chapter 4

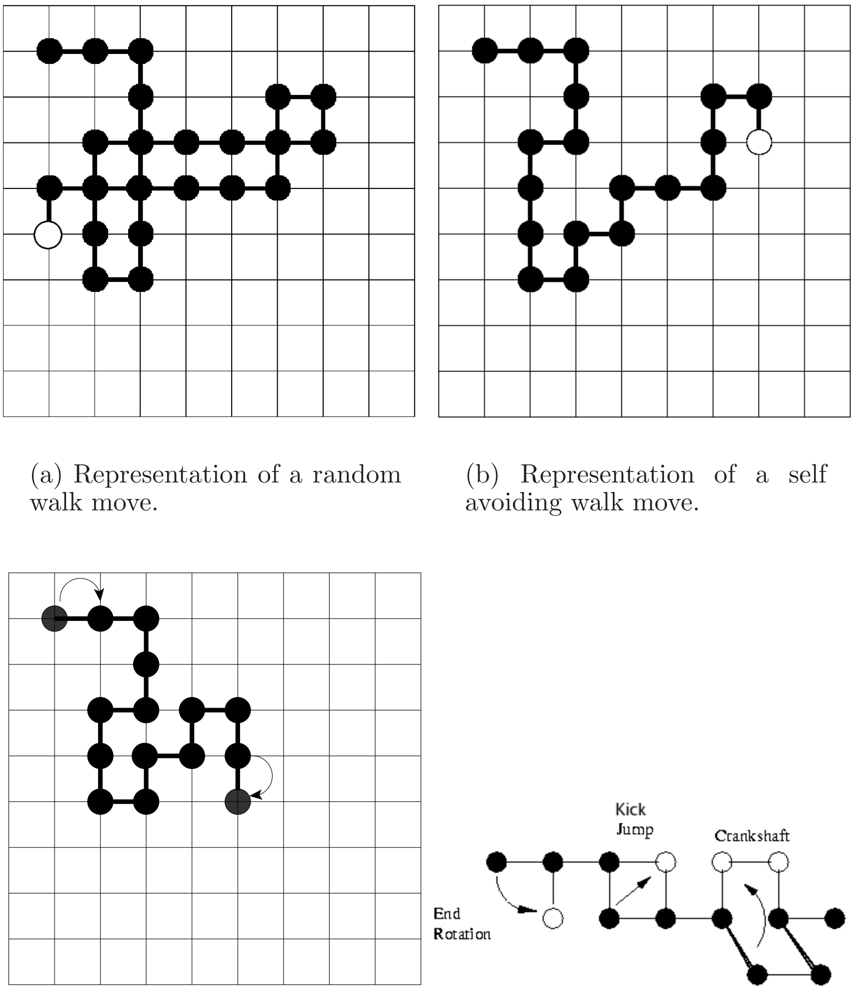
Non time-dependent move for polymers and proteins

Polymers are of great industrial importance. Theoretical studies under different conditions (temperature, density, chain lengths) may offer valuable insights in understanding their behaviour[75].

Several algorithms for sampling the conformational space of polymers exist. Lattice and off-lattice models of polymers such as the crankshaft[76], the reptation moves[77,78] or general bias algorithms[79–81] are widely used for polymers but are not efficient for heteropolymers such as proteins. New local or concerted moves[82–86] are more appropriate to sample moves of protein backbones.

4.1 Algorithms for polymer sampling

Using a lattice-polymer allows several simple moves, from the random walk to the Verdier-Stockmayer algorithm using a combination of several other moves (crankshaft, kink jump and end rotation)[76]. Schemes for the different moves are represented in figure 4.1. Sampling polymers is usually time consuming, due to physical properties (i.e. the chain cannot cross itself) and real motion algorithms will suffer from inefficiency. Random walk algorithms



(a) Representation of a random walk move.

(b) Representation of a self avoiding walk move.

(c) Representation of a reptation move.

(d) Representation of the Verde algorithm.

Figure 4.1: Scheme of several lattice Monte Carlo moves for polymers[36].

(figure 4.1(a)), change the lattice occupation of the polymer and most often lead to non-physical configurations, as nothing stops the polymer from "walking" onto itself.

To solve this problem, a set of constraints needs to be imposed (self avoiding walk moves 4.1(b)). This has been described first by Rosenbluth[79]. The use of the Rosenbluth sampling[79] to create polymer chains has solved the inefficiency in polymer sampling. The Rosenberg scheme aims to insert a polymer is a two step approach. First, a new conformation of the chain is generated by biasing the coordinates in such way that the polymer cannot cross itself. Next, the bias is corrected by re-weighing the system. In the original scheme a chain is rebuilt step by step with a bias favouring the conformations with a high Boltzmann factor. Then once the chain is totally rebuilt, detailed balance is fulfilled by a conformation-dependent weight applied to correct the bias. This method, although correct in theory, practically works mainly for short chains. One other possibility is to use the configuration bias Monte Carlo method (CBMC) (see ref[79–81,87]) that biases the chain towards low energy states (and thus avoiding crossing as high energy barriers).

Both methods are used in a rebuilding fashion often called reptation (the chain is locally rebuild at each step and the acceptance test is performed at the end), and can be applied to lattice as well as non-lattice models of polymers.

Kick jump and crankshaft (see figures 4.1(c), 4.1(d)) involve changes in dihedral and angles along the polymer chain. For the crankshaft it is easy to imagine a car crankshaft pushing the pistons up and down by rotating around an axle, main axis of rotation if fixed, but some parts of the crank undergo large moves rotating around the axle (pushing the piston up and down). Same happens here, the bond between two atoms (atoms 3 and 4 in figure 4.2) rotate around the adjacent parallel bond (bonds 1-2 and 5-6 in figure 4.2).

The kick jump move involve jumping from one corner of the lattice to the opposite one, changing the appropriate degrees of freedom (dof). Both the kick jump and the crankshaft can be used on and off lattice.

From the geometric construction of the previous algorithms, one can find very little use for these moves to sample proteins. For example, the reptation move only works in a case of a mono-residue protein. Crankshaft and kink jump would lead to steric clashes if applied in the protein core or binding pockets. Sampling proteins therefore requires specific moves. The section below investigates a few of the specific algorithms for proteins.

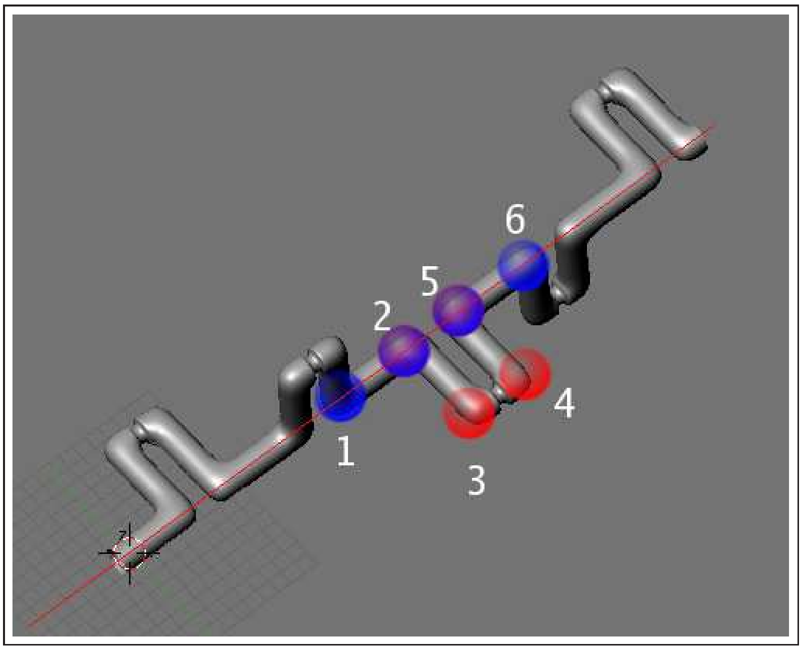


Figure 4.2: Representation of the crankshaft move[88]. The thin red line represents the rotation axis.

4.2 Algorithms for protein sampling

The concerted-rotation approach is a powerful method that can generate large local deformations by finding discrete solutions to the re-bridging problem described by Go and Schegara[89]. However, the method is not easy to implement and large local deformations may be difficult to accomplish if, for example, the chain is folded and has bulky side groups. The first mention of solving ring closure problems in a polymer chain was provided by Go and Schegara[82], but this method did not conserve the metric volume and hence failed to satisfy detailed balance. The algorithm by Dodd *et al.*[82] uses a jacobian matrix to conserve the metric volume and the detailed balance criterion, and is known as the concerted rotation algorithm, also referred to as CONROT. Other concerted algorithms exist, such as the concerted rotation with angles CRA[83], the gaussian bias[90], the lmProt algorithm[91], the wriggling motion[92,93], algorithm using rectangular shape models[94,95], algorithms derived from robotics[96–101] or the PAR-ROT algorithm[102]. These algorithms will be briefly described to give the reader an overview of the state of the art of sampling protein loops.

4.2.1 Non-Boltzmann weighted algorithms

The chain closure problem is well known in the field of robotics. The robot arm is a single chain consisting of joints connected by links. The first and last elements of the chain are special; they actually are not considered joints and are called the base and the effectors (see figure 4.3).

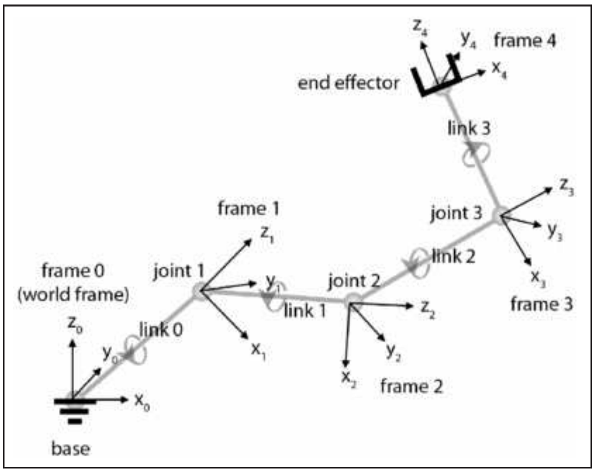


Figure 4.3: Representation of the robot arm[96].

The analogy of the robot arm and the protein is easy to understand; an atom between two bonds in the protein is represented by a joint connecting two links together in the robot arm. Then a frame of reference is attached to each joint/bond of the chain (see[75] for more details). In the paper by Lee *et al.*[96] the loop closure is solved by using the jacobian matrix relating the change of the effector position due to changes in the joints[75,96]. The algorithm works in the following way:

- An external force is applied to break the loop.
- The loop is closed by the use of the internal attractive forces. Closing the loop in such peculiar cases, means connecting the base to the effector where the loop has been broken.

This algorithm does not take into account steric clashes so this method is unable to solve complex loop motion. Other algorithms such as the *random loop generator* (RLG)[99,100] or the *rapidly-exploring random trees* (RTT)[98] use a *probabilistic road map* (PRM) approach to solve the ring closure problem[101]. The PRM algorithm, is a two step algorithm. First, a road map is built and stored as a graph with nodes corresponding to collision free configurations, and edges as path between the nodes. Second, the base and the effector of the robot, are connected to two nodes of the road map, and then the road map is search for a path linking the two nodes. The RLG algorithm does not suffer from the clash problem, as the algorithm is built in such way that the robot arm does not collide with itself or any other solid object. So the constraints are set when the mapping is built (in this case, distances between atoms shorter than 70% of their van der Waals radii are to be avoid). The RLG algorithm keeps both bond length and bond angles fixed and rebuilds the loop by avoiding collision at each node along the road map (further information can be found in references[99,100]). The RLG algorithm has been tested on several systems such as the endo-β-1,4-xylanase protein and has been proved to give good sampling of the loop.

The RTT method incrementally grows a random tree rooted at the initial conformation that explores the reachable conformational space and finds a feasible path to connect the goal conformation. The RTT algorithm is also coupled with elastic network normal mode analysis[103] or EN-NMA. This method drastically reduces the number of *dof* to explore. The search space of the RTT algorithm does not lie in the molecular conformational space of all the *dof* (i.e. the torsion angles), but only in the phase space of the low frequency normal modes from the EN-NMA. Vibrational modes given by the EN-NMA are only valid around the initial conformation and the RTT search would not be accurate when exploring larger regions. So the EN-NMA has to be regularly updated during the conformational change to generate the correct low-frequency vibrational modes. The RTT and RLG algorithms use connectivity matrices to solve the dependencies of the end base of the arm with respect to the joints (here the moving joints are the dihedral angles).

Another algorithm from the robotics field has been proved to be efficient in solving the chain closure problem. It is referred to as the *cyclic coordinate descent* CCD[104]. The CCD is a loop builder algorithm where the loop is built in such a way that the three backbone atoms (N, C_α and C) of the last loop residue (i.e. C anchor) are superimposed with the goal conformation (see figure 4.4).

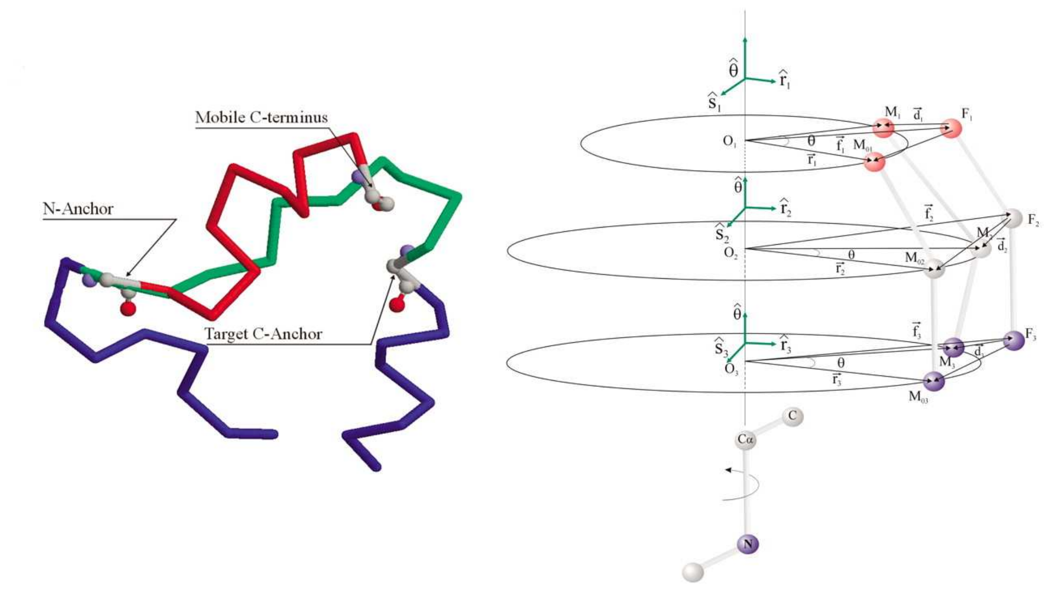


Figure 4.4: Representation of the anchors and the vectors for the CCD algorithm[104].

As shown in figure 4.4, $\overrightarrow{f_1}$, $\overrightarrow{f_2}$, and $\overrightarrow{f_3}$ are vectors that represent the fixed target positions for the atoms of the C-terminal residue of the loop. The positions of the moving C-terminal residue atoms are represented by M_{01}, M_{02}, M_{03}, and M_1, M_2, M_3, before and after a change, respectively, in a dihedral angle of any residue in the loop. The rotation axis (containing O_1, O_2, O_3) is given by the direction of the bond corresponding to the dihedral angle that is modified (N-Cα for ϕ , Cα-C for ψ), where O_1, O_2, and O_3 are the footpoints of vectors from the rotation axis to the three atoms of the moving C-terminal anchor. The CCD rebuilds the loop by iteratively changing the random values of the dihedral angles ϕ and ψ of the backbone chain until the loop is closed. Closing the loop means minimising the distance S:

$$S = |\overrightarrow{F_1 M_1}|^2 + |\overrightarrow{F_2 M_2}|^2 + |\overrightarrow{F_3 M_3}|^2 \tag{4.1}$$

A Ramachandran map for the different rotamers of the angles ϕ and ψ of the chain is built so a constraint can be added to the system. For each residue of the loop, the angle ϕ is built by solving equation 4.1 (for more details on how to solve this equation see reference[104]) and then the new angle ψ is build according to the Ramachandran map. The new ϕ, ψ pair is then accepted with a probability of 1 if the new pair is more probable, or a probability of p_{new}/p_{old} if the new pair is less probable then the old one in the Ramachandran map. However, the literature quotes the Ramachandran mapping to have no noticeable effect on the closure of the loop[104]. One extension to the CCD algorithm is the full cyclic coordinate descent or FCCD by Boomsman *et al*[86]. This method uses both bond angles and dihedrals to solve the loop problem, but instead of considering the whole atomistic chain, the algorithm is computed between the C_α. The distance between two C_α is kept fixed at 3.8 Å, and instead of rotating the end anchor around an axis, the C_αs are used as centre of rotation. The end tail anchor, is also made of three consecutive C_αs, rather than three consecutive atoms. This is the only difference between the CCD and the FCCD. They work in a very similar fashion, changing every *dof* along the chain so that the distance S between the goal and the tail anchors is minimised. One disadvantage of the CCD methods is to induce large changes in the pseudo angles at the start of the loop and small ones at the end. The FCCD algorithm has the possibility to perform the pivot selection in a random fashion (choosing randomly which pair of angles ϕ, ψ is used to minimise S), so that the difference in the value of the changes in the pseudo angles is not localised at the beginning of the chain.

One other algorithm called the wriggling[92,93] uses a concerted motion and some geometrical properties of vectors to ”wriggle” four dihedral at the same time in a protein backbone. The ”wriggling” relies on the fact that for four vectors $\overrightarrow{v}_1$, $\overrightarrow{v}_2$, $\overrightarrow{v}_3$, $\overrightarrow{v}_4$in the three dimensional space there is a linear combination of these four vectors, whose sum is equal to zero:

$$\sum_{i=1}^{4} x_i \overrightarrow{v}_i = 0 \tag{4.2}$$

This condition is used to produce a change in the $[-0.0125, 0.0125]$ radian range of four dihedrals, in such way that the change remains local (for more details see references[92,93]). This method has been tested using at 0 K, to see if it could fold a protein with more efficiency than the standard thrashing method. It is not clear due to the temperature (the test is hence a minimisation and not a simulation) and the energy function (a linear correlation between the energy and the RMSD between the simulated protein and folded structure) that the wriggling is much more efficient that thrashing or other concerted rotation algorithms.

Since all of the above algorithms do not really sample the phase space of a protein loop, but rather build a loop conformation that avoids clashes and links both ends of the loop. No energetic criterion is considered, and the new conformation of the loop is never tested according to a Boltzmann distribution. Choice has been made to focus on other types of concerted rotation that respect detailed balance, specifically the CONROT, CRA and gaussian Bias methods that will be described below.

4.2.2 Boltzmann weighted algorithms

The CONROT move performs local moves along a protein backbone by changing dihedral angles in a concerted fashion. First, a driver angle called ϕ_0 of a randomly chosen atom from all the coordinates is changed by a (random or not) known small amount. Then a rearrangement of a minimum number of neighbours is performed, keeping the preceding and the following atoms in the chain fixed. In moving the atoms in the neighbourhood of the driver angle, both bond lengths and angles remain unchanged, and thus the only degree of freedom allowed to move are the torsion angles. This kind of change must be done using internal coordinates. The chain is then closed satisfying the constraints needed to keep the ends fixed. The use of the driver angle, and the geometry of the system give us that 7 dihedrals $(\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)$ have to be changed to perform a move (see reference[82] for more details).

The peptide bond dihedral is kept fixed, so a minimum of nine atoms are necessary to compute a concerted rotation move. The set of values for the dihedrals can be expressed in the frame of reference of the first atom of the local chain (for details on frames of references see[75])and we can now turn to the problem of incorporating the concerted rotation move as an elementary move within a MC algorithm. The conditions to close the chain are expressed as a function of the first dihedral, and the equation $f(\phi_1) = 0$ is analytically solved. All the solutions for the forward N^n move are computed, one is randomly chosen and the reverse solutions for the move N^m are computed. To preserve the metric weight after the chain closure, the jacobian $J = |\frac{1}{detA}|$ (where A is a geometric dependent matrix, with r_5 the constraint geometric vector, u_6 the constraint unit vector, γ_6 the constraint Euler angle vector and $\boldsymbol{e}_1$ the unit vector[*]) is computed.

$$\mathbf{A} = \begin{vmatrix} \frac{\partial \boldsymbol{r}_5}{\partial \phi_1} & \frac{\partial \boldsymbol{r}_5}{\partial \phi_2} & \frac{\partial \boldsymbol{r}_5}{\partial \phi_3} & \frac{\partial \boldsymbol{r}_5}{\partial \phi_4} & \frac{\partial \boldsymbol{r}_5}{\partial \phi_5} & \frac{\partial \boldsymbol{r}_5}{\partial \phi_6} \\ \frac{\partial \boldsymbol{u_6}}{\partial \phi_1} \cdot \boldsymbol{e_1} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_2} \cdot \boldsymbol{e_1} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_3} \cdot \boldsymbol{e_1} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_4} \cdot \boldsymbol{e_1} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_5} \cdot \boldsymbol{e_1} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_6} \cdot \boldsymbol{e_1} \\ \frac{\partial \boldsymbol{u_6}}{\partial \phi_1} \cdot \boldsymbol{e_2} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_2} \cdot \boldsymbol{e_2} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_3} \cdot \boldsymbol{e_2} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_4} \cdot \boldsymbol{e_2} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_5} \cdot \boldsymbol{e_2} & \frac{\partial \boldsymbol{u_6}}{\partial \phi_6} \cdot \boldsymbol{e_2} \\ \frac{\partial \gamma_6}{\partial \phi_1} & \frac{\partial \gamma_6}{\partial \phi_2} & \frac{\partial \gamma_6}{\partial \phi_3} & \frac{\partial \gamma_6}{\partial \phi_4} & \frac{\partial \gamma_6}{\partial \phi_5} & \frac{\partial \gamma_6}{\partial \phi_6} \end{vmatrix} \tag{4.3}$$

Then attempted probabilities for the move are calculated:

$$\alpha_n(m \rightarrow n) = 1/N^n \tag{4.4}$$

$$\alpha_m(n \rightarrow m) = 1/N^m \tag{4.5}$$

[*] the values of the vectors $\boldsymbol{r_5}$ and $\boldsymbol{u_6}$ are refered as the vectors $\boldsymbol{s}$ and $\boldsymbol{u}$ respectively in figure 4.6

and the probability to accept the final move is:

$$acc(n \to m) = min\Big[1, \frac{N^m exp(-\mathcal{U}(n)/k_\beta T)J(n)}{N^n exp(-\mathcal{U}(m)/k_\beta T)J(m)}\Big] \tag{4.6}$$

where J^m and J^n are the jacobian for the foward and reverse move respectively.

A method described by Farvin *et al.* makes use of a biased gaussian step in order update the conformational sampling of the protein[90]. Small steps are taken, so that large local deformation cannot take place. For a set of local deformations in the dihedral angles $\delta\bar{\phi} = (\delta\phi_1, ...\delta\phi_n)$ a conformation-dependent $n \times n$ matrix called $\boldsymbol{G}$ is introduced. The matrix $\boldsymbol{G}$ has to fulfil the condition that:

$$\delta\bar{\phi}^T \boldsymbol{G} \delta\bar{\phi} \approx 0 \tag{4.7}$$

The steps $\delta\bar{\phi}$ are then drawn from a gaussian distribution:

$$P(\delta\bar{\phi}) \propto exp\Big[-\frac{a}{2}\delta\bar{\phi}^T(\mathbf{1} + b\boldsymbol{G})\delta\bar{\phi}\Big] \tag{4.8}$$

where a and b are tunable parameters. The parameter b controls the force of the gaussian bias whereas the parameter a controls the acceptance rate. For large b, the bias is really strong, and disappears in the limit $b \to 0$. The probability of the attempted move is:

$$W(\delta\bar{\phi}' \to \delta\bar{\phi}) = \frac{det(\frac{a}{2}(\mathbf{1} + b\boldsymbol{G}))}{\pi^3} exp[-(\delta\bar{\phi}' - \delta\bar{\phi})\boldsymbol{A}(\delta\bar{\phi}' - \delta\bar{\phi})] \tag{4.9}$$

To move from the configuration $\delta\bar{\phi}$ to a new configuration $\delta\bar{\phi}'$ the acceptance test has to be modified so as not to break detail balance. The new acceptance test is now:

$$P_{acc} = \Big(1, \frac{W(\delta\bar{\phi}' \to \delta\bar{\phi})}{W(\delta\bar{\phi}' \to \delta\bar{\phi})} exp[(E' - E)/kT]\Big) \tag{4.10}$$

Where the factor $\frac{W(\delta\bar{\phi}' \to \delta\bar{\phi})}{W(\delta\bar{\phi}' \to \delta\bar{\phi})}$ is the bias of the move necessary to keep the detail balance criterion.

This move uses the Gausssian bias to move a set of diherdals. Such move is faster than CONROT moves as:

- the reverse move does not need to be performed to compute the acceptance test (see below)
- no chain closure is performed.

The concerted rotation with angle algorithm (CRA) performs local moves along the protein backbone. The CRA move involves two steps. The first is a prerotation move using a gaussian bias on all the degrees of freedom (both bond and dihedral angles, in blue in figure 4.5) followed by a chain closure move (in red figure 4.5). Both ends of the chain remain fixed to keep the move local (in black in figure 4.5). Mathematical details can be found in reference[83]. The derivatives of the cartesian coordinates of the atom **a** with

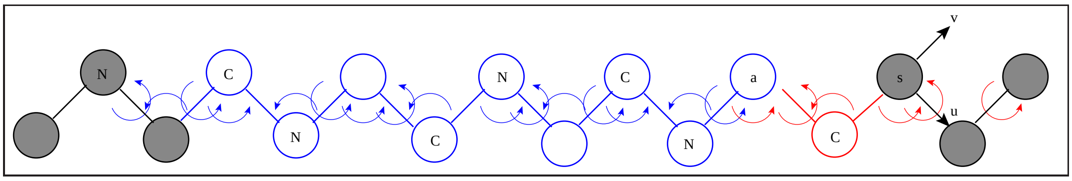


Figure 4.5: Scheme of the Concerted Rotation with Angle move

respect to the n degrees of freedom (dof) are calculated to build a $n \times 3$ matrix. Then this matrix is squared to obtain the $n \times n$ matrix $\boldsymbol{I}$:

$$\boldsymbol{I}_{ij} = \left(\frac{\partial \boldsymbol{a}}{\partial \phi_i} . \frac{\partial \boldsymbol{a}}{\partial \phi_j} \right) \tag{4.11}$$

Then the matrix $\boldsymbol{J} = c_1(\mathbf{1} + c_2(\boldsymbol{I} \times \boldsymbol{I}))$ is calculated (were $\mathbf{1}$ is the identity matrix). The parameters c_1 and c_2 control respectively the acceptance rate and the force of the bias. The bias aims to minimise the displacement of the atom **a** such that : $d^2 = (\delta \boldsymbol{a})^2$. The Cholesky decomposition of the matrix $\boldsymbol{J}$ is used to calculate the matrix $\boldsymbol{L}$. Then a set of n random numbers $\delta \boldsymbol{\chi}$

following a gaussian distribution are used to solve the equation:

$$\delta\boldsymbol{\chi} = \boldsymbol{L}^t \delta\phi. \tag{4.12}$$

where the vector $\delta\phi$ represents the changes in the *dof*. The random gaussian vector $\delta\boldsymbol{\chi}$ is built in such a way that the displacement $\boldsymbol{d}$ of the vector $\boldsymbol{a}$ is minimised:

$$\delta\boldsymbol{\chi}^t \delta\boldsymbol{\chi} = d^2. \tag{4.13}$$

Then the new conformation is built, using the new *dof* and the new matrices $\boldsymbol{I'J'L'}$ are recomputed. Using the linear transformation:

$$\delta\boldsymbol{\chi'} = \boldsymbol{L'}^t \delta\phi. \tag{4.14}$$

the values of $\delta\boldsymbol{\chi}'$ and $d^2 = \delta\boldsymbol{\chi}'^t \delta\boldsymbol{\chi}'$ for the reverse move are calculated and the biasing probability for both forward and reverse move can be expressed:

$$P(a \to b) = (det\boldsymbol{L})e^{-d} \tag{4.15}$$

$$P(b \to a) = (det\boldsymbol{L'})e^{-d'} \tag{4.16}$$

The matrix $\boldsymbol{L}$ is a lower triangular matrix so its determinant can be easily calculated by :

$$det\boldsymbol{L} = \prod_i L_{ii} \tag{4.17}$$

In the original reference[83], moves are limited to 9 dihedral angles, but nothing stops the move from being longer or shorter, as the method can in principle work with any number of *dof*.

Once the prerotation move is complete, the second part of the move is computed. A scheme of the notation used in the chain closure can be found in figure 4.6. To close the chain, several constraints have to be respected. The position of the last atom **s** and the orientation of the vectors **u** and **v** have to be kept fixed which gives us 3 constraints for the condition on the atom **s** and 3 other constraints on the vectors **u** and **v** (for more detail see reference

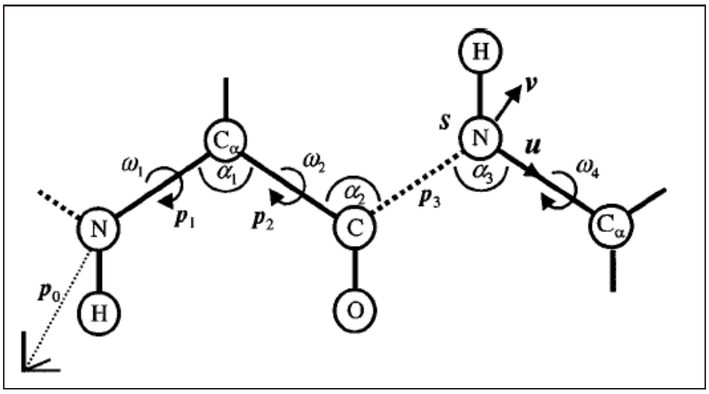


Figure 4.6: Scheme of the Chain Closure algorithm[83]

[83]). To be able to solve the chain closure and to satisfy the set of constraint, 6 *dof* have to be moved. Three dihedrals (peptide bond dihedral being kept fixed) and three angles are moved to solve the geometric problem. Using 3×3 matrices to perform rotations along bond ($\boldsymbol{T}$) and dihedral ($\boldsymbol{R}$) angles and to change of frame of reference the equation below has to be solved:

$$\boldsymbol{R}_3^{-1}\boldsymbol{T}_2^{-1}\boldsymbol{R}_2^{-1}\boldsymbol{T}_1^{-1}\boldsymbol{R}_1^{-1}\boldsymbol{T}_0^{-1}\boldsymbol{u} = \begin{pmatrix} \cos\alpha_3 \\ sin\alpha_3 \\ 0 \end{pmatrix} \tag{4.18}$$

Using the change in frame of reference we can now express each *dof* as a function of the first dihedral ω_1.

We use the matrices corresponding to the rotation along the bond angle α_i and the rotation along the dihedral ω_i, respectively $\boldsymbol{T_i}$ and $\boldsymbol{R_i}$, (for more details about frames of reference see the Nobel Price lecture by Flory[75]) which are defined as:

$$\mathbf{T}_i = \begin{pmatrix} \cos\alpha_i & -\sin\alpha_i & 0 \\ \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.19}$$

$$\mathbf{R}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_i & \sin\omega_i \\ 0 & -\sin\omega_i & \cos\omega_i \end{pmatrix} \tag{4.20}$$

So the z-axis component of the right end side of the equation 4.18 is equal to zero. So the left hand side can be numerically solved as a one unknown equation, $G(\omega_1) = 0$ and the others *dof* can be calculated with respect to ω_1.

The equation 4.18 has only two branches instead of 4 for the CONROT, and there is no need to perform a reverse move anymore (this is due to the mathematical construction of the move). This method is currently about four times faster than the original CONROT (for more details see references[82,83]) in terms of speed for the closure of the chain.

Many other algorithms and methods that satisfy detailed balance are available for sampling proteins such as LmProt[91] or the Parrot[102] algorithm. To be able to sample loop motions with a good efficiency, the gaussian Bias and the CRA methods have been investigated, and implemented in an existing molecular modelling package. Both methods satisfy detailed balance and hence can be used to perform MC simulations. Some applications of these concerted motions can be found in the section below.

4.3 Applications for proteins

The CONROT method can be used either for folding[105,106] or for energetic studies[30,107]. In the case of protein folding, good agreement with experimental data (NMR) has been found even for small cyclic peptides. In this case, the use of MC moves aims to lead to the true, cis/trans population of the amide bond. In five different peptides, the MC simulations lead to the same configurations as the experimental data (even with a boat like configuration leading to a cis-trans-cis-trans sequence). The CONROT method has also been used in the investigation of nucleic acid and small protein folding[106].

The efficiency has been calculated using the formula below:

$$s_w = \lim_{h\to 0} \frac{n\sigma^2\langle (A)\rangle_n}{n\sigma^2(A)} \tag{4.21}$$

where A is the observable value (the energy, the Ramachandran angles or any other physical property), $\sigma^2(A)$ the variance, and $\sigma^2\langle (A)\rangle$ is the variance average for windows of length n. The algorithm has been used on a small protein (65 residues), the chymotrypsin inhibitor 2 (CI2) and a 12 nucleotide ribosomal RNA hairpin whose sequence was (GGGCGAAAGCCU)[106]. The results show the efficiency of CONROT moves to sample all the phase space for the protein with a reduction in computational time. The simulations on the nucleic acid lead to the same conclusion; good efficiency (close to the result obtained with MD simulations) and a reduction in computational time.

CONROT has also been used to sample phase space for free energy perturbation studies[107]. The algorithm can perform better sampling of the phase space and hence obtain more precise free energies during the simulation. This enhanced sampling leads to very efficient results with small calculated standard errors. This study shows the efficiency in using the CONROT algorithm for the investigation of the binding free energy of a host-guest system[107]. The relative binding free energy of three amino acids for macro-bicycle 12 in chloroform were calculated. The efficiency of CONROT moves to perform large conformational changes in the hydrocarbon segments allowed accurate sampling of the host, and lead to free energy values close to experiment.

The CRA algorithm, despite being a quite recent method, has been used efficiently with both proteins and nucleic acids[84,108]. This method has been first tested by calculating the average dihedral step size per local move[84,85]:

$$|s_w| = \sqrt{\frac{1}{n_\omega}\sum_{i=1}^{n_\omega}(\delta\omega_i)^2} \tag{4.22}$$

and the statistical efficiency s_w of the sampling of the main chain dihedral angles

$$s_w = \lim_{n \to 0} \frac{n\sigma\langle(A)\rangle_n}{n\sigma(A)} \tag{4.23}$$

where $A = \cos\omega_i$ is the observable main-chain dihedral angle. The systems used for the test were tetradeca-alanine and a 36-residue peptide taken from the villin headpiece sub domain.

Both simulations were performed using OPLS-AA force field in vacuum at 30 °C . A more realistic series of runs were performed in an implicit solvent model (GB/SA) using the (Ala)$_8$, (Ala)$_{10}$, (Ala)$_{12}$ as benchmarks for the algorithm. The study showed a good agreement between the CRA and the preceding studies using both MD and MC. Another study on a small system, β-Hairpin U(1-17)T9D derived from a globular protein, shows the efficiency of this method. The study shows a clear relationship between the number of H-bonds, RMSD of the backbone and the energy. The conformation of the low temperature converged structure was close to the NMR determined conformation.

The CRA algorithm has also been used in the folding of nucleic acids[84]. As in the studies of proteins, the use of CRA in both vacuum and GB/SA against a modified CONROT or a local update of the main chain torsion angles, showed the efficiency of CRA. CRA allows for more sampling of the main chain configuration than the CONROT algorithm. This is due to the fact that the CRA algorithm is more efficient in sampling all the conformational flexibility of the main chain as both bond and dihedrals angles are changed. The use of the gaussian bias for the final displacement of the prerotation move, also increases the sampling as the method achieves a very good closure rate.

The CRA algorithm has been compared to MD simulation[108] in a protein folding investigation. Thus both methods lead to conformations close to experimental native states for three different peptides and MC simulations tend to be 2-2.5 times faster than MD simulations.

4.4 Concluding remarks

As described in chapters 2 and 3 biological processes involving major reorganisation of protein structures occur in a time scale too long to be sampled using MD. Using MC method to sample such large moves involves the use of specific algorithms. Many algorithms exist that generate loop or random configurations of a polymer or a protein.

Moves used for polymers cannot be used in proteins due to the non-homogeneity of biological systems. Moves inspired from robotics give good results in closing the loop. However, to be used during a MC simulations, moves have to comply with the detailed balance criteria and most of the loop closure algorithms inspired from robotics introduce a bias that cannot be corrected and hence, break detailed balance.

To sample large scale motion of proteins, Boltzmann weighted algorithms need to be used. The CRA algorithm has been described to enhance sampling of protein backbone loop and in several case to be faster than MD methods. Owing the flexibility of of several class of protein, choice has been made to implement it in the ProtoMS package[51] to use it on protein-ligand interaction problems.

Details of the implementation of the CRA in ProtoMS, are described in the next chapter.

Chapter 5

Software Development

This chapter describes the overall work of implementing the CRA algorithm in the ProtoMS[51] package. First a summary of the capabilities of the existing packages will be discussed, then a section on how the CRA has been implemented and then enhanced in the ProtoMS[51] package, will be discussed.

5.1 Existing Monte Carlo simulation package

The ProtoMS[51] package (locally developed in Southampton) does not perform concerted motion moves, whereas the MCPRO package[50] incorporates the concerted rotation with angles algorithm in addition to standard thrashing moves (see 3.3.1 for a description of the thrashing move). However, the MCPRO[50] package is slower than the ProtoMS[51] package, less user friendly, and the CRA algorithm is not modifiable in terms of its parameters or structure. Ideally we would like to have the best of both *i.e.* having a flexible CRA algorithm in the ProtoMS[51] software.

The MCPRO package does not handle the PDB format as input. Instead, a specific tool called *pepz* has to be used[52] to generate a Z-matrix, making the use of MCPRO[50] less intuitive and more fastidious. The user of *pepz* need to know the sequence of the protein. In ProtoMS[51], a pdb file can be used as the standard input file with no need to create the Z-matrix. MCPRO[50] has been used to obtain benchmarks for the tryptophan protein (see figure 5.1). To get the optimum combination of ProtoMS[51] and both the CRA and the gaussian bias algorithms, a good understanding of the code is needed. This understanding has been achieved through the use of the simulations and some small modifications of several routines (see 5.2).

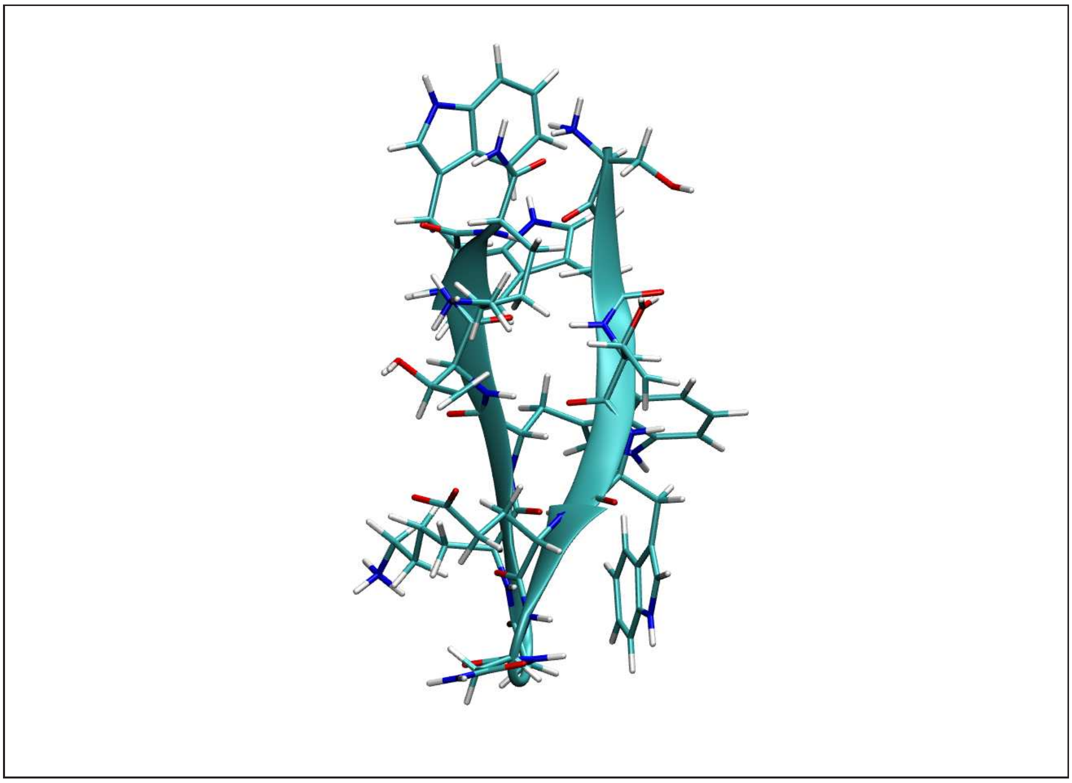


Figure 5.1: 3-D representation of the Tryptophan zipper protein (PDB reference: 1le1)

5.2 Efficiency of existing methods.

Standard MC and MD simulations have been performed on two different systems, the chicken villin protein, and the ala-(14) polypeptide in both linear and α-helical conformations, to get conformational sampling data. To check the implementation of CRA in ProtoMS[51], the tryptophan job from the test section of the CPRO[50] has also been run.

5.2.1 Classical MD and MC simulations.

For each system, a 2.4 ns MD simulation has been performed using the AMBER package and force field[39,109], the SHAKE algorithm[48] to constrain the bonds, at 300 K temperature. The MD simulation has been performed in both vacuum and the GBSA implicit solvent model[67,68]. For both chicken villin protein (PDB reference 1yu8) and the α-helix polypeptide ala-(14) (build using the molden[110] package) the same equilibration process has been used. First, 1000 steps of minimisation were performed, followed by 10 ps of dynamics both with 5 kcal/Å^2 restraints applied to the atoms of the system. Then the same process was repeated with 1 kcal/Å^2 restraints. Then 1000 steps of minimisation and 10 ps dynamics without any restraints ended the equilibration period. The 2.4 ns production trajectory was generated for each of the two systems using randomised velocities. The analysis of specific items of the trajectory was made using the ptraj tool of the AMBER package to compute the RMSD with respect to the first structure of the trajectory.

Monte Carlo simulations have been carried out using the ProtoMS[51] software. Simulations have been run at 298 K and using constant volume and temperature conditions. A first period of equilibration of 5000 MC steps is carried out, and then a 100000 MC step simulation is performed. The average acceptance rates for the MC simulation are displayed table 5.1. Table 5.1 shows that the acceptance rate for the backbone moves are poor (around 2.56% for the polypeptide and 1.1% for the chicken villin protein). The size of the move is between 0 and 2 Å for the translations and between 0 and 0.5 radians for the rotations for the rigid units (see figure 3.1 and section 3.3.1).

Such low rates indicate that the backbone sampling is poor and that most of the phase space sampling is due to side chain moves (in the case of the polypeptide, the moves are actually quite small considering the geometry of the side chain). The acceptance rates for backbone moves are about ten times smaller than the total acceptance rate. An increase of the acceptance

	ala(14)polypeptide		chicken villin	
GB	22.92%	2.64%	14.63%	1.26%
Vacuum	22.76%	2.53%	14.61%	1.13%

Table 5.1: Acceptance rate during the MC simulations for the linear polypeptide and the chicken villin protein. Backbone (blue) and complete molecule (red) in both vacuum and GB.

rate will be possible by allowing smaller amplitude to the move. However, that would lead to smaller sampling, and the computational time needed to sample a given phenomenon would increase dramatically.

If we compare the values of the RMSD with respect to the first structure for both MD and MC, we can clearly see that standard MD is more efficient in terms of sampling than classical MC. The RMSDs are plotted in table 5.2. Value of the RMSD for the MD simulations of the ala(14)polypeptide are two order of magnitude bigger than the RMSD for the MC simulations. Values of the RMSD for the MC simulation of the chicken villin protein are about one third of the value of the RMSD for the MD simulations. The very poor sampling of the ala(14)polypeptide is due to the linear form of the polypeptide. The sampling achieved with the classical MC method clearly shows the need for novel sampling algorithms for protein backbones. A third simulation

	ala(14)polypeptide		chicken villin	
Vacuum	0.07±0.03	5.64±1.00	1.16±0.18	3.04±0.27
GB	0.08±0.03	2.50±0.47	1.08±0.16	3.40±0.92

Table 5.2: RMSD of the backbone for MD (blue) and MC (red) in both vacuum and GB. RMSD are express in Å with standard deviations for blocks of 1000 MC steps given.

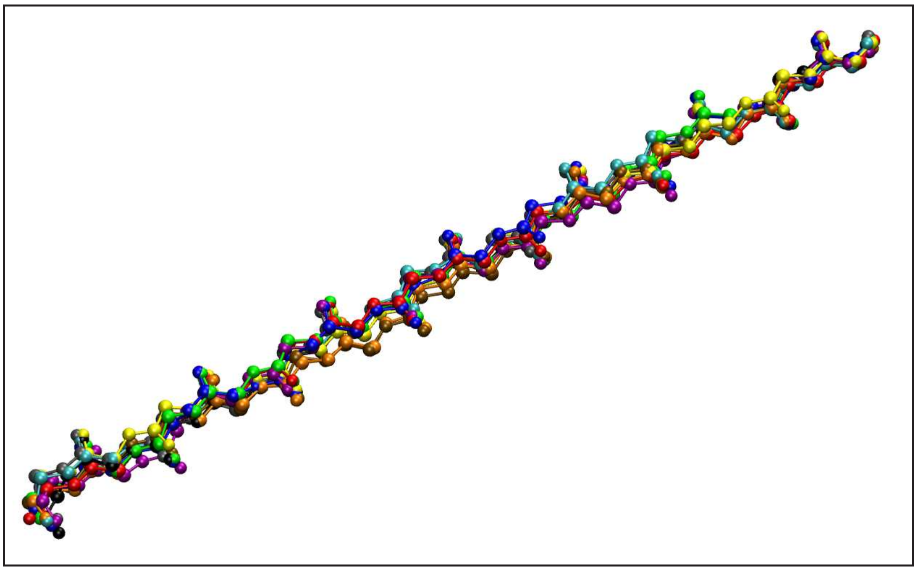


Figure 5.2: Sampling of standard MC moves on a linear ala(14) polypeptide. Simulation have been carried on for 1000000 steps and snapshots taken every 10000 steps. Using rigid unit backbone moves as describe chapter 3.3.1

using a linear ala-(14) polypeptide has been performed. Figure 5.2 shows the superposition of snapshots of the backbone along the MC simulation. This figure clearly shows the inefficiency of the rigid backbone unit to sample large scale moves on proteins, as very little deviation of the backbone geometry occurs.

A good solution would be to have the CRA algorithm implemented into the ProtoMS package[51]. The process about how the existing package has been modified and how the CRA algorithm has been implemented in ProtoMS[51] is described below.

5.3 Code implementation

To enhance the sampling of the protein backbone, the CRA algorithm has been implemented in the ProtoMS[51] package. Then the algorithm has been modified in such a way that the length of the move could be adapted to the biological problem. This scheme gave us several advantages:

- Using the ProtoMS[51] structure allows faster computations and a friendlier interface.
- Using a modified CRA, allow the length of the move to be adapted to the biological problem.

The first step was to implement the original CRA algorithm from the MCPRO[50] package into ProtoMS[51].

5.3.1 Standard CRA into ProtoMS

The CRA code of the MCPRO[50] package has been incorporated almost directly into ProtoMS[51] to model biological targets and to implement the existing code. The original code for the CRA algorithm has been designed to be used according to the reference[83]. The algorithm does not allow concerted rotations to be performed on longer or shorter segment then a nine dihedral segment of the protein backbone. The implementation of the CRA algorithm into the ProtoMS[51] code has been done in several stages.

- The first step was to create a new `movetype` for ProtoMS[51]. New variables have been created and handle the new move, and the probabilities of moves have been reassigned.
- ProtoMS[51] uses rigid backbone unit moves. So, to be able to make the changes in the internal *dof*, routines converting the cartesian coordinates into internal degrees of freedom (bond length, bond and dihedral angles) have been built. Cartesian coordinates of the atoms N, C_α and C are stored, along with the bond lengths and angles.
- CRA moves are performed as described in the reference[83].
- Energy is recomputed, and a new Metropolis Monte Carlo test is performed including the bias of the prerotation move, and the Jacobian for the chain closure.

The rebuilding of the protein and the way coordinates are stored in a stack pile, have been modified in ProtoMS[51] to manage the number of residues involved in the concerted motion. The user can choose to perform either the whole simulation using CRA, or to mix CRA with the standard moves already available in ProtoMS. The use of the CRA move can be restricted to a specific region of the protein (to sample only a specific loop for instance). This allows greater flexibility to perform more precise simulations.

To get as close as possible to the code described in the reference[83], only the first part of the move is performed on the first or last three residues of the protein (*i.e.* only the gaussian bias[90]). This implementation allows a complete sampling of the system, whereas the CRA algorithm by construction (both ends of the rotated chain being kept fixed), cannot move the first and the last residues of the protein and hence, folding would not be observed. From this point, CRA moves will refer to moves as described in the literature[83]. A complete scheme of the software design is presented figure 5.3 with the blue square representing the implementation at this stage. So the ProtoMS[51] package can run several moves from the same input:

- Standard ProtoMS[51] moves using the rigid unit backbone moves.
- CRA moves as described in reference[83] using a gaussian bias without chain closure for both ends of the protein. The CRA moves can be restricted to a specific region of the protein.

The results obtained in developing such moves for the ProtoMS[51] package are described below.

5.3.2 Standard CRA in ProtoMS: results

4 million step MC simulations have been run in implicit GBSA solvent on both ala-(14) polypeptide and chicken villin headpiece protein to test the efficiency of the CRA move implemented in ProtoMS[51]. Different ratios of CRA move have been tried: first a ratio of one CRA move every four standard moves (1/4 green and black curves), and then a ratio of one for two (1/2 red curve), standard moves being backbone, residue, and side chains moves described in ProtoMS[51].

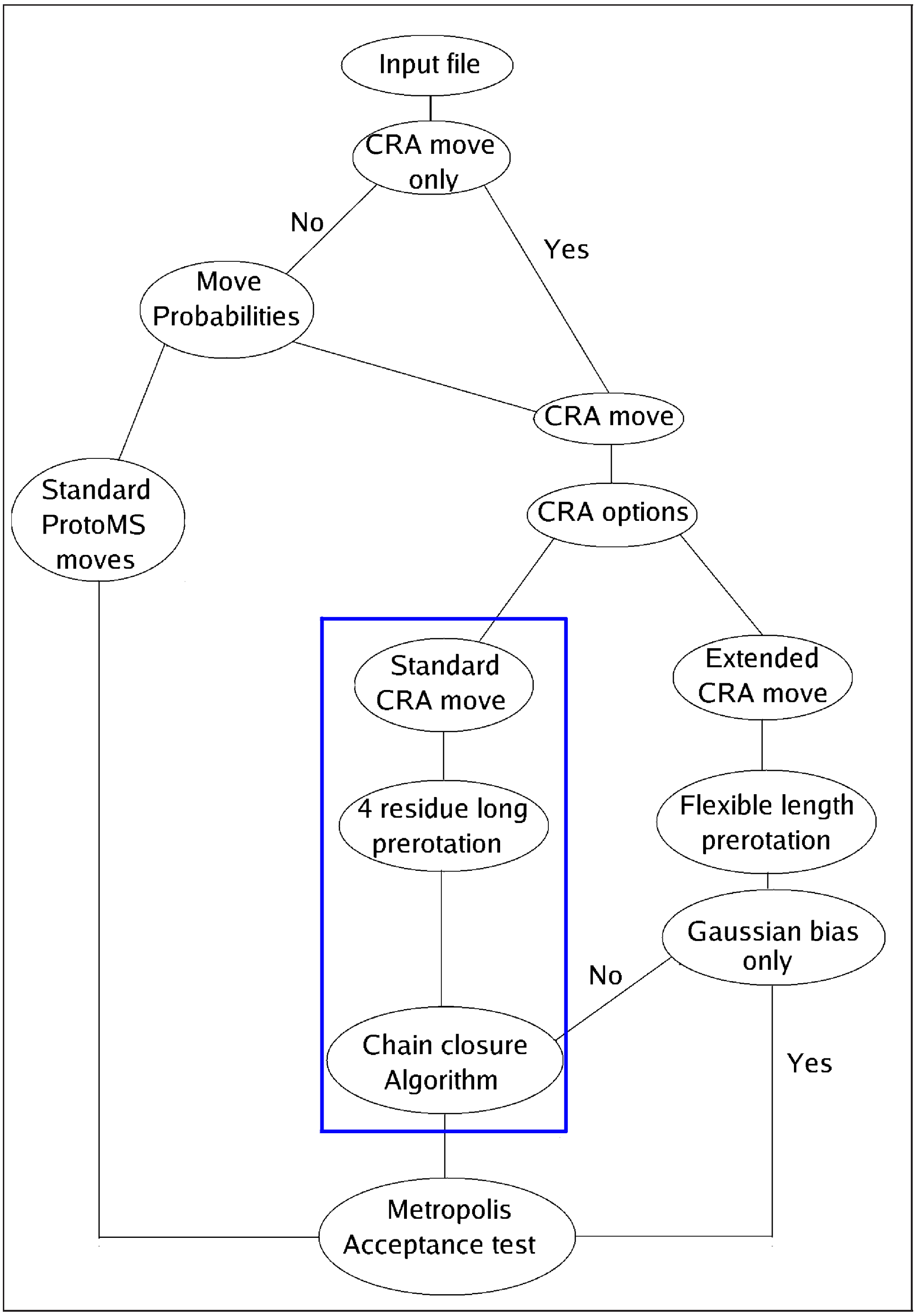


Figure 5.3: Scheme of the CRA implementation in ProtoMS. Both CRA and standard moves can be performed as the same time, the length of the prerotation move can be chosen, and a CRA only option can be used to perform only CRA moves.

While the use of CRA enhances the sampling (see figures 5.4 red and green curve, for enhanced sampling of the polypeptide), both simulations were unable to fold the protein. The RMSDs (with standard deviations) with respect to the original structure are bigger than using standard moves. RMSD for the simulation using a ratio of one CRA move for 4 standard move is 2.85±0.63Å for the backbone only, and the RMSD for the simulation where the ratio is one for two is 2.46 ± 0.68Å. In both case, the RMSD is of the same order of magnitude as the RMSD from the MD simulation. RMSDs are computed using the structures of the snapshots, obtained every 1000 steps.

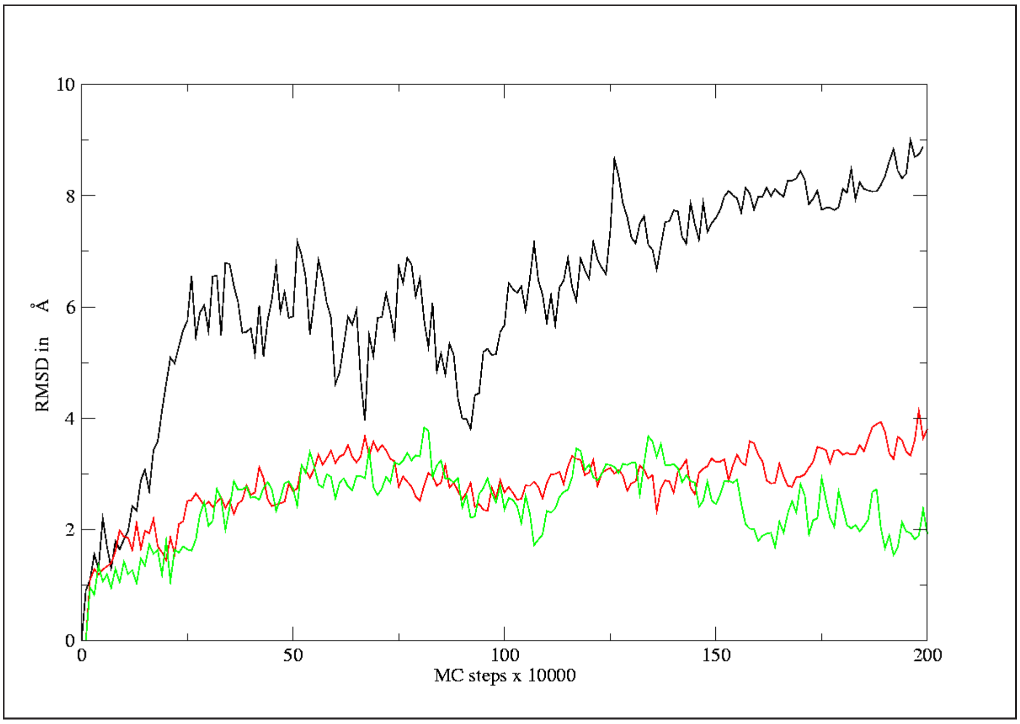


Figure 5.4: RMSD of the ala(14) polypeptide using standard and gaussian implementation of CRA moves in Å. The red and green curve for the RMSD are the RMSD obtained with CRA moves only. The black curve is the RMSD obtained using CRA move and the gaussian bias for the ends of the protein (see chapter 5.3.3. RMSD for the black curve is obtained after super-imposition of the structures.

So far the ProtoMS[51] package has been implemented with the CRA as described in original reference[83]. This implementation allows the gaussian bias move on both ends of the protein (changing both bond and dihedral angles) to be performed. The end move is not part of the sub-routine from MCPRO[50] but suc move is performed in MCPRO[50]. It has been implement in ProtoMS[51] in new routines. Details on the implementation of the end move (gaussian bias) are described below.

5.3.3 Implementation of the gaussian bias in ProtoMS

Implementing the gaussian bias in the ProtoMS[51] package aims to two goal. First allowing end move for the protein and to later implement a extended prerotation move for the CRA algorithm. To implement the gaussian bias into the source code of ProtoMS[51], the concept of frame of reference described by Flory[75] has been investigated and applied to calculate the matrix $\boldsymbol{I}$ and to the rebuilding of the chain.

We have first attempted to get the derivatives of the vector $\mathbf{a}$ (as described in[83]) using the change of frames of reference. This change of frame of reference allows us to describe a bond vector $\boldsymbol{p_i}$ whose coordinates in the frame of reference i are $\begin{pmatrix} p_i \\ 0 \\ 0 \end{pmatrix}$, in the frame of reference (i_{-1}) by using two rotations along the z and x axes. The matrices corresponding to the rotation along the bond angle α_i and the rotation along the dihedral ω_i are respectively $\boldsymbol{T_i}$ and $\boldsymbol{R_i}$ (for more details see[75]) which can be defined as:

$$\mathbf{T}_i = \begin{pmatrix} \cos\alpha_i & -\sin\alpha_i & 0 \\ \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

$$\mathbf{R}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_i & \sin\omega_i \\ 0 & -\sin\omega_i & \cos\omega_i \end{pmatrix} \tag{5.2}$$

So the total transformation matrix changing the coordinates from the frame of reference i to the frame of reference $i-1$ can be expressed as follows:

$$\begin{pmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\sin\alpha_i\cos\omega_i & \cos\alpha_i\cos\omega_i & \sin\omega_i \\ \sin\alpha_i\sin\omega_i & -\cos\alpha_i\sin\omega_i & \cos\omega_i \end{pmatrix} \tag{5.3}$$

The vector of the coordinates of an atom $\mathbf{r}_i$ can then be expressed in the preceding frame of reference by using the relation

$$\mathbf{r}_{i-1} = \mathbf{T}_{i-1}\mathbf{R}_i\mathbf{r}_i + \mathbf{p}_{i-1} \tag{5.4}$$

So the coordinates of the last atom of a chain can be expressed in the frame of references of the first atom of the chain and then using the transformation matrix $\mathbf{M}_{lab}$ the coordinates can be expressed in the laboratory frame of reference (generally a cartesian space frame) by using a product of matrices. These matrices only depend of one degree of freedom and so we can differentiate the cartesian coordinates of the last vector, with respect to the degrees of freedom.

To test the routines responsible of change of cartesian coordinates, an initial algorithm was coded, fully independent of the CRA algorithm and the ProtoMS[51] package, in which a chain of atoms is built when the values of the bond and dihedral angles are used as input. The first atom has cartesian coordinates of $(0,0,0)$ and the second of $(0,0,l)$ where l is the length between the two atoms (length that for testing purposes is the same for all the atoms of the chain). To test this first step, the values of the coordinates of the resulting chain have been compared to the coordinates obtained from ProtoMS[51]. The use of such a frame of reference as the one described in the reference[75] involves changing the value of the dihedral angle n from ϕ_n to $\phi_n + \pi$, when the value of the bond angle $n-1$ is greater than π. Then the derivatives of the coordinates of the last atom with respect to the degrees of freedom are computed. To test the derivatives of the position of $\mathbf{a}$ with respect to the degrees of freedom using the matrices $\boldsymbol{T}$ and $\boldsymbol{R}$ from the reference[75], the numerical approximation of the derivative is calculated using ProtoMS[51] by changing the value of one *dof* at each step. The formula for the numerical approximation of the derivative of a function $f(x)$ for the value of $x = \phi_i$ is:

$$\lim_{h \to 0} \frac{f(\phi_i + h) - f(\phi)}{h} \tag{5.5}$$

So computing the values of the coordinates for five dihedrals ϕ and using the formula 5.5 we can compute an approximation of the values of the derivatives. Theses values are compared to the values of $\frac{\partial \boldsymbol{a}}{\partial \phi_i}$ using the analytical results. Then, in order to increase the accuracy of the results the numerical approximation of the derivative is computed using :

$$\lim_{h \to 0} \frac{f(\phi_i + h) - f(\phi_i - h)}{2h} \tag{5.6}$$

Results of the numerical approximation and derivative method are show in table 5.3. Table 5.3 shows that the values of the derivatives of the coordinates

	X**a**		Y**a**		Z**a**	
value of the *dof*	$\frac{\partial \boldsymbol{a}}{\partial \phi_i}$	Num Approx	$\frac{\partial \boldsymbol{a}}{\partial \phi_i}$	Num Approx	$\frac{\partial \boldsymbol{a}}{\partial \phi_i}$	Num Approx
39	-1.389	-1.375	0.000	0.000	-0.611	-0.630
49	2.031	2.062	1.271	1.260	1.154	1.432
110	0.355	0.401	1.123	1.416	1.350	1.318
-110	-0.156	-0.115	1.039	1.033	-0.823	-0.859
153	-0356	-0.344	-0.400	-0.401	-0.288	-0.286
53	1.259	1.318	-1.620	-1.604	0.447	0.458
90	-0.248	-0.229	-0.786	-0.802	-0.946	-0.974

Table 5.3: Numerical approximation of $\frac{\partial \boldsymbol{a}}{\partial \phi_i}$. Where X**a**, Y**a**, Z**a** are the values of the derivatives of the vector **a** along the axes respectively X, Y, Z.

of the atom **a** with respect to the degrees of freedom computed with the use of Flory's frame of references[75] are close to the numerical approximation using a value of h of 1°.

Once the derivatives are computed, the matrices $\boldsymbol{I}$, $\boldsymbol{J}$, $\boldsymbol{L}$ as defined in the reference[83] are built and the n gaussian numbers are randomly chosen, and following the process described in section 4.2.2, the biasing probability is then calculated for both forward and reverse moves. Each routine has been tested separatly using simple matrices and the results have been double-checked by hand. Once every routine has been tested and shown not to be faulty, the complete prerotation move has been tested by computing 1 million steps of Monte Carlo simulation. This test has been performed outside the ProtoMS[51] package, with no energy function so only geometric changes were considered. The distribution of both angles and dihedrals has been plotted by increments in bins of 5 degrees and compared to the distribution obtained by using the CRA algorithm from MCPRO[83] incorporated in ProtoMS[51] under the same conditions (no energetic function). The test system is a linear "phantom" chain 6 dihedrals long. These results were obtained using the same parameters c_1 and c_2 which control the acceptance rate and the size of the bias described in reference[83] for both simulations. Simulations have been repeated twice. Figure 5.5 shows that for the two sets of data, the dihedral angels are equally distributed between $-\pi$ and π. The standard deviation of the distribution are for both methods of the same order of magnitude and this shows that the two methods are not significantly different. The prerotation move using a gaussian bias can then be implemented in ProtoMS[51].

Once the gaussian bias has been implemented in ProtoMS[51], it is used to move the first two and the last three residues of the protein, so the CRA moves in ProtoMS[51] are now implemented as described in the original paper[83]. RMSD for the simulation using the gaussian bias move for protein ends is shown figure 5.4. If we compare the value of the RMSD (black curve in 5.4) with the ones without the gaussian bias move for ends, it becomes obvious that this implementation provides the necessary tool for protein folding.

The CRA algorithm has then been tested by comparing the tryptophan (see figure 5.1) test job in MCPRO[50] with the equivalent simulation using

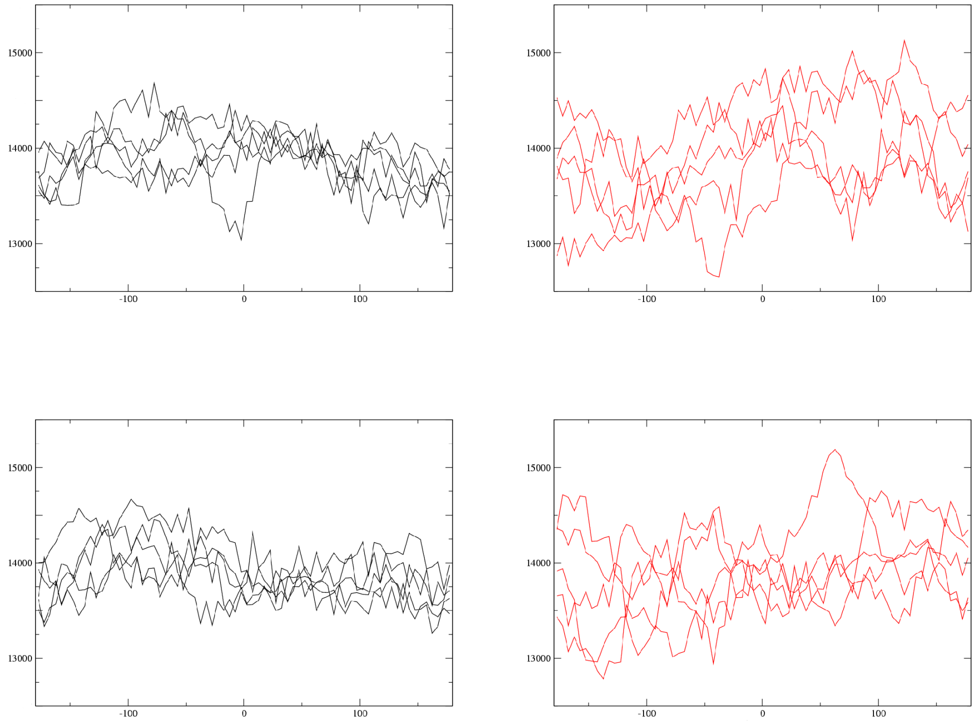


Figure 5.5: Distribution of the 5 dihedrals used in the prerotation move. The red curves represent the distribution obtained using the original CRA algorithm incorporates in ProtoMS[51]. The black curves represent the distribution obtained using the frame of references describe by Flory[75] used to implement the gaussian bias move for protein ends. All set of data were obtained during a 10 millions steps MC simulation with no energetic potential.

ProtoMS[51].

5.3.4 Gaussian bias implementation results

The extended tryptophan zipper protein has been used as starting configuration for 2.5 million MC steps in GBSA, using both the ProtoMS[51] and the MCPRO[50] packages. For the simulation run with ProtoMS[51], the average acceptance rate for the tryptophan protein is $8.72 \pm 4.87\%$ whereas the total acceptance rate for CRA moves is $7.57 \pm 3.57\%$ (average over blocks of 50000

MC steps). RMSD of both package with respect to both the folded and unfolded structure is represented figure 5.6. Figure 5.6 shows that in both cases, the RMSD with respect to the folded state gets smaller whereas the RMSD with respect to the extended state gets bigger. This shows that the change in the structure is toward the folded state. Not only do both simulations achieve the same range of deviation of within 1 Å, but the overall shapes of the curve with respect to the folded state are similar. So the CRA implemented in the ProtoMS[51] package leads to the same results as the CRA in the MCPRO package.

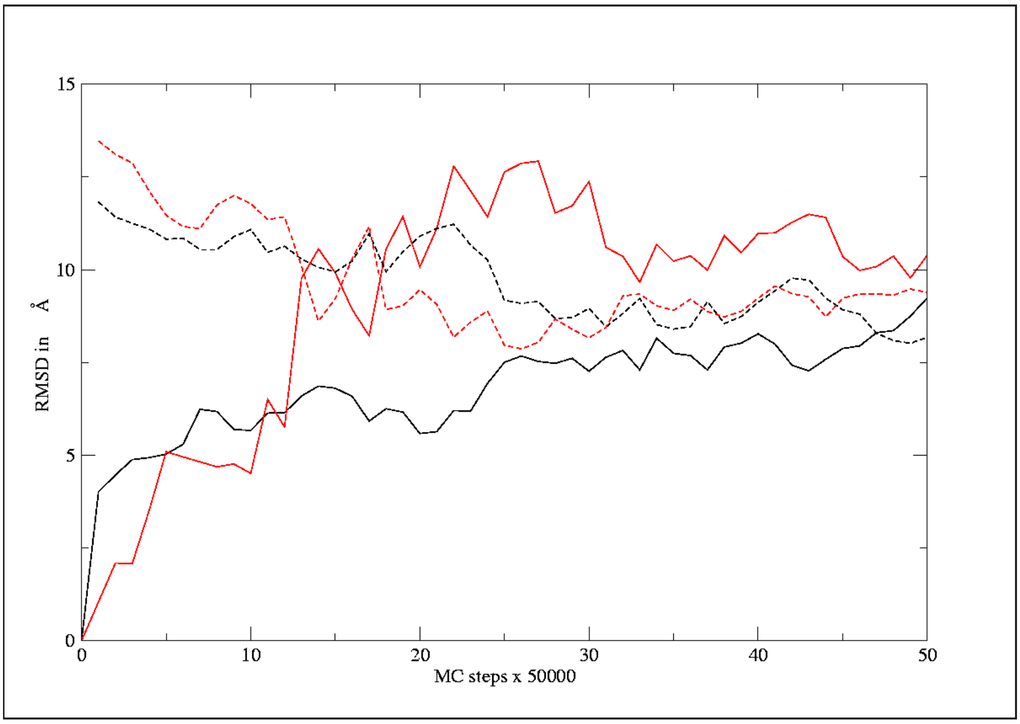


Figure 5.6: RMSD of the tryptophan zipper protein (PDB code 1le1) with respect to the initial structure (solid) and the folded structure (dash). Black curves are obtained using the MCPRO[50] package, the red using the ProtoMS[51] package.

Table 5.4 shows the values of the RMSD between the folded NMR structure and the last configuration of both simulations. Values of the RMSD are close to each other, showing similar behaviour from both packages.

The final structure of the 2.5 million MC step simulation is shown in

	Backbone	Heavy atoms	All atoms
ProtoMS	6.6	8.5	8.9
MCPRO	6.2	7.6	8.2

Table 5.4: RMSD between the folded structure and the last step of the simulation. Structures are aligned on the NMR structure of the folded protein. RMSDs are expressed in Å.

figure 5.7. The final conformations from both packages have been superimposed with the folded NMR structure.

The CRA algorithm has been implemented in the ProtoMS[51] package as described in the literature[83]. However, the use of the ProtoMS[51] package provide useful features that do not appear in the MCPRO package[50] such as:

- the possibility to mix CRA moves, rigid unit backbone moves, side chain moves
- the possibility to apply only the CRA move to a specific fraction of the protein.

The implementation of the gaussian bias in ProtoMS[51] for protein ends uses an iterative algorithm which means that the gaussian bias can be extended to any number of *dof* or residues. So we have decided to implement it with the chain closure algorithm move, so the CRA could be extended to any length. This implementation will be describe below.

5.3.5 Extended concerted rotation moves

The use of the matrices $\boldsymbol{T}$ and $\boldsymbol{R}$ is slower than the algorithm used in the CRA code to compute the derivatives of $\mathbf{a}$ (using cross product, see reference[93] for some definitions), so the computation of the derivatives of the atom $\mathbf{a}$ have been modified to use the cross product method. However the iterative design is kept so the gaussian bias move can be extended to many degrees of freedom. So the standard ProtoMS[51] package features several implementations:

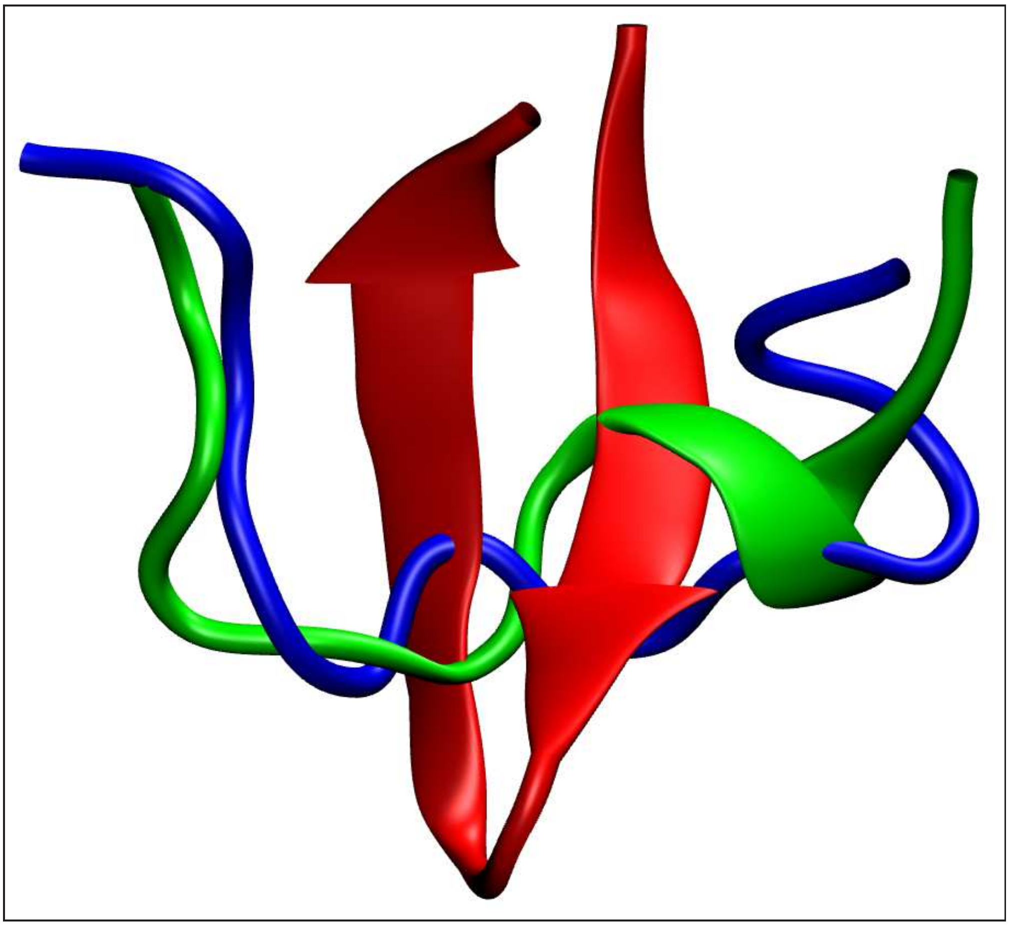


Figure 5.7: Snapshots of the last structure with MCPRO (blue) and ProtoMS (green) for the Tryptophan zipper protein. NMR structure for the unfolded protein is represented in red.

- Standard CRA algorithm from the MCPRO[50] package into the software, moving only 4 residues during the prerotation phase. Both ends of the protein are moved using the prerotation move only (gaussian bias move).

- Gaussian bias move without chain closure. No restriction on the length of the moved segment.

- Gaussian bias move with the chain closure algorithm. No restriction on the length of the moved segment, giving much more flexibility than standard CRA algorithm (for example useful to sample a 5 residue long loop/chain).

The last two options are not available in the MCPRO[50] package which makes the ProtoMS[51] software more adaptable to the various biological problems. Theses new implementations have been tested using a long linear polyalanine protein and results are described below.

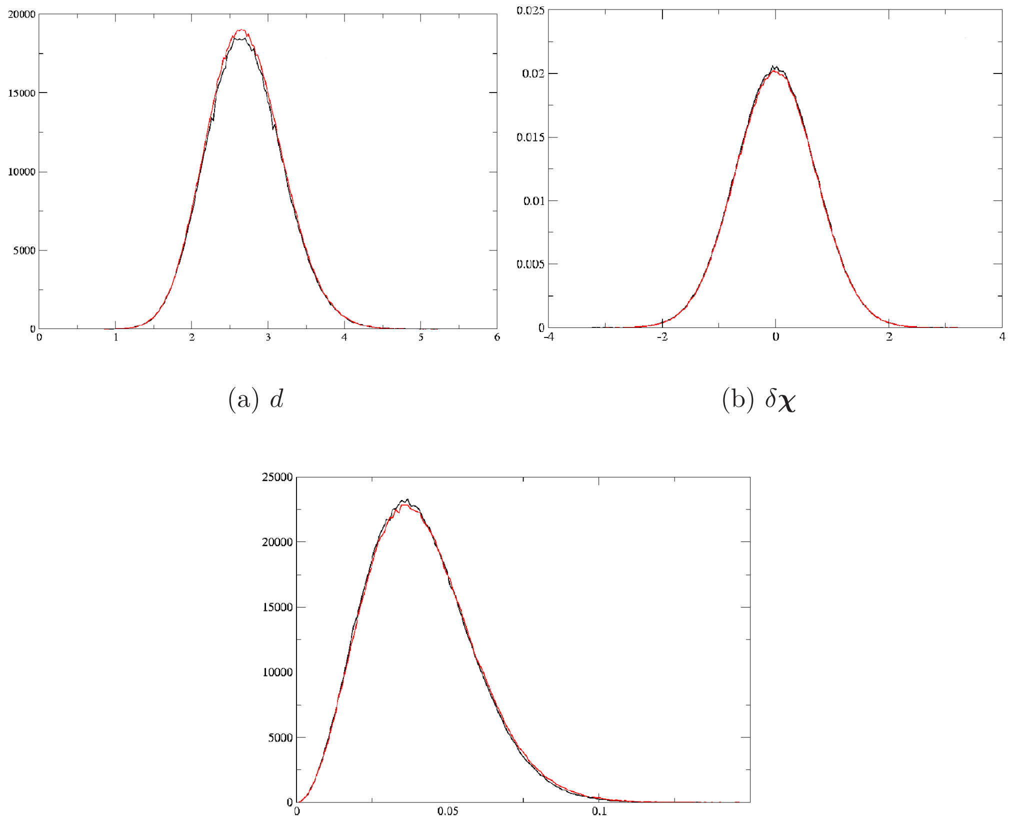
5.3.6 Extended concerted rotation moves: Results

The speed of the original CRA move has been compared to the ProtoMS[51] standard backbone move to yield the computing time per step in table 5.5. The first column shows the time ratio per move between standard backbone move and CRA move in ProtoMS. Standard backbone moves are about 7 times faster (CRA is slower, but it moves 4 residues in a concerted fashion). The second column shows the time ratio per residue between backbone moves and CRA moves in ProtoMS. The CRA move appears to be less than twice as slow per residue moved, but on the other hand has a better acceptance rate. The last column show the difference in speed when the derivatives of the atom **a** are computed in the original CRA algorithm implemented in the ProtoMS package with respect to the speed when the same derivatives are computed using the Flory frame of references. Flory's frame of references is slower as all the coordinates and the matrices has to be recomputed at each step but both methods lead to the same results.

	ratio time per step	ratio time per step per residue	ratio of the derivatives method
CRA Move/ProtoMS BB move	7.05	1.76	0.18

Table 5.5: Computing time comparison between the original CRA algorithm and backbone moves in ProtoMS.

To implement the prerotation move correctly, the original algorithm from MCPRO[50] implemented in ProtoMS[51] is compared to the modified version using the extended gaussian bias. Several parameters from reference[83], such as different gaussian random number distributions (and consequently $d^2 = (\delta \boldsymbol{a})^2$) have been generated along a million step MC trajectory of the "phantom" chain. The distribution of d, of the vector $\delta \boldsymbol{\chi}$ and the distance of the prerotation move $\delta \boldsymbol{a}$ (distance of the atom $\mathbf{a}$ before and after the move) have been plotted in figure 5.8.



(a) d

(b) $\delta \boldsymbol{\chi}$

(c) $\delta \boldsymbol{a}$

Figure 5.8: Distribution along a 1 million step trajectory, for the original CRA code and the gaussian bias implementation (distance $\delta \boldsymbol{a}$ in Å). The black curves are obtained using the original CRA algorithm implemented in ProtoMS[51]. The red curves are obtained using the cross product method.

Two different sets of simulations have been performed to test the variables plotted in figure 5.8: a first simulation using a set of numbers built from a gaussian distribution, and a second one a set using a fixed random seed. This is to obtain the same series of gaussian number to generate the bias during the prerotation phase. Both simulations lead to the same properties. Figure 5.8 compares the values of the original CRA algorithm and the modified gaussian bias in the case where both algorithms are using the same seed. The values drawn from the random distribution using the same seed are close one to each other, so the recursive part of the algorithm that differentiates the coordinates of the atom **a** can now be extended to more than 4 residues.

Simulations using a gaussian bias move (implemented in ProtoMS[51]) for 4 to 8 residues long have been run during 1 million steps in vacuum on the ala(14) polypeptide. All moves start from the 3^{rd} residue of the chain, with the number of moving residue extended from 4 to 10. This aims to test the extended gaussian bias in terms of acceptance rate. The distribution of the distance between the atom **a** before and after the prerotation move is plotted in figure 5.9. Figure 5.9 shows an increase in the distance the atom **a** is moved during prerotation. As the number of residues increases, the number of prerotation moves that are sufficiently small to lead to a chain closure decreases. Obviously, the longer the prerotation move, the more difficult the closure.

However, the use of gaussian bias on both bond and torsion angle but without chain closure leads to good results in terms of enhanced sampling. The c_1 and c_2 parameter for the acceptance rate and the force of the bias are those used in the reference[83]. The acceptance rate per chain length are shown in table 5.6. Table 5.6 shows clearly that the acceptance rate for the gaussian bias moves decreases with the length of the chain. It has to be noticed, that even with an 8 residue long chain, this implementation achieves a better sampling and a higher acceptance rate then the rigid unit backbone moves. So gaussian bias is a promising method to sample large scale motion. Being able to close any chain length after a gaussian bias move and hence perform a full CRA move should lead to even better results in terms of sampling.

The chain closure algorithm has been added to the extended prerotation

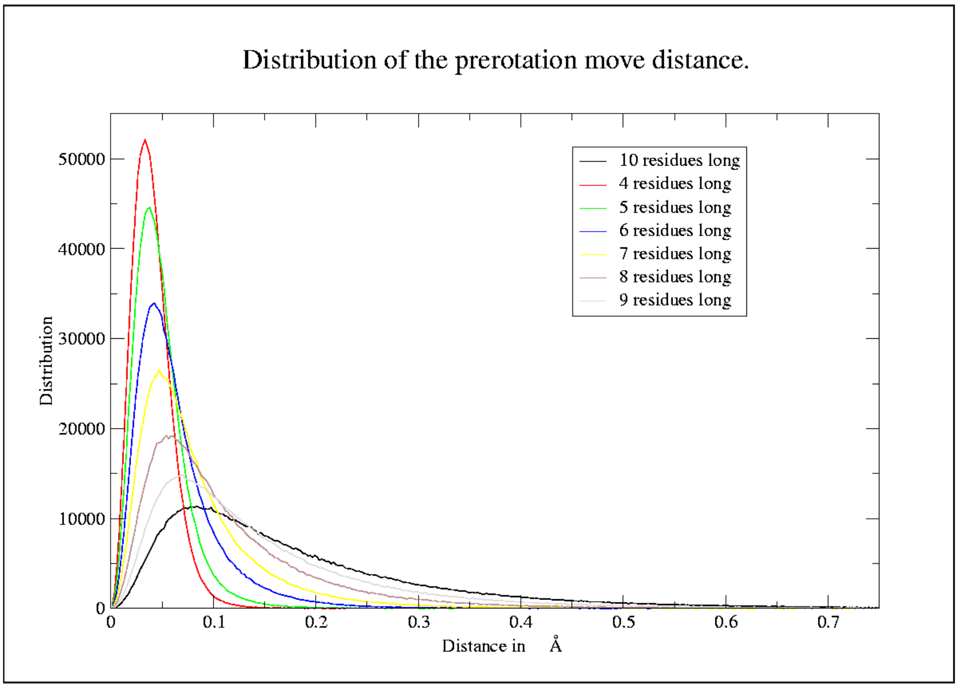


Figure 5.9: Distribution of the length of $\delta \boldsymbol{a}$ for various chain length. The distribution is obtained over a 100000 MC steps simulation.

number of residue	4	5	6	7
acceptance rate	32.85±3.38	21.99±2.39	13.98±1.50	9.92±1.10

number of residue	8	9	10
acceptance rate	7.65±0.83	6.42±0.70	5.75±0.62

Table 5.6: Acceptance rate for the gaussian bias move with no closure.

move. Several lengths of prerotation move have been tried with the chain closure algorithm on a 32 residue long poly-ala using the amber force field[39]. Simulation were run in GBSA solvent for 100000 steps (100 block of 1000 steps). Results of the different simulations are reported table 5.7. In red are reported the total acceptance rates (average of the total acceptance rate per block), in blue, the acceptance rates without the end move (acceptance rate over the whole simulation). So the use of the extended prerotation with closure still gives a good acceptance rate even for long chains.

number of residue	5	6	7	8	9
acceptance rate	20.86	21.87	22.87	26.77	27.80
acceptance rate	15.27	16.04	14.95	18.30	17.84

Table 5.7: Acceptance rate for extended prerotation with chain closure.

So different possibilities are now available for the ProtoMS[51] package, most of them not present in MCPRO[50]:

- Standard ProtoMS[51] move
- CRA move (as described in the reference[83]) on a random segment of a protein
- CRA move (as described in the reference[83]) on a random segment of a chosen loop of a protein
- Gaussian bias move of any length on a random segment of a protein.
- extended CRA of any length on a random segment of a protein
- extended CRA of any length on a random segment of a chosen loop of a protein
- any of the previous mixed with standard ProtoMS moves

The CRA method has been used on the CDK2 kinase (see figure 5.10). A 4 million step MC simulation has been carried out in implicit solvation without any ligand in the binding pocket. Regrettably, this simulation is unable to give energetic information, as some parameters for the loop are actually missing in the force field and have for this purpose been estimated. The CRA moves with standard length have been performed on the activation loop only. The active configuration is in blue. The inactive is in red. The cyan configuration is obtained after a 4 million steps MC simulation and the green ones show the pathway along the trajectory starting from the active structure. Owing to the difference between active and inactive forms of the protein and to the missing parameters in the force field, we do not expect to see a complete interconversion of the loop between the two forms. However, a motion of the loop clearly happens. The trajectory snapshot shows that the loop is starting a closure motion (going from the active to inactive conformation). This approach is really promising. Using the correct force field this move should lead to an accurate sampling of the phase space and give insight on the closure mechanism.

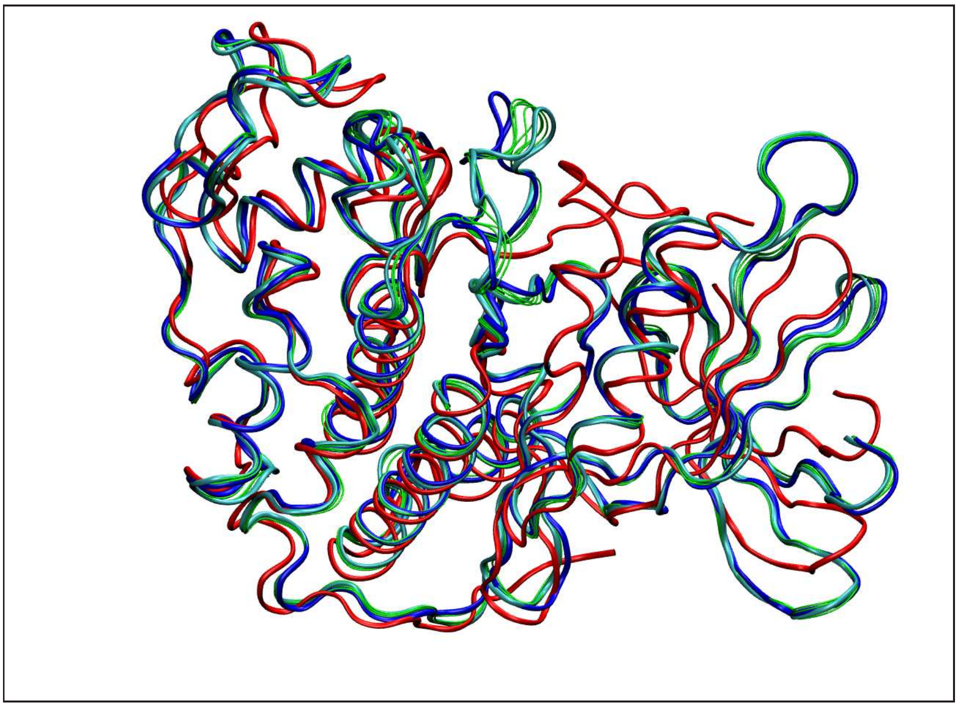


Figure 5.10: Snapshot of the CDK2 simulation using a modified CRA. In blue 2c5p pdb database reference and in red the 1pxm pdb database reference.

5.4 Miscellaneous

The ProtoMS[51] software although very fast and efficient is not perfect. Drawbacks are

- it does not incorporate any analytical tools
- its core structure is written in F77 limiting the use of memory
- a limited number of files can be opened at the same time

The last two points make running long simulation from one input file impossible (each time a pdb file is written or a restart is closed/open, it stays in the stack pile, and the maximum number of files F77 can handle is limited to 30). To overcome the problem, MC simulations using ProtoMS[51] are run in block. Each block can be described from the following process:

- Load force fields, simulation parameters and proteins, solvents, solutes
- Load conformational information
- Run N steps of simulation
- Write new conformational information

So each block can be repeated, reading conformational information from the previous. To perform this task, the use of Perl scripts has come in very handy. A Perl script has been written to write the input file and then run the simulation. The script can display block numbers in two different ways, using standard increments from 1 to N, or using a Cshell incrementation from 001 to NNN.

The ProtoMS[51] package allows the user to write PBD structure of the system every N steps. So conformational analysis can be made using the PBD output.

Once again, the use of the Perl language has proved to be an efficient tool.

Several script have been coded to perform several type of analysis:

- script to compute the radius of gyration of a protein for the whole simulation

- script to compute the Ramachandran plot of a protein for the whole simulation (minus N and C terminal residues)
- script to build a water shell around a given number of residues or solutes
- scripts to compute small task such as reordering outputs, normalising angled between 0 and 2π, compute distribution of angles from a set of data, repeating commands, computing standard and floating average etc etc.

Parallel tempering simulations aim to improve sampling of the phase space, by exchanging replicas at different temperatures (see section 3.6). The ProtoMS[51] package does not feature such an implementation. However, as the temperature of the simulation can be user-defined for a given simulation, a parallel tempering script has been written.

The script runs several simulations at different temperatures and every X steps, performs an exchange test based on the energy of the system. If the test is successful, the restart coordinates are exchanged and the next block of the simulations is run. At the end of the simulation the output file is used to calculate the acceptance rate and another Perl script to draw the exchange plot.

5.5 Concluding remarks

A variety of ways to use standard or enhanced CRA is now available in the ProtoMS[51] package to allow greater sampling of protein backbone. We have been interested in using the CRA algorithm to compute free energies and study loop flexibility rather than study folding of proteins. The next chapters will describe the use of the CRA to solve biological problems in several systems.

Chapter 6

Lysozyme

Lysozyme was discovered by Alexander Fleming in 1922, during his search for medical antibiotics[111]. Like most great discoveries, luck played its part. During a cold, Fleming had a drop of mucus fall into a bacterial culture and discovered that the bacteria were killed. This phenomenon led to the discovery of lysozyme, which had killed the bacteria. Sadly, owing to its size, Lysozyme could not be used as a drug (but later Fleming discovered the first anti-biotic penicillin, once again a share of talent and luck).

Lysozyme serves as a non-specific innate opsonin* by binding to the bacterial surface, reducing the negative charge and facilitating phagocytosis of the bacterium before opsonins from the acquired immune system arrive at the scene.

The mechanism[113] responsible for reducing the negative charge, involves hydrolysis of the β (1-4) glycosidic bond between N-acetylglucosamine sugar (NAG) and N-acetylmuramic acid sugar (NAM) (see figure 6.1). This reaction takes place in a long deep cleft, which contains the active site of Lysozyme (residues Glu35 and Asp52 for chicken egg white Lysozyme).

The first crystal structure of Lysozyme was obtained in 1974 by Diamond with a resolution of 2 Å[114] and can be found in the PBD database under the

*Opsonins are macromolecules binding to the surface of a cell and aiming to enhance the phagocytosis

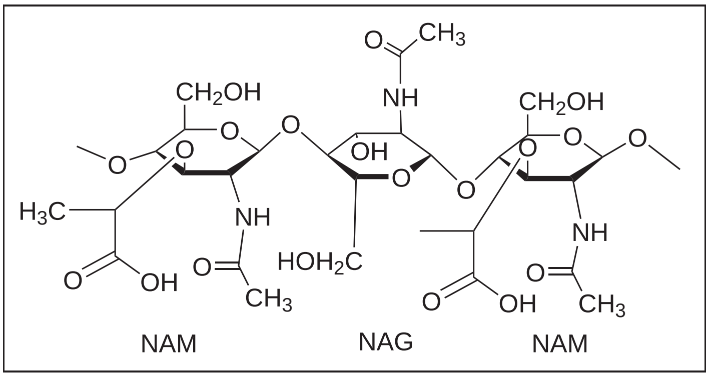


Figure 6.1: Representation of the target site of the lysozyme[112]

references 1lyz to 6lyz. There are more than 1400 hits on the PBD database for the keyword Lysozyme. The L99A mutant is known to make enough space in the cavity to accommodate a benzene ring in the binding pocket[115,116]. This mutant has been use for protein engineering and such binding inhibits the function of the protein[115,116].

In this chapter we will first review previous work on the T4 lysozyme L99A mutant and give some insight into the crystal structures (rcsb database references 181L to 188L[117,118]). Then we will detail and discuss the work achieve by using the CRA algorithm to sample the lysozyme phase space.

6.1 Previous work and structure

6.1.1 Protein structure

The L99A mutant of the T4 lysozyme, has been crystallised bound to several ligands[117,118]. Entries for each ligand are:

- benzene 181l
- benzofuran 182l
- indene 183l
- isobutylbenzene 184l

- indole 185l
- n-butylbenzene 186l
- para-xylene 187l
- ortho-xylene 188l

Figure 6.2 shows a superposition of some crystal structures described in the references[117,118].

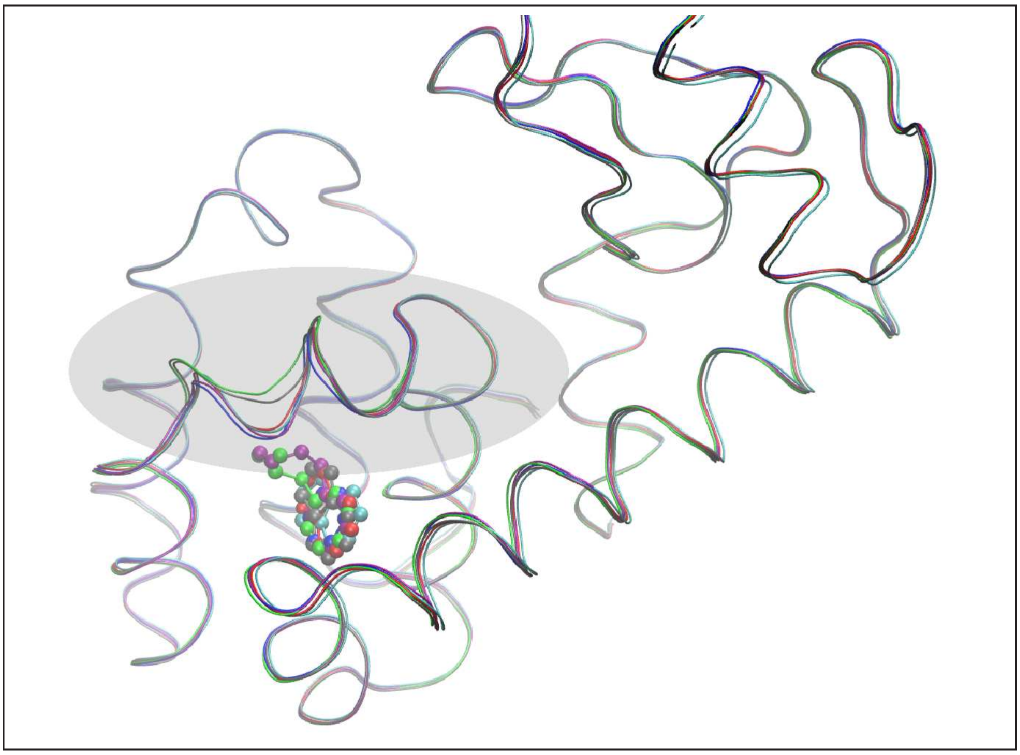


Figure 6.2: superposition of the crystal structure of the lysozyme bound to different ligands[117,118]. F-loop is represented in the shaded region. Colour code: 181L in blue, 182L in red, 183L in cyan, 184l in green, 185l in grey and 186L in magenta.

Each crystal structure is bound to a different ligand and the F-loop of the protein adopts a different conformation. The RMSD between the different structures can be found in table 6.1.

Table 6.1 shows that most of the deviation occurs in the F-loop region. Although the RMSD between the various structures is small, the F-loop

PDB code	182l	183l	184l	185l	186l	187l	188l
protein	0.314	0.377	0.389	0.422	0.296	0.233	0.363
F-loop	0.586	1.034	1.233	0.534	0.701	0.593	1.095

Table 6.1: RMSD between the lysozyme bound to benzene and the lysozyme bound to other ligands (values in Å). Second row shows the RMSD with respect to the complete crystal structure, third row, the RMSD of the F-loop only, both RMSDs computed with all the heavy atoms of the loop. RMSD have been calculated after superposition of the backbone of the crystal structures.

adopts quite different conformations for each ligand (see figure 6.2 for a more graphical view). The next section will describe a brief overview of the existing work executed on the lysozyme protein.

6.1.2 Existing work

Both experimental and theoretical studies have been performed on the T4 lysozyme L99A mutant[25,117–126] to obtain binding free energies for the set of ligands. *In silico* results were obtained using MD and several techniques to enhance the sampling. Some methods used restraints on the ligands[25,125] or another method called confine and release[119,120] to overcome some internal energetic barriers. The experimental binding free energies were obtained using the protocol described in the references[117,118].

Details of the simulations can be found in the respective publications[25,119,120]. To summarise the methods, Roux *et al* uses restrains on the ligands. The ligand in the bulk is restraints to the position it adopts in the bound state and is then translated into the binding site where it is released completely. The method developed by Soichet *et al* deals with the high energy barrier of the rotational changes of the side chain of the valine 111 by using a confine and release method. The Binding free energy is computed by first driving the protein to its bound conformation. The ligand is inserted in the binding pocket while the protein is kept confined. To close the cycle, the bound system is released from any constraints. Such method is used to overcome the

	Experimental value	MD value[119]	MD value[25]
benzene	-5.19±0.16	-4.56±0.20	-5.96±0.19
benzofuran	-5.46±0.03	-3.53±0.06	-5.62±0.20
indene	-5.13±0.01	-1.75±0.07	-2.47±0.24
isobutylbenzene	-6.51±0.06	-5.01±0.20	-9.67±0.38
indole	-4.98±0.06	-0.42±0.08	-4.24±0.17
n-butylbenzene	-6.70±0.02	-4.87±0.14	-8.75±0.36
p-xylene	-4.60±0.06	-1.27±0.18	-9.06±0.21
o-xylene	-4.67±0.06	-3.54±0.17	-7.59±0.19

Table 6.2: $\Delta G^{\circ}_{binding}$ in kcal/mol for various ligands from previous studies.

kinetic trapping of the metastable state created by the side chain. Results in table 6.2 shows that theoretical studies do not reproduce systematically the experimental binding free energies. There are several issues to be dealt with. The first issue is the conformational change in the F-loop of the lysozyme. The binding pocket of the lysozyme is big enough to accommodate a benzene ring plus a small "blob". However the binding pocket is very tight, and the F-loop has to accommodate for changes in the conformation of the ligand. Owing the nature of the shape of the ligand, MD might not be able to sample the system for "long enough". Work from Roux[25] seems to suggest that the length of a typical MD run is not enough to sample such changes which indicates that the amplitude of the sampling could not be achieved using time related methods. The second issue is the presence of a rotamer on the valine 111 (see figure 6.3). Two different rotamers of the valine 111 exist in different crystal structures to accommodate different ligands. These rotamers create repulsive/attractive interactions with the ligand, making the sampling of the binding energy more difficult.

The conformational change from one rotamer to the other cannot be sampled using the standard MD method due to the high energetic barrier. To over come this barrier, specific methods have to be used[119,120]. But even

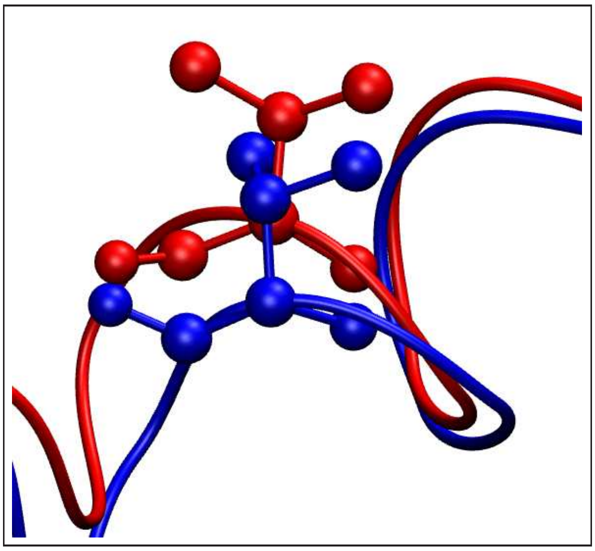


Figure 6.3: Two rotamers of the valine 111 in the PDB references 184l (blue) and 185l (red).

using such methods, the calculated relative binding free energies are different from the experimental ones.

Prior studies using MC methods have been performed within our group to try to reproduce the experimental relative binding free energy between the indole and the isobutylbenzene ligands. To compute relative binding affinities between two ligands, two routes are possibles (see figure 6.4). The binding free energies for both ligands are computed and then the difference between the energies is made (route $\Delta G_4 - \Delta G_2$ in figure 6.4), or the alchemical transformation[56] route is used. One ligand is mutated into another in both the protein and solvent environment, and the difference of the energies is made (route $\Delta G_1 - \Delta G_3$ in figure 6.4).

We have been using the alchemical transformation route[56]. For both the 184l and 185l crystal structures, the ligand has been perturbed from indole to isobutylbenzene and the relative binding free energy computed. Several simulation were performed using different solvent models and a scoop of the protein:

- using an explicit water cap and no backbone moves on the scoop.

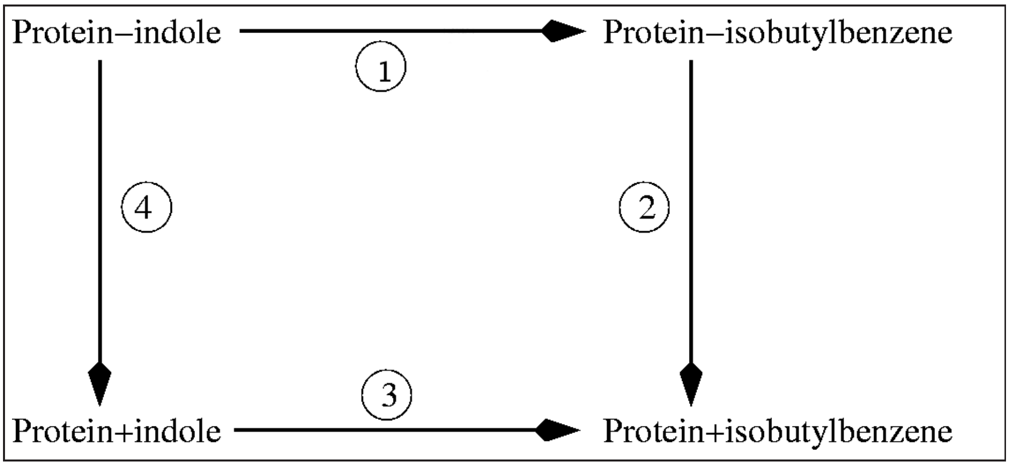


Figure 6.4: Thermodynamical cycle used for the MC simulations.

- using an explicit water cap and backbone moves on the F-loop of the scoop.
- using an explicit water cap and backbone moves on the whole scoop.
- using GBSA and no backbone moves on the scoop.
- using GBSA and backbone moves on the F-loop of the scoop.
- using GBSA and backbone moves on the whole scoop.

Backbone moves are rigid unit backbone moves as defined in section 3.3.1. Each of these simulations have been performed at 25 °C using NVT dual topology[57,58] and the Amber and GAFF forcefield[39,40]. The simulations using the water cap were run in blocks of 10K MC steps. First 100 blocks of equilibration were run, and then 500 blocks for data collection. RETI[57,58] moves were performed every 2 blocks (20K steps). GBSA simulations follow the same protocol only with a 20 Å cut off, a threshold of 0.005 Å for the update of the GBSA shell, and blocks of 3×1000 MC steps. Results are displayed table 6.3. Values of the relative binding free energy are in kcal/mol. The first column tells the nature of the solvent, second column the nature of the backbone moves, none (off), everywhere (on) or only on the F-loop (Helix-F).

	Backbone moves	Starting from 184l	Starting from 185l
Water cap	None	-7.45±0.17	6.16±0.22
	On	-3.44±0.26	2.08±0.38
	Helix-F	-7.16±0.20	3.67±0.22
GBSA	None	-4.80±0.35	8.34±0.32
	On	-2.41±0.29	0.97±0.44
	Helix-F	-4.73±0.24	5.81±0.40

Table 6.3: relative binding free energy between the indole and the isobutylbenzene in the 185L crystal structures(courtesy of Dr Michel)

Table 6.3 shows that standard MC simulations do not reproduce experimental results. There are several difficulties associated with the mutation from indole to isobutylbenzene:

- the two ligands have a completely different shape

- experimental relative binding affinity is less than 2 kcal/mol. Computing relative binding free energies for such small difference within 1 kcal/mol is acceptable, however ideally we would like to look at a set of data in which the difference in affinity is more significant.

- the valine 111 presents different rotamers in the two crystal structures.

However the use of backbone moves is a clue that the conformation of the F-loop is critical in the binding process. The hypothesis was made that the use of large scale moves such as the CRA will benefit the sampling and the calculation of the relative binding affinity.

The next section will discuss the effect of using MC simulations and the CRA algorithm in the sampling of the F-Loop and the influence in the calculation of the relative binding affinity.

6.2 Use of CRA in the lysozyme study

The conformational changes happening on the F-loop of the lysozyme protein have proved to be an interesting challenge for the standard sampling methods. In this section we will discuss the effect of the CRA algorithm on the sampling of the loop and the computation of the relative binding free energies.

6.2.1 Conformational change

Owing our prior knowledge of lysozyme, several simulations have been run on the crystal structures bound to the isobutylbenzene (PDB file 184l) and the indole (PDB file 185l). For all the simulations, unless stated otherwise, solvent was modelled using an implicit model (GBSA see section 3.6), cut off for electrostatic interactions was set to 10 Å, the cut off for the GBSA to 20 Å, the threshold for the re-computation of the GBSA was set to 0.005 Å, a scoop of 15 Å around the biggest ligand was used (see figure 6.5), CRA moves were used with a prerotation length of 4 segments (as described by Ulmschneider *et* al[83]). The coordinates of the following residues were constrained: 3, 5-7, 10-11, 22, 70-73, 76, 80, 92-94, 123-128, 135, 137, 139-143, 145, 147-148, 151-152, 154-156, 158-159, 161. These residues are located outside a 10 Å radius of the ligand.

The scoop of the protein had an initial charge of +5. The charge was reduced to zero by neutralising three lysine residues lying in the outer part of the scoop K124,K135,K147. Afterwards, two extra residues were added to the scoop, Asp159 and Glu5.

First the influence of various parameters have been tested. Parameters such as having rigid unit backbone moves outside the F-loop, the length of loop on which the CRA moves were applied as well as the influence of keeping some of the residues fixed. Owing to its concerted nature (both ends to be kept fixed), the CRA move has been applied outside the F-loop (residues 106-115) from residues 105 to 118.

To enhance the sampling, ligands have been swapped over. By having the ligands crossed from one crystal structure to another, we were expecting

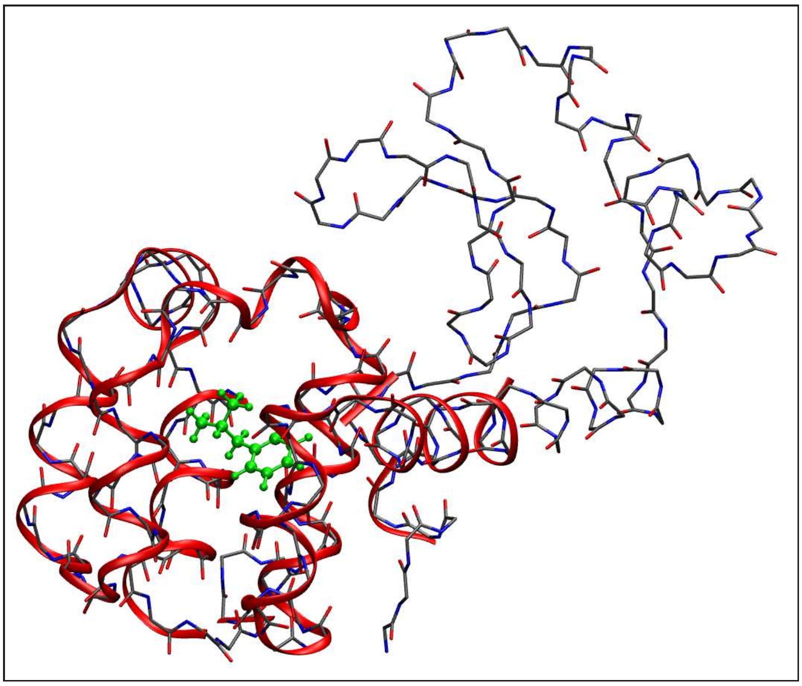


Figure 6.5: Scoop of the 184l protein used in the simulations over the original crystal structure (red ribbon). The backbone atoms of the complete protein and the isobutylbenzene (green) are also represented.

to see the conformation of the F-loop change towards the corresponding configuration of the F-loop. A first set of five simulations has been run, changing several parameters. All simulations have been run for 2 million steps.

- First simulation where no rigid unit backbone moves were allowed on any residues, residues mentioned above were kept fixed (no side chain moves, no backbone moves) and CRA moves performed between the residues 105 and 118;
- second simulation where rigid unit backbone moves were allowed on residues inside a 10 Å radius, residues mentioned above were kept fixed and CRA moves performed between the residues 105 and 118;
- third simulation where rigid unit backbone moves were allowed on residues inside a 10 Å radius, no residues were kept fixed and CRA

moves performed between the residues 105 and 118;

- fourth simulation where rigid unit backbone moves were allowed on residues inside a 10 Å radius, no residues were kept fixed and CRA moves performed between the residues 101 and 122;
- fifth simulation where no rigid unit backbone moves were allowed on residues inside a 10 Å radius, residues mentioned above were kept fixed and CRA moves performed between the residues 101 and 122.

The RMSD of the residues 105 to 118 along the simulation with respect to the 184l crystal structure during the simulation are plotted figure 6.6. The effect on the sampling to the different parameters is discussed according to the observation on the RMSD from figure 6.6.

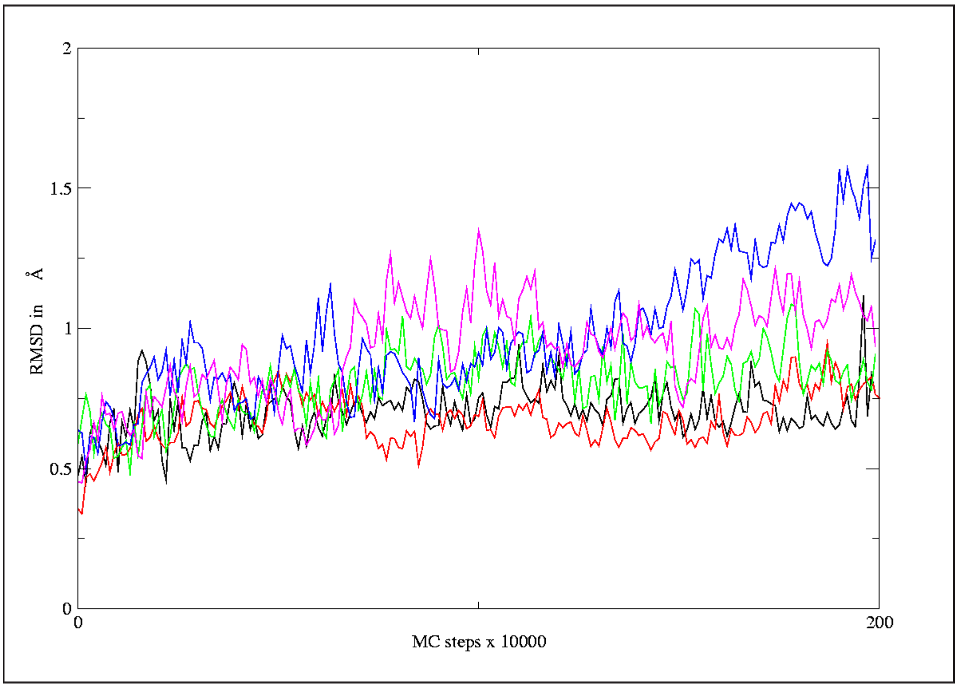


Figure 6.6: RMSD of the trajectories of the five simulations with respect to the 184l crystal structure. Simulations 1 to 5 are respectively black, red, green, blue and violet.

The length of the loop on which CRA was applied seems to have an important effect on the sampling (violet curve against black curve in figure 6.6).

Allowing rigid unit backbone moves on the other parts of the protein (green curve) enhances the sampling of the F-loop too by allowing the other parts of the protein to relax in order to accommodate the change of geometry of the F-loop (the green curve achieves greater sampling than the black and red ones). Allowing all residues to move increases the sampling (blue curve in figure 6.6), but due to the nature of the system (scoop of the protein) and the type of moves (see section 3.3.1) such a protocol is not recommended. Moving the outer ring of the scoop is not recommended as the residues composing it would just drift away, leading to an incorrect structure of the protein.

So the optimised sampling is achieved when rigid unit backbone moves are allowed outside the F-loop and CRA moves performed on a slightly longer segment of the protein (violet curve in figure 6.6). This protocol aimed to achieve the best sampling of the F-loop will be later used in the free energy perturbation (see section 6.2.3).

Others sets of simulations were run, with the appropriate ligand and no ligand respectively, for both crystal structures. A second set of simulations has been run on both crystal structures without ligand. This set is made up of 4 simulations:

- first simulation where rigid unit backbone moves were allowed, but CRA moves performed between the residues 101 and 123 at 25°C;
- second simulation where rigid unit backbone moves were allowed and CRA moves performed between the residues 101 and 123 at 100°C;
- third simulation where rigid unit backbone moves were allowed and CRA moves performed between the residues 101 and 123 at 150°C;
- fourth simulation where rigid unit backbone moves were allowed and CRA moves performed between the residues 101 and 123 at 200°C.

The third set of simulations has been performed on both crystal structures including their respective ligand. This set is made of 4 simulations:

- first simulation where no rigid unit backbone moves were allowed, but CRA moves performed between the residues 101 and 123 at 25°C;

- second simulation where rigid unit backbone moves were allowed and CRA moves performed between the residues 101 and 123 at 100°C;
- third simulation where rigid unit backbone moves were allowed and CRA moves performed between the residues 101 and 123 at 150°C;
- fourth simulation where rigid unit backbone moves were allowed and CRA moves performed between the residues 101 and 123 at 200°C.

The last two sets of simulations have confirmed the results of the first set as to which parameters to use in terms of sampling. In the third set, no major changes in the conformation of the F-loop were observed from respect to the crystal structures (as expected) but rather a nice sampling around the starting structure.

Another set of longer simulations was run for the crossed ligands using the CRA between the residues 105 to 118 and allowing rigid unit backbone moves on the whole scoop of the protein. The RMSDs of both trajectories with respect of both the 184l and 185l crystal structure have been plotted figure 6.7 and 6.8 respectively.

When the indole is used in the 184l crystal structure (crossed ligands), the RMSD shows that the F-loop does not converge toward the conformation of the 185l crystal structure. The black and red curve should cross each other. The average value of the black curve should go to near zero and the average value of the red one should converge around 1.5 Å. Observing such behaviour would mean that the F-loop has adopted the conformation relevant to the ligand in the binding pocket. However, converging structures are obtained until 5 million MC steps where the F-loop starts to evolve freely (no more convergence of the RMSD toward a definite structure). This could simply be explained by the fact that the indole occupies a smaller volume than the isobutylbenzene and the fact that the binding pocket of the 184l crystal structure is bigger than the binding pocket of the 185l crystal structure. Snapshots of the simulations confirm this hypothesis, and shows a good sampling of the indole within the binding pocket. The value of the dihedral angle defined in figure 6.10(a) is plotted in green figure 6.7 and aims to quantify the sampling of the ligand within the binding pocket (see also figure 6.9

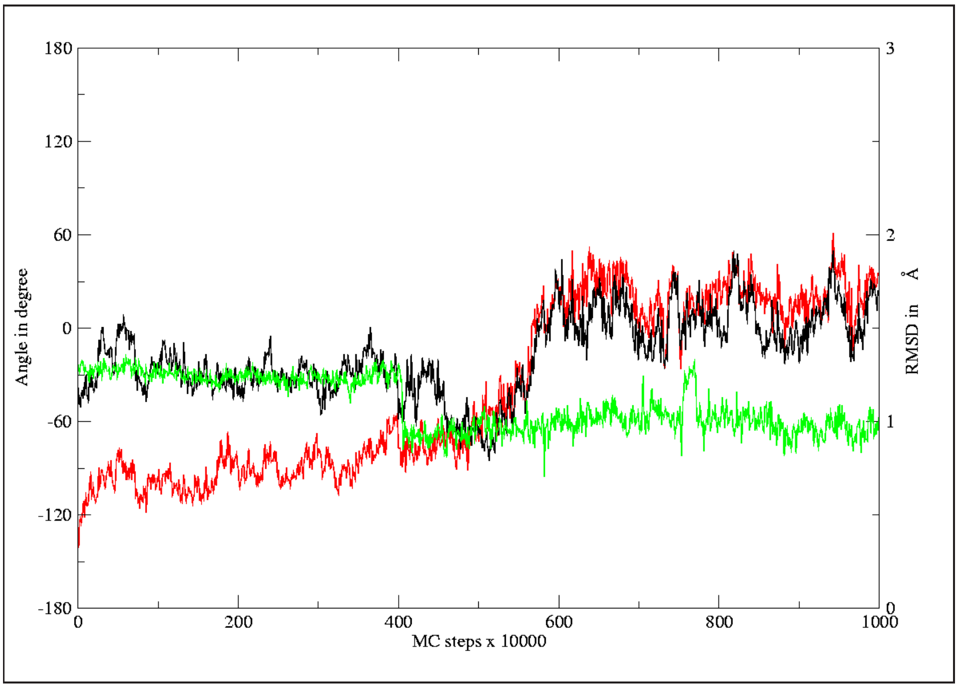


Figure 6.7: RMSD of the simulation starting with the 184l crystal structure bound to indole. RMSD is computed with respect to the 184l (black) and 185l (red) crystal structure. The RMSD is calculated on the backbone atoms only. Green curve is the value of the angle defined in figure 6.10(a).

for a superposition of two structures of the indole during the simulation).

The value of the angle defined in figure 6.10(a) shows that the indole samples much of the binding pocket during the simulation. The sudden change in the value at around 4 million and 8 million MC steps is due to the ligand drifting away from its original conformation in the binding pocket.

When the binding pocket of 185l crystal structure is filled with the isobutylbenzene, the results are however not up to expectations. Having the isobutylbenzene in the indole binding pocket, repulsive interactions were expected to lead to the F-loop quickly adopting a conformation close to the 184l crystal structure. The RMSD of the F-loop (figure 6.8) shows this is not the case.

However, expecting to capture the subtle change between the two conformations of the F-loop by using only RMSD is a bit optimistic. As the CRA algorithm drives changes in bond and dihedral angles, using a Ramachandran

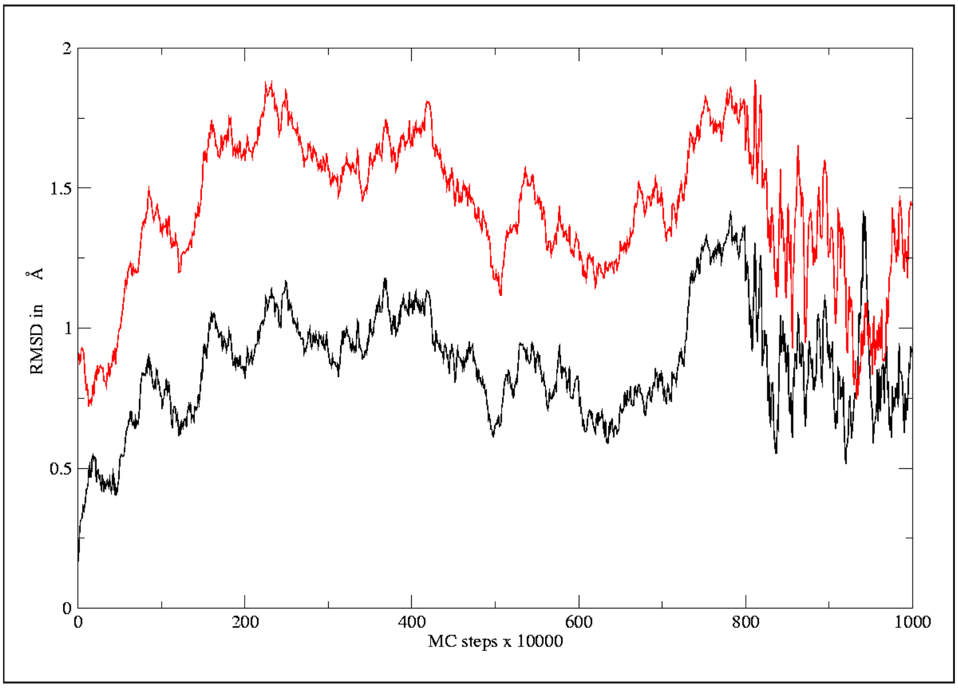


Figure 6.8: RMSD of the simulation starting with the 185l crystal structure bound to isobutylbenzene with respect to the 185l (black) and 184l (red) crystal structure. The RMSD is calculated on the backbone atoms only.

plot[26] should provide good insights of the conformational changes. For three crystal structures (apoprotein (1l92), 184l, 185l), the Ramachandran plot[26] has been plotted in figure 6.11.

The colour code of the figure 6.11 is: black residue 102, red residue 103, green residue 104, blue residue 105, dark green residue 106, brown residue 107, maroon residue 108, violet residue 109, cyan residue 110, magenta residue 111, orange residue 112 and indigo residue 113. The Ramachandran plot[26], shows that except for 3 residues, the conformation of the dihedral angles of the loop are very similar for the three structures. Residues that have relatively different Φ, Ψ conformations are residues 108, 110, and 111, but the value of the angles Φ and Ψ are still within a space of twenty degrees. The average value of the Φ and Ψ angles with the standard deviation during both crossed simulations (184l crystal structure with indole and 185l with isobutylbenzene) are plotted in figure 6.12.

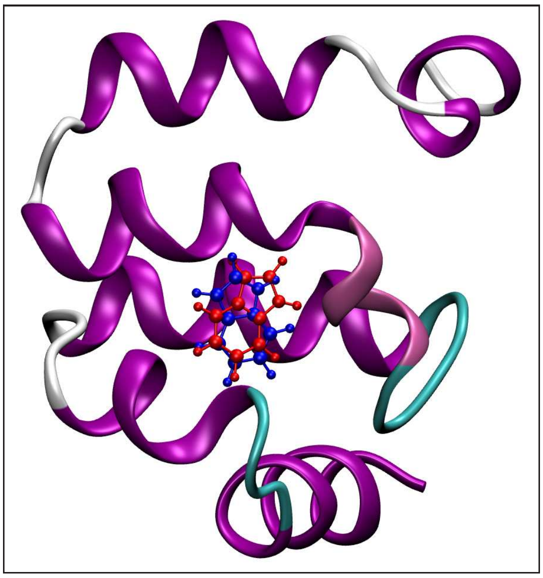
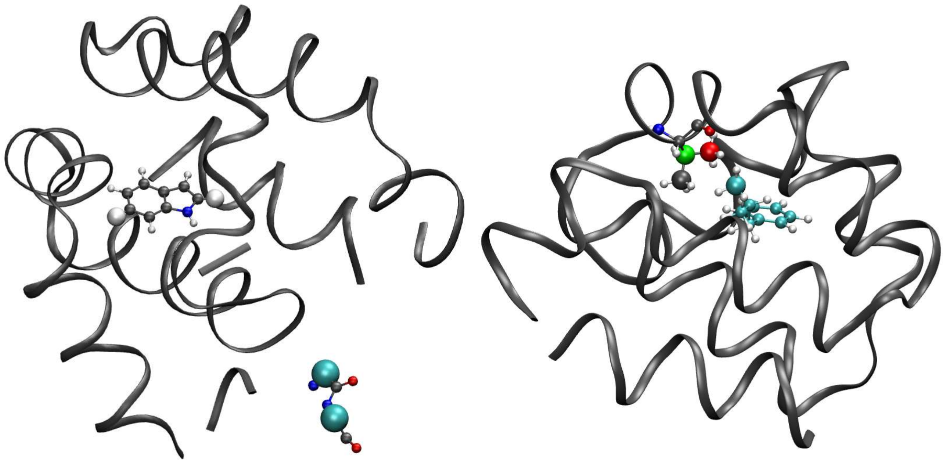


Figure 6.9: Superposition of to structures of the indole in the crystallographic binding pocket after 2 and 7 million of MC steps (respectively blue and red).

An interesting feature to note from the figure 6.12, is that during both crossed simulations, the F-loop samples regions of the conformational space that are very close to each other. For both simulations, the sampling of the backbone is very similar. The spread of the Φ and Ψ angles for the 184l crystal structure with indole overlap the spread of the Φ and Ψ angles for the 185l crystal structure with the isobutylbenzene. The plot for the simulation using the 184l crystal structure shows that the residue 106 achieves a greater sampling than its counterpart in the 185l crystal structure. However, both residues are sampling the same region of the phase space. Thus the distribution of the Φ and Ψ angles cannot give any insight into the phenomenon that occurs when the isobutylbenzene is inserted in the binding pocket of the 184l crystal structure (limited sampling). The explanation of such phenomenon is hence, unlikely to be backbone related.

In figure 6.3 the two valines are shown to have different rotamers. This is of critical importance in describing the behaviour of the F-loop of the 185l crystal structure when bound to isobutylbenzene.

The next section will discuss the issues related to the existence of the two



(a) Atoms used to define the dihedral angle between the indole (white spheres) and the Cα of residues 10 and 11 (blue spheres).

(b) Atoms used to define the distances between the valine 111 and the isobutylbenzene (spheres). On the valine the atom CG1, CG2, CB are respectively red, grey, green. One the isobutylbenzene the atom C10 is cyan.

Figure 6.10: Atom used in the simulations to compute angles and distances. Simulation of the 184l crystal structure with indole (sub-figure 6.10(a)) and simulation of the 185l crystal structure with the isobutylbenzene(sub-figure 6.10(b)).

rotamers and the implications of such a change in the side chain conformation on the sampling of the F-loop.

6.2.2 Rotamer dependency

Simulations using the isobutylbenzene in the 184l crystal structure binding pocket have not led to the expected results. Both the RMSD and the Ramachandran plot[26] of the F-loop have failed to prove significant sampling of the F-loop towards the 184l crystal structure. Careful observation of the snapshot of the simulation, shows that the sampling of the F-loop is linked to the sampling of the isobutylbenzene. This suggests the presence of new interactions between the isobutylbenzene and the F-loop. The behaviour of the

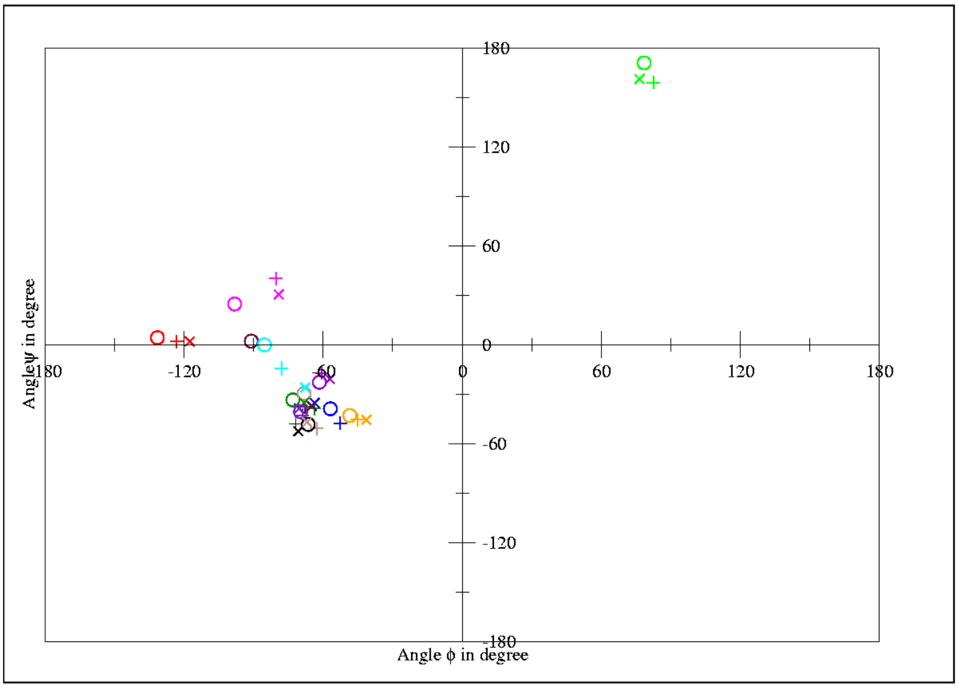


Figure 6.11: Ramachandran plot of the apoprotein (circles), 184l (+) and 185l (x) crystal structures.

isobutylbenzene and the valine 111 have been investigated during the simulation and interesting features have been discovered. It appears that after a very short period of time (500 000 MC steps), the distance between the isobutylbenzene and the valine 111 remains constant. The figure 6.10(b) shows the atoms used to compute some specific distances between the isobutylbenzene and the valine 111.

The distance between the atom C10 of the isobutylbenzene with the atoms CB, CG1, CG2 of the valine 111 are plotted figure 6.13 respectively in green, red and black.

The figure 6.13 clearly shows that the position of the isobutylbenzene and the valine 111 are linked. The three distances quickly become trapped into a local energy minima. The distances plotted figure 6.13 suggest the existence of a hydrophobic cluster between the side chain of the valine 111 and the methyl group of the tail of the isobutylbenzene. The two methyl group are facing one to the other and the hydrogens have a staggered position when

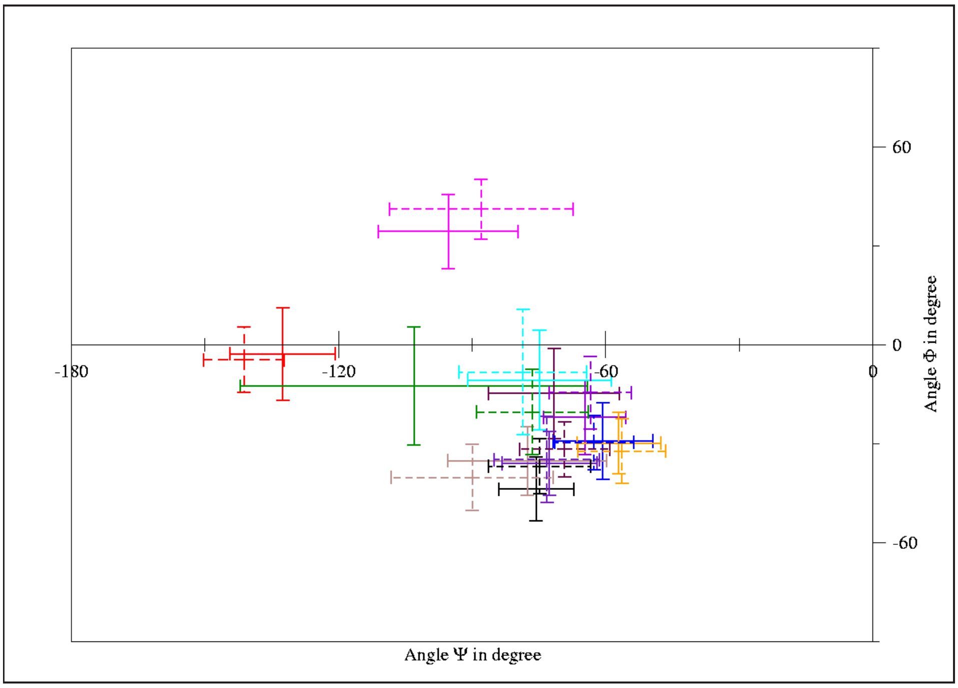


Figure 6.12: Ramachandran plot for the 184l with indole (plain) and 185l with isobutylbenzene (dashed). The average of the angles during the simulation is represented centred on the standard deviation.

looked through the atom C10 of the isobutylbenzene and the atom CG1 of the valine 111. So whereas some repulsive interactions were expected, the different position of the side chain of the valine 111 creates favourable interactions. Whereas such interactions are weak, they are nevertheless strong enough to restrain the position of the F-loop close to the isobutylbenzene. The interaction between the two methyl group is strengthened by the tight fit of the binding pocket. The aromatic part of the isobutylbenzene is tightly bound to the binding pocket and hence little space is accessible for the ligand to move.

To confirm this hypothesis, the same simulation has been run at 523 K (25°C). The values of the angle χ of the valine 111 for both simulations (298 and 523 K) are plotted figure 6.13. The data at 523 K clearly show a change in the value of the dihedral angle χ between the value 180° and 300° (-60°). These two values correspond to the two different rotamers of the valine. This

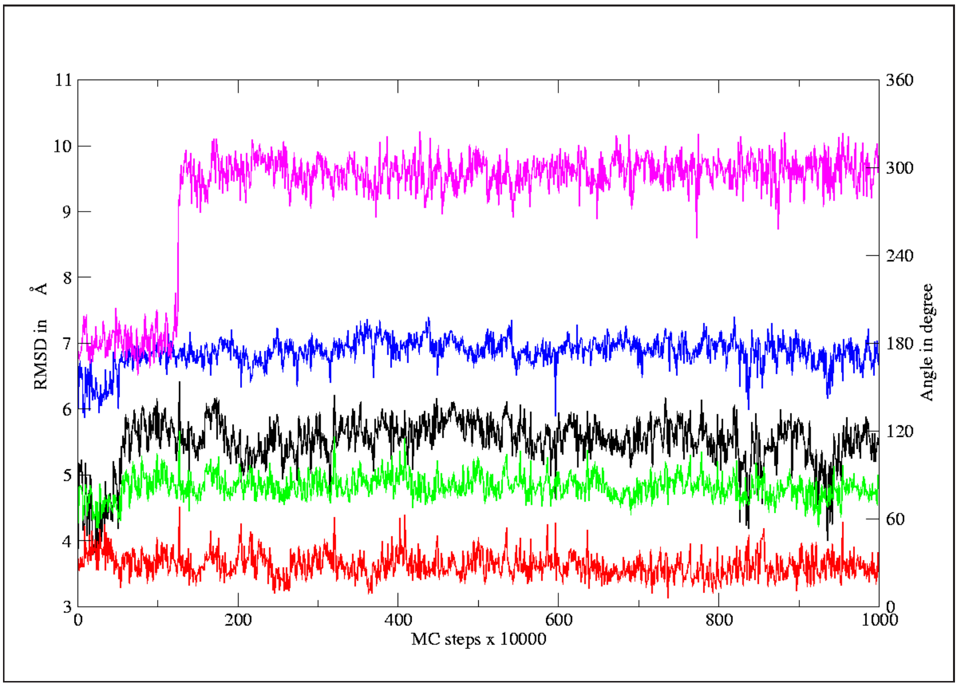


Figure 6.13: Distances between the atom C10 of the isobutylbenzene and the atom CB (green curve), CG1 (red curve), CG2(black curve), and dihedral χ of the valine 111 at 298 K (blue curve) and 523 K (violet curve).

phenomenon confirms our hypothesis of a hydrophobic cluster. At 523 K, the energetic barrier of the rotamer position is easily overcome, allowing the valine to adopt the appropriate rotamer. The change in conformation (in purple in figure 6.13) appears after only 1.5 million MC steps.

To try to enhance the sampling, PT[59,60] techniques have been used. For both crystal structures, ligands have been swapped and a set of 14 parallel simulation starting from the same configuration at different temperatures have been run. Temperatures were spread between 298 K and 473 K as follow: 298, 303, 310, 315, 323, 333, 345, 358, 373, 393, 408, 423, 443, 473. Every 10000 MC steps, the exchange test is performed according to the equation 3.41. The path of the simulations starting with the 185l crystal structure and the isobutylbenzene at 298 K (25°C), 323 K (50°C), 373 K (100°C), 423 K (150°C) and 473 K (200°C) are represented in figure 6.14.

Figure 6.14 shows that all of the five simulation are exchanged along the

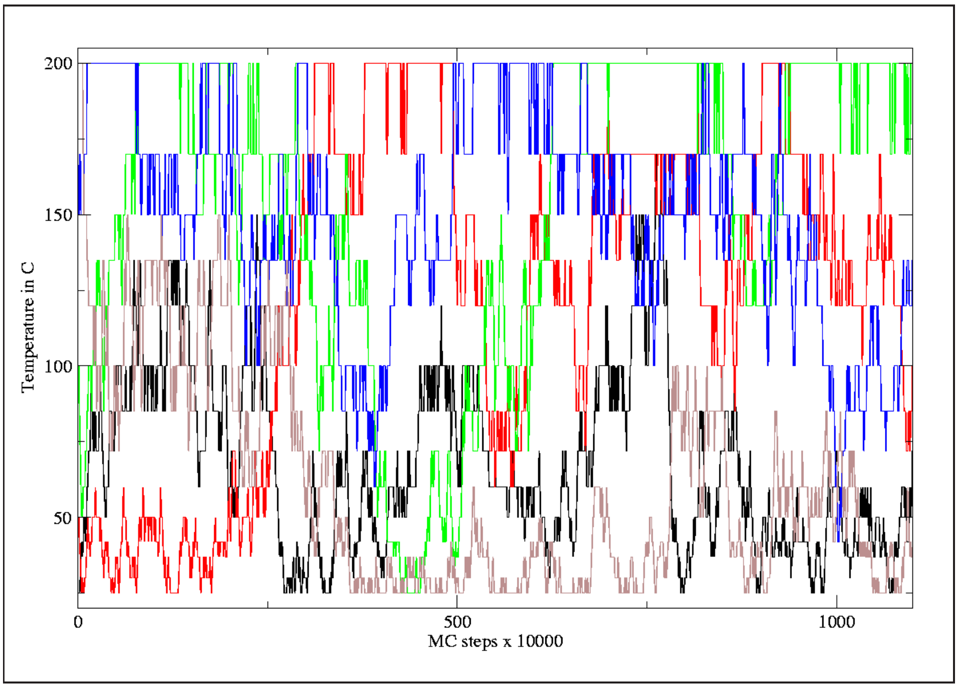


Figure 6.14: Path for the simulations starting with the 185l crystal structure and the isobutylbenzene at 25°C (Black), 50°C (red), 100°C (green), 150°C (blue) 200°C (brown)

temperature gradient allowing greater sampling of the phase space. Simulations at the extreme range of temperatures manage to travel across the whole range of temperatures. Such sampling enables the system at 298 K to exchange configuration with higher temperature as expected. To see if the use of the PT[59,60] has an effect on the sampling of the dihedral χ of the valine 111, the value of the dihedral at 298 K has been plotted in figure 6.15.

The value of the angle χ of the valine 111 oscillates between the value of the two rotamer after 3 millions MC steps. Careful examination of figure 6.14 shows that the change in the dihedral occurs when the configurations generated at 473 K are exchanged with the configuration generated at 298 K. So the rotamer problem can be overcome by the use of a PT[59,60] simulation. Figure 6.15 shows that both rotamers are present at 25°C. Only the appropriate rotamer for the ligand was expected at 25°C. This is related to some

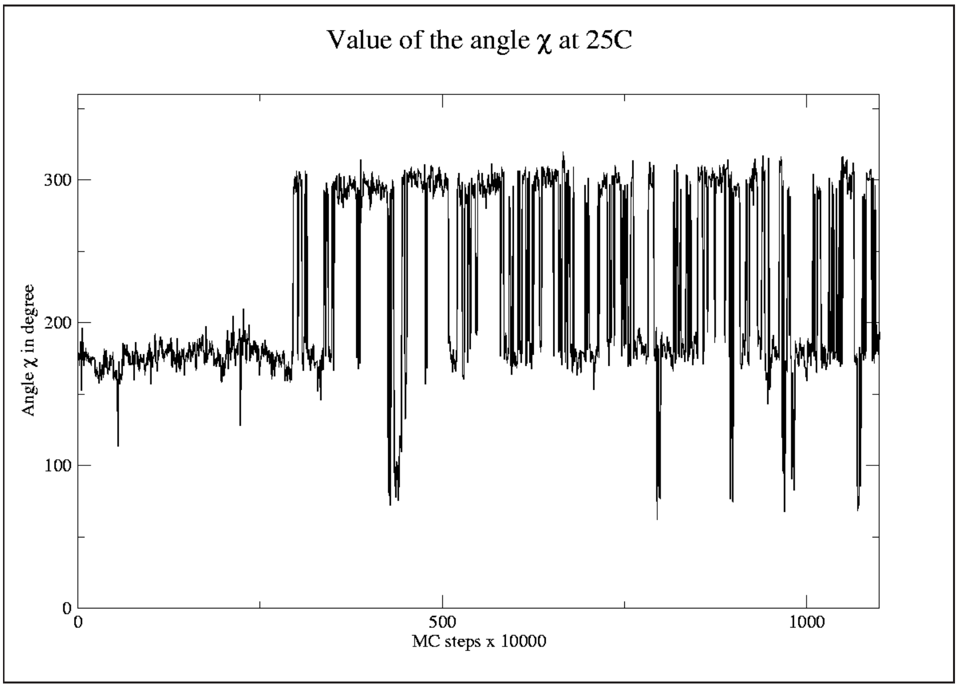


Figure 6.15: Value of the angle χ of the valine 111 in degrees for the simulations starting with the 185l crystal structure and the isobutylbenzene at 25°C.

issues with the forcefield, that failed [*] to capture all the changes in the valine conformation.

The next section will focus on computing the relative binding free energy between the two ligands and how the rotamer issue has been addressed during such computations (using for example the results of the PT[59,60] simulations). The relative binding free energy for the whole sets of ligand will be investigated as well.

[*] The term failed however might not be correct from a semantic point of view. The two different conformations appears to be have the same weight. The force field sees each of them as being statistically relevant and do not sample one preferably over the other.

6.2.3 Free Energy perturbation

Relative binding free energy between indole and isobutylbenzene

To complete previous work on the lysozyme the relative binding free energy between the isobutylbenzene and the indole have been computed for both crystal structures:

- Starting from the 184l crystal structure mutating the isobutylbenzene to the indole.
- Starting from the 184l crystal structure mutating the indole to the isobutylbenzene.

The RETI[57,58] method has been used to compute the relative binding free energy between the indole and the isobutylbenzene. Before performing the simulations, the system was equilibrated 50000 MC steps with both ligands present in the binding pocket and a λ of value 0.5 has been run. The final configuration of the equilibration run was scattered across the twelve values of λ and used as a starting configuration to compute the relative binding free energy. Simulations were performed using the optimised protocol for sampling discussed above (fixed residues, rigid unit backbone moves outside the F-loop and CRA moves used between the residues 101 to 123) and the dual topology method[57,58]. 10 RETI[57,58] moves each of 150000 steps were performed. Each simulations has been repeated 3 times. Twelve λ windows were used to perform the RETI[57,58] perturbation: (0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 0.95, 1.00). To be able to compute the relative binding free energy according to the figure 6.4, the ligands have been perturbed each into the other in GBSA. The results of the three runs were averaged and are displayed in the table 6.4.

Table 6.4 shows that starting from the 184l structure gives a relative binding energy in good agreement with the experimental values. The relative binding free energy between the isobutylbenzene and the indole is only 0.8 kcal/mol higher with a standard error of 0.6 kcal/mol. On the other hand, the perturbation from the indole to the isobutylbenzene starting from

Starting structure	Experimental values	MD simulations[25]	MC simulations using CRA
184l	1.6	5.4±0.3	2.4±0.6
185l	-1.6	-5.4±0.3	2.1±0.6

Table 6.4: $\Delta\Delta G_{bind}$ between the indole and the isobutylbenzene with standard errors (average of the 3 simulations). Values are in kcal/mol.

the 185l crystal structure, underestimates the relative binding affinity of the isobutylbenzene by 3.7 kcal/mol.

Literature states that the initial conformation is of critical importance for the results of the calculations[25,119] and the presence of the wrong rotamer can bias the computational value of the relative binding free energy by up to 4 kcal/mol[119]. Most important is that valine 111 plays a key role in the sampling of the F-loop. This hypothesis and the hydrophobic cluster discovered during the previous simulation would explain the over-estimation of the relative binding free energy.

Side-chain moves were modified so that the dihedral angle χ of the valine 111 was allowed to move freely between $-\pi$ and π and this specific move was apply to the RETI simulations (both sets) starting from the original crystal structure to try to reproduce experimental results and observe a change in the conformation of the side chain of the valine 111. However results were non conclusive. The valine 111 retained its original conformation and the relative binding free energies obtained are still within the range of standard errors from the previous simulations.

To see if more accurate results could be achieved, both set simulations were re-run, using the appropriate rotamer of the valine 111 with respect to the final ligand and the same protocol as before. The angle χ of the valine 111 was manually changed to the expected value and the remainder of the protein was unchanged. The relative binding free energy from the isobutylbenzene to the indole has been reduced so that the experimental value of the relative binding energy is only 0.2 kcal/mol lower to the computed value. Changing manually the rotamer of valine 111 on the 185l to accommodate the isobutylbenzene has proved to be even more successful[25]. The relative binding free energy of the mutation indole to isobutylbenzene has been reduced from 2.1 kcal/mol to -1.0 kcal/mol (see table 6.5).

	Experimental results	MC/CRA	MC/CRA on the rotamers	MC/CRA PT
184l	1.6	2.4±0.6	1.8±0.6	-
185l	-1.6	2.1±0.6	-1.0±0.6	-2.9±0.7

Table 6.5: $\Delta\Delta G_{bind}$ between the indole and the isobutylbenzene with standard errors (average of the 3 simulations). Values are in kcal/mol.

In the previous section we have investigated the effect of using the PT[59,60] method on the side chain of the valine 111. The PT[59,60] have been proved to enable the rotation of the valine 111 side chain to the appropriate position. Twelve configurations of the system at 298 K were chosen from the PT[59,60] simulation and used as starting configurations for the RETI[57,58] simulation using the 185l crystal structure to see if the configurations from the PT run could overcome the rotamer issue. The use of configurations drawn from the PT[59,60] simulations lead to a decrease of the relative binding free energy from 2.1 to -2.9 kcal/mol (with a standard error of 0.7 kcal/mol). The relative binding free energy is still overestimated by around 1 kcal/mol in favour of the isobutylbenzene but the ranking order is in agreement with the experiment (isobutylbenzene is more likely to bind than indole). Relative binding free energies obtained using different rotamers or simulations techniques to enhance the rotamer sampling are summarised table 6.5. Figure 6.16 shows the value of the dihedral χ of valine 111 for the simulations where the indole is mutated into the isobutylbenzene for λ equal to one.

Figure 6.16 shows that whereas values using the minimised crystal structure do not sample the change in the dihedral (average value of χ is 180 degree), the use of random configurations drawn from the PT[59,60] simulations ables the system to jump over the rotational energy barrier, allowing the dihedral to sample more configurations using its appropriate value (300

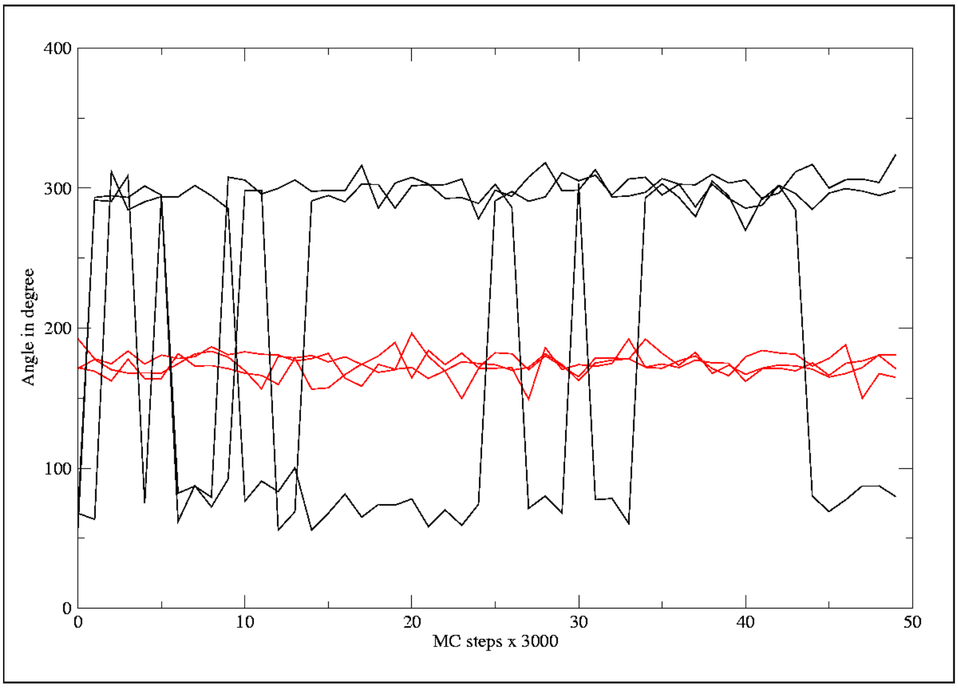


Figure 6.16: Value of the angle χ of the valine 111 in degrees for the simulations starting with the 185l crystal structure (indole mutated into isobutylbenzene) for $\lambda = 1$. Simulations using random configurations drawn from the PT[59,60] run are in black. Simulations starting with a minimised crystal structure are in red.

degree). The value of 180 degree is not sampled when the configurations drawn from the PT are used. This suggest either a problem with the crystal structure (as a methyl group is only nine electrons) or with the forcefield.

Relative binding free between benzene and the whole set of ligands.

The relative binding free energy between the benzene and the whole set of ligands used in the literature[25,119] has been computed to see if the results of the indole to isobutylbenzene simulation could be reproduced. A 15 Å scoop centred on the isobutylbenzene of the 181L crystal structure was used (same scoop as in the previous section with the same simulation protocols) for all the simulations. A first batch of 50 RETI moves was run, extended to 150 RETI moves. Relative binding free energies over the three simulations have been averaged and results can be found in table 6.6.

	181L		184l	185l	$\Delta\Delta G^{bind}_{exp}$
RETI moves	50	150	50	50	
Benzofuran	-3.3±0.5	-3.3±0.3	-4.7±0.4	-4.4±0.4	-0.3±0.0
Indene	0.272±0.5	0.4±0.3	-2.4±0.4	-1.5±0.4	-0.1±0.0
Isobutylbenzene	6.0±0.7	5.9±0.4	0.5±0.6	4.0±0.7	-1.3±0.0
Indole	-4.8±0.5	-5.1±0.3	-7.1±0.4	-6.1±0.5	0.3±0.0
n-butylbenzene	11.7±0.8	10.7±0.5	7.0±0.9	10.8±0.8	-1.5±0.0
o-xylene	N.A.	2.3±0.3	-0.3±0.4	0.8±0.5	0.5±0.0
p-xylene	N.A.	2.5±0.3	-0.3±0.6	1.9±0.5	0.5±0.0

Table 6.6: Relative binding free energy between the benzene and a set of ligands, with standard errors. First row display the PDB name of the structure used for the perturbation. Each RETI move is composed of 30000 MC steps. Energy is in kcal/mol.

The computed relative binding free energies do not reproduce the experimental results. This is probably due to the difference in the shape of the ligands. Starting from the 181L crystal structure, the F-loop has to undergo major changes in its conformation to adapt to the ligand as the binding pocket for the benzene is the smallest. Then the simulations were run using other starting crystal structures (184l and 185l) with larger cavities.

However, these simulations stressed the fact that the initial structure of the protein seems to be of a critical importance in the computation. The results of the calculated relative binding free energies are very sensitive to the initial structure and can be changed by up to 4.7 kcal/mol in the extreme case of the n-butylbenzene, depending on the shape of the starting structure and the rotamer of valine 111.

The use of the 184l crystal structure always lowers the relative binding free energy between the benzene and the other ligands. One probable explanation for such phenomena is the presence of the rotamer of the valine 111. This has been described in the literature by Soichet *et al* as a factor of errors up to 4 kcal/mol in the computed binding free energies[120].

Another explanation is the size of the binding pocket. The main reason being that the larger pocket ables the system to avoid high energy configuration due to repulsive effects. To accommodate the isobutylbenzene, the binding pocket is bigger than in the 181L crystal structure. So more space is available for the ligand to sample the cavity, thus increasing the sampling and leading to better results.

6.3 Concluding remarks

The CRA algorithm has been applied to the lysozyme protein to try to sample the F-loop. The CRA successfully provides enhanced sampling of the backbone for the F-loop. The complete interconversion of the F-loop when ligands are crossed is not observed, although the CRA samples the possible configuration of the loop with efficiency. In the case of the lysozyme, the size of the binding pocket is not the only parameter to consider. Other parameters such as the side chain of the valine 111 and the position of the ligand in the binding pocket have a great effect on the sampling. Methods to enhance the backbone sampling such as CRA have little effect on the side chains.

The rotamer issue was overcome using the PT[59,60] method. By using PT[59,60], high temperature configurations were brought down to 25 °C allowing the appropriate conformation of the valine 111 to be sampled. This method however is very expensive.

The relative binding free energies between the benzene and the whole set of ligands were different from the experimental values. This raises some issues. Are the results poor due the non-bonded parameters, or is it only the case of sampling the valine and the F-loop? As the results are greatly influenced by the starting conformation and the rotamer of valine 111, the issues of the rotamer and the F-loop conformation seem to be the most likely to influence the results.

These issues seem to have been solved using the PT[59,60], so running the whole set of ligand with configurations drawn from the PT[59,60] simulations would probably give more accurate results, but at a very expensive cost.

The next chapter is going to discuss the effects of using the CRA algorithm and GBSA solvation in biological systems where loop sampling is of critical importance.

Chapter 7

Biological systems

In this chapter, the use of the CRA algorithm on two different biological systems will be discussed. Proteins such as kinases and phosphodiesterases which undergo major changes in a conformational loop will be investigated using the CRA algorithms. Both systems have proved to be a challenge for standard computational methods.

7.1 Kinases

Kinases are one of the most important classes of enzyme in human physiology (kinases constitute almost 2% of the human genome) and are critical to the transmission of signals both within and between cells. They are widely studied in cancer therapeutics[*].

7.1.1 Kinases, function and conformation

Protein kinases function as components of signal transduction pathways, playing a central role in diverse biological processes such as control of cell growth, metabolism, differentiation and apoptosis. During cancer, many kinases are not able to function properly leading to eternal activation of kinases

[*] This chapter does not aim to give a complete overview of kinase structure and function. The reader is referred to the work of Fabbro for further information[127].

such as Bcr-Abl responsible for chronic myelogenous Leukemia. All kinases share a common fold of around 250 residues known as the kinase core[128] that contains the binding pocket and the phosphorilation site. Several crystal structure of the common fold are available in the PDB database (the key word kinase gives more than 2700 hits). Several drugs exist on the market[129] giving insights into the mechanism of inhibition.

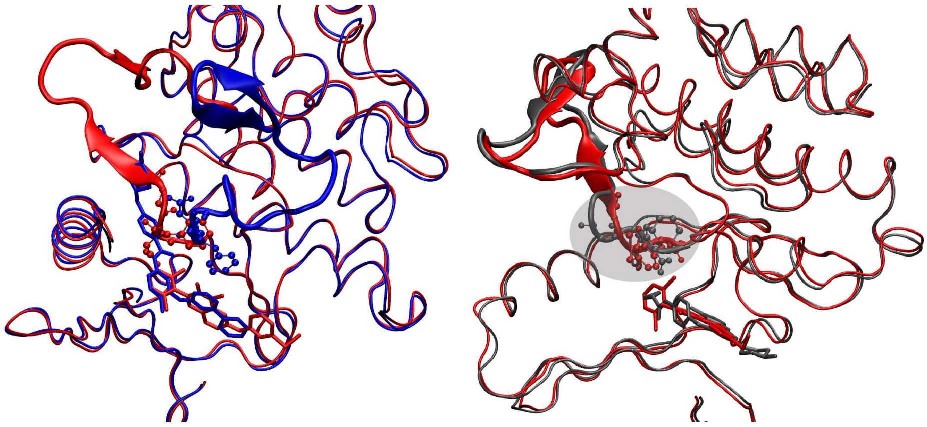
Tackling the kinase problem using computational methods has proved to be difficult due to several key points in the kinase structure. The activation loop undergoes major displacement during the activation process. A domain reorganisation then occurs, triggered by the activation of the kinase and then, the DFG loop (part of the activation loop and involved in the binding of the ligand) adopts a different conformation. The following sections, will describe the work performed to try to shed light on the mechanisms involved in the change of both the activation and DFG loops.

Activation loop in the Bcr-Abl Kinase.

Sampling conformational changes in the activation loop of kinases is of major importance and could illuminate the mechanisms related to the activation or de-activation of kinases.

Several crystal structures of mutant of the Bcr-Abl kinase exist (PDB databases 1iep[130], 1m52[131], 2f4j[132], 1opj[133], 1fpu[134]). All the structure are different in geometry and function. The activation loop is present in both forms (in and out), the DFG loop adopts either of the two known conformation and the kinases are present in both active and non-active forms. To add to the problem, mutations such as H396P and T315L (the later referred as the gate keeper) have been reported. Figure 7.1(a) shows the difference in the activation loop between the 1iep and 1m52 crystal structures, and figure 7.1(b) shows the difference in the conformation of the DFG loop between the 1m52 and 2f4j crystal structures.

Sampling the conformational changes of the activation loop or the DFG loop for the Bcr-Abl kinase using modelling methods should be challenging. Domain reorganisation presents an even greater challenge due to the am-



(a) Superposition of the 1iep (blue) and 1m52 (red) crystal structures.

(b) Superposition of the 1m52 (red) and 2f4j (grey) structures.

Figure 7.1: Representation of the 1iep, 1m52 and 2f4j crystal structures with activation loop (cartoon representation), DFG loop (CPK representation) and ligand (licorice representation). Difference in the DFG between the 1m52 and 2f4j crystal structures is highlighted in grey.

plitude of the change from both a geometrical and temporal point of view. The Bcr-Abl presenting the T315L or the H396P mutation are known to be resistant to the action of the Abl inhibitor imatinib (STI-571 or gleevec[135]) and understanding the effect of the mutant on the reorganisation process of the activation loop may lead to better drug design.

Being able to use MC simulations to solve one or several of the issues raised above would represent a major breakthrough in computational science. However such a herculean task will requires extensive amount of resources and more time than one (or maybe several) PhD could provide. To address the effects of the mutations, one would need to be able to mutate the residues in the protein whereas investigating the domain reorganisation would need a coarse grain approach to the problem[136–139] due to the time scale and the number of degrees of freedom changed.

Rather than trying to tackle all the issues related to the kinases, we have first tried to apply the CRA algorithm to some of the kinase conformational problems.

7.1.2 Use of CRA on the Bcr-Abl Kinase

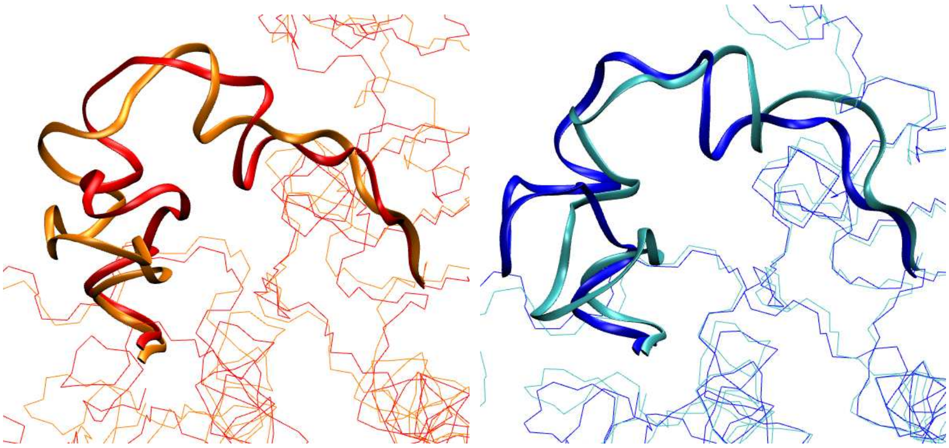
Conformational sampling

The CRA algorithm has been used to increase the sampling of MC simulations for the 1iep and 2f4j kinases without ligand at 298 K. Results have been compared to existing MD simulation performed *in-situ* in our lab*. For both systems, the holo protein has been sampled using MC and MD simulations. By removing the ligands from the binding pocket we expect to see some changes in the conformation of the activation loop. MD simulations were run in explicit solvent with TIP3P water molecules[61] and the AMBER forcefield[39,40]. The simulation was run in 200 blocks of 0.1 ns each with a time step of 2 fs due to the use of the SHAKE algorithm[48] to constrain the bonds involving hydrogens. The cut-off for electrostatic interaction was set to 11 Å and the Particle Mesh Ewald (PME) was use for the long distance interactions. MC simulations consists in 4500 blocks of 10000 MC steps using implicit solvent (GBSA) and run in the NVT ensemble. CRA moves were used between the residues 383 to 409 (activation loop) and the other residues of the protein were moved using standard ProtoMS[51] moves (see section 3.3.1). One CRA move was performed every 4 moves. Cut off for electrostatic interaction was set to 10Å.

To compare the efficiency of MD and MC methods, one has to rely either on CPU time or on the sweep method. Owing the difference in the solvent modelling (TIP3P[61] for the MS simulations and GBSA[65,140] for the MC) the comparing CPU time will not be accurate and hence the sweep method will be used†. One MC sweep corresponds to the number of MC moves to statistically move all the residues of a system once, being then equivalent to one MD time step. To do so, the assumption is made that at each MC move, a different residue is moved. This might not be the case for one sweep,

*Courtesy of Miss Clapton

†Comparing CPU time also implies the use of similar machines. Southampton University's local cluster Iridis is made of 12 switches each with different types of processor.



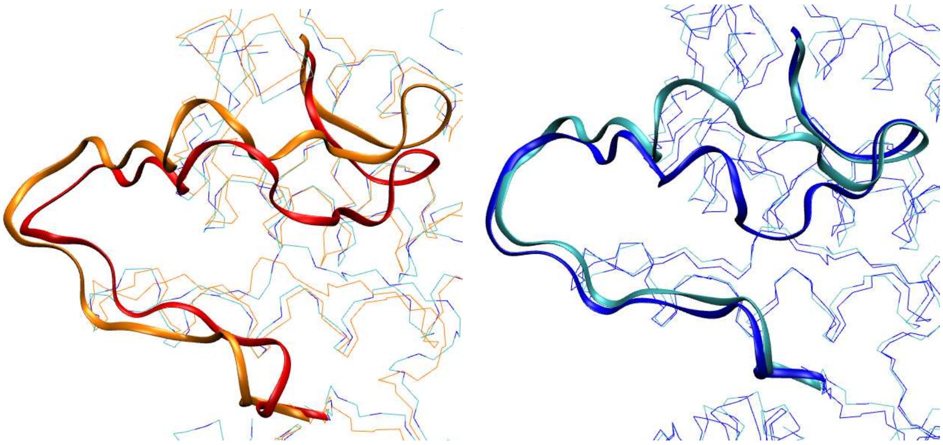
(a) Superimposition of the first (red) and last (orange) structure for the MD simulation.

(b) Superimposition of the first (blue) and last (cyan) structure for the MC simulation.

Figure 7.2: Superimposition of the first and last structures of the simulation for the 1iep crystal structure without ligand.

but the ergodicity of the system tells us that over the great number of steps of one simulation this becomes true. However comparing the sweeps is not as accurate as comparing CPU times, as the size of the sampling has to be accounted for.

One CRA move changes the coordinates of 5 residues and one standard ProtoMS[51] move changes the coordinates of 2 residues. Owing to the move probabilities, every 4 MC moves, 11 residues have their coordinates changed. The proteins have 274 (1iep) and 287 (2f4j) residues. I need to perform 100 MC moves for the 1iep to move all the residues in the protein (104 for the 2f4j). So one MD move corresponds to a sweep of 100 MC moves. If I want to use the sweeps to compare the sampling achieved with both methods, I would need to run 100 times 10 million MC step. That represents a one billion step trajectory. Such a vast number of steps is not achievable using ProtoMS[51], as the code is not build to be parallelised. However, confident in the use of the CRA algorithms, we have decided to run one 45 million step trajectory for each structure first and compare the level of sampling of



(a) Superimposition of the first (red) and last (orange) structure for the MD simulation.

(b) Superimposition of the first (blue) and last (cyan) structure for the MC simulation.

Figure 7.3: Superimposition of the first and last structures of the simulation for the 2f4j crystal structure without ligand.

the activation loop. For both structures, initial and final structures have been superimposed, and are displayed figures 7.2 and 7.3. Once the structures have been superimposed, the RMSDs of the activation loop between the starting and the final configurations for both structures have been calculated. Results are given in table 7.1.

Crystal structure	MD simulations	MC simulations
1iep	3.29	3.83
2f4j	3.01	2.63

Table 7.1: RMSDs of the activation loop between the initial and final structure of the simulations. RMSD are in Å and calculated after the superimposition of the two configurations.

Table 7.1 shows that both techniques give similar RMSD. However, several points have to be clarified. The number of sweeps performed using MC simulations is twenty times smaller that what it should be. The starting structures are different. Whereas the MD starting structures have been minimised without a ligand, the structures used in the MC simulations were minimised with the ligand inside. Thus, the starting conformation for the MC simulations is biased toward the bound state. So the use of MC simulations using CRA manage to sample the activation loop of the kinases with the same efficiency as MD, but manage it faster and in this particular case, are less sensitive to the starting structure, as the penalty introduced by the difference in conformation is easily overcome during the MC simulations (however MD simulations have to deal with the explicit representation of the solvent whereas MC simulations were performed using GBSA).

To try to get more information about the conformational changes, the g_cluster tool from the gromacs package has been used for both methods on both systems[141,142]. For the MC simulations, conformations were saved every 10 000 steps and used as a trajectory. For the MD simulations, conformations were saved every 10000 steps (20 ps) and used as a trajectory. Owing to the difference in the length of the simulations for both methods, the number of snapshots in the MD trajectories is twice the number of snapshots in the MC trajectories. Different values of the cut-off and two different methods to compute the distance distance values for the RMS matrix have been used. The single linkage method has been use to build the clusters. The number of clusters identified are given table 7.2 for the MC simulations and table 7.3 for the MD simulations.

For each simulations, the RMS matrix has been computed using two different methods. The first one, by computing the RMSD of the distances (column indexed RMSD in table 7.2) and the second one, by computing the RMS deviation after fitting (column indexed RMS in table 7.3) when building the RMS matrix. For each of the two methods, the clusters have been calculated using both the backbone and all the atoms of the activation loop.

Table 7.2 and 7.3 show that there are very few clusters for the different conformations of the activation loop for all but small values of the cut-off when only the atoms of the backbones are used. Furthermore, the number of clusters obtained from the MC simulations is greater than the number obtained from the MD simulation. This trend is inverted for the number of

	2f4j				1iep			
	Backbone		All atoms		Backbone		All atoms	
	RMS	RMSD	RMS	RMSD	RMS	RMSD	RMS	RMSD
1	1	1	1	1	1	1	2	1
0.7	1	1	6	1	13	1	18	1
0.6	10	1	125	3	145	7	167	5
0.5	191	5	414	91	406	56	438	72

Table 7.2: Number of clusters for the activation loop of the 1iep and 2f4j MC simulations. The values of the cut-off are represented in the first column.

	2f4j				1iep			
	Backbone		All atoms		Backbone		All atoms	
	RMS	RMSD	RMS	RMSD	RMS	RMSD	RMS	RMSD
1	1	1	9	3	1	1	1	1
0.7	7	1	704	717	1	1	194	143
0.6	63	2	982	992	4	1	727	745
0.5	298	38	1001	1001	74	1	953	999

Table 7.3: Number of clusters for the activation loop of the 1iep and 2f4j MD simulations. The values of the cut-off are represented in the first column.

clusters obtain when all the atoms of the activation loop are used, probably due to the method used to sample side-chains. However, for the MC trajectories, the number of cluster for a given calculation is greater for the 1iep simulation, meaning there are more conformational changes during the simulation confirming the results from table 7.1. Another interesting point, is that due to the use of rigid backbone unit moves outside the activation loop, the brownian motion of the protein (drift) is less important for the MC simulations and such behaviour could explain the number of clusters for the 2f4j kinanse.

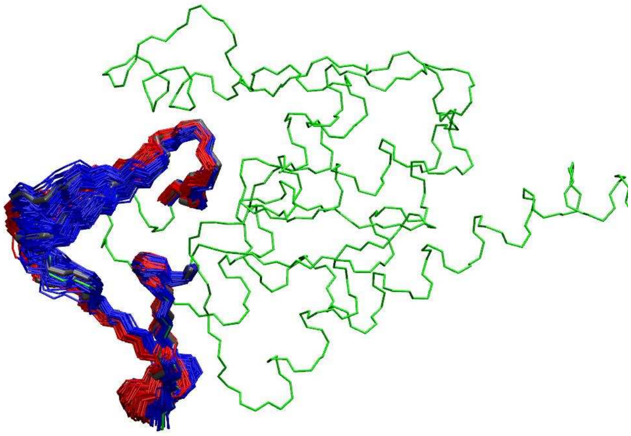
Figure 7.4 shows the main clusters for both MC simulations (clusters in blue in table 7.2) For the simulations starting from the 2f4j and 1iep structures without ligands, 3 of the 6, and 6 of the 18 clusters are represented respectively (the most representative clusters during the trajectories). The clusters and the step number of the trajectories are related. Clusters appear sequentially along the trajectory and do not return, characterising a displacement of the activation loop. This shows again the important changes of conformation undergone by the activation loop during the simulations due to the use of the CRA moves, where the level of sampling can be compared to the MD method.

The CRA algorithm managed to enhance the sampling of the activation loop, however the complete interconversion of the loop is quite demanding in terms of CPU time. An interesting challenge would be to convert the DFG loop between two structures bound to different substrates by mutating one ligand into another.

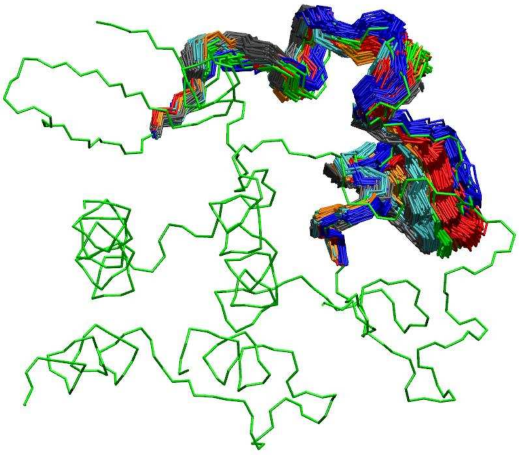
Free energy perturbation

Figure 7.1(b) shows that the superimposed structures of the 2f4j and 1m52 proteins share the same conformation of the activation loop but a different DFG loop conformation.

However the primary structures are different. The 2f4j crystal structure presents the H396P mutation. The modeller tool[143] has been used to mutate the residue 396 of the 1m52 crystal structure into a proline. To observe a change in the conformation of the DFG loop, both VX6 (for 2f4j) and P17



(a) Clusters for the 2F4J simulation.



(b) Clusters for the 1IEP simulation.

Figure 7.4: Superimposition of the clusters for the simulations with both 2f4j and 1iep crystal structure without ligands. Clusters have been computed using all the atoms of the activation loop, a 0.7 Å cutoff, the RMS deviation on the fitted structure and the single linkage method. Initial structures are represented in green.

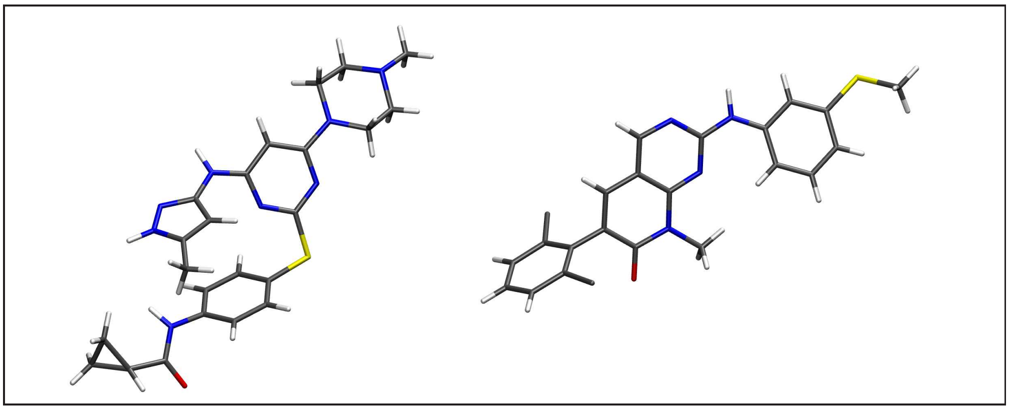


Figure 7.5: VX-6 (right hand side) and P-17 (left hand side).

(for 1m52) ligands have been perturbed one into each other. Figure 7.5 shows the geometry of both ligands.

The two crystal structure have been minimised prior to the RETI simulations. For both crystal structure the same protocol has been used. Five hundred cycles of minimisation in GBSA using the amber forcefield have been performed. Then a scoop of 15Å around the VX6 compound with a inner sphere of 10Å as been created, residues in the outer sphere have been altered so the the total charge of the system was lowered to zero. For each scoop an equilibration of 50000 MC steps at 0 K and 298 K in GBSA using dual topology with both ligands present and a λ of 0.5 have been run to remove the most important steric clashes. For each of the two crystal structures, two set of simulations have been run, each of three RETI simulation starting with a different random seed using. For the first set, the CRA move as described in the literature[83] has been used on five residues. The DFG loop and the first neighbouring residues on both sides. For the second set of simulations, the CRA with the extended prerotation move as described in section 5 has been used on the same five residues. Each set was composed of 50 RETI moves of 30000 steps using the dual topology method and the values of λ were scattered between zero and one identically to the values used for the lysozyme simulations (see section 6.2.3).

None of the twelve simulations managed to give an accurate value of the relative binding free energy. The $\Delta\Delta G_{bind}$ is several hundreds of kcal/mol negative going from P17 to VX6 in the 1m52 protein (ranging from -570 kcal/mol to -415 kcal/mol). For the perturbation VX6 to P17, the relative binding free energy is two orders of magnitudes higher (but with a positive value). These results however are not very surprising. Figure 7.6 shows the last structure of three RETI simulations at $\lambda = 1$ for both perturbations (VX6 to P17 in 2f4 and P17 to VX6 in 1m52) and the RMSD for respective simulations are plotted figure 7.7.

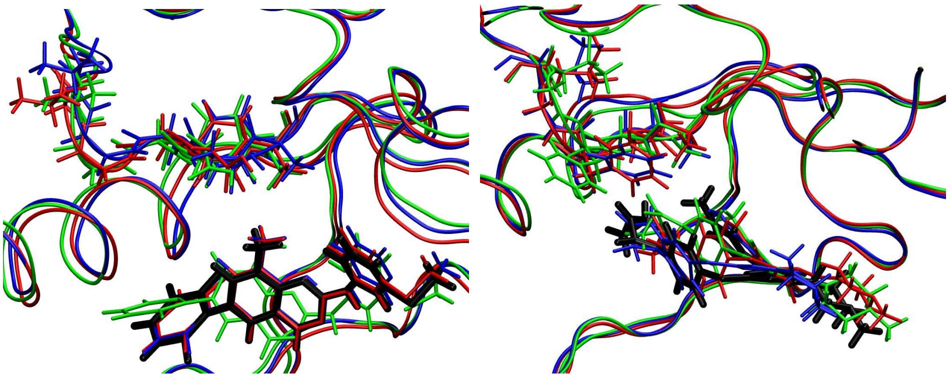
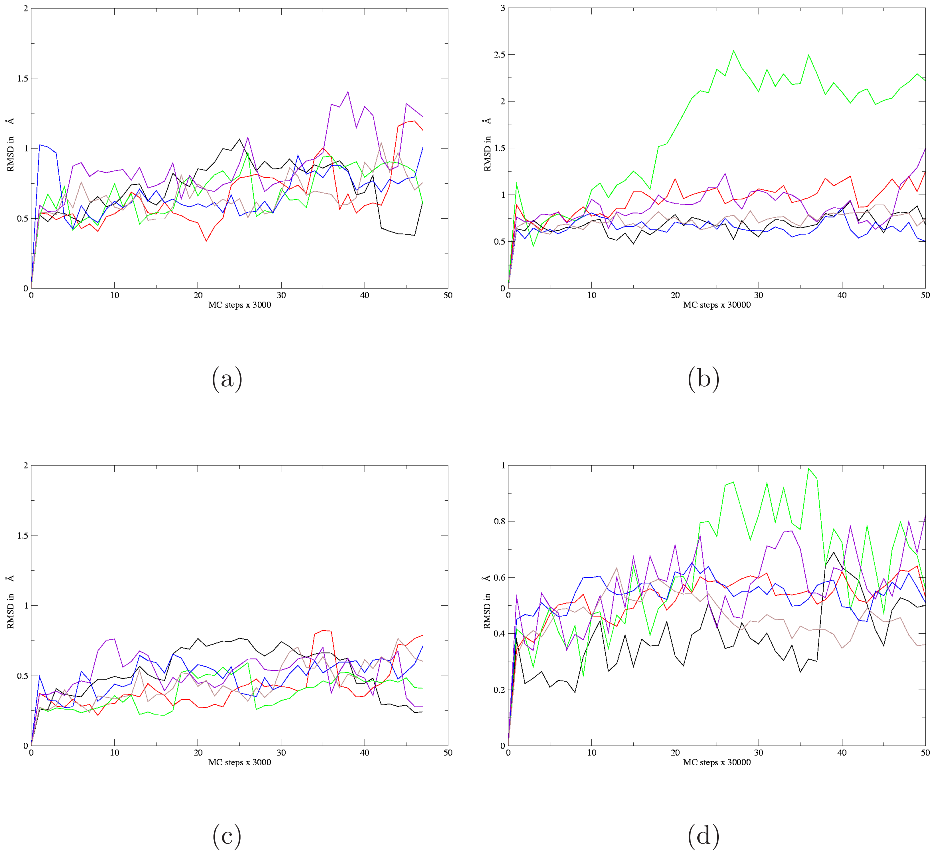


Figure 7.6: Superimposition of the last structures at $\lambda = 1$ for 3 RETI simulations starting from the 2f4j (left) and 1m52 (right) crystal structures. All the atoms of the DFG loop and the ligand for $\lambda = 1$ are represented. The initial conformation of the ligands are represented in black.

Figure 7.6 stresses several points. The sampling of the backbone of the DFG loop does not allow the interconversion of the conformation of the loop. This is true for simulations starting from both crystal structures. The ligands at $\lambda = 1$, do not undergo major changes of conformation. The two issues can be linked together to explain the results of the relative binding free energies. The DFG loop cannot sample sufficient phase space, hence the existence of de-favourables interaction with the ligands. Such interactions seem to have more effect mutating the VX6 into P17 in the 2f4j crystal structure than mutating P17 into VX6 in the 1m52 crystal structure. This is simply due to the initial conformations of both ligands and proteins and the steric clashes resulting from such conformations.

Owing to the size of the ligand and the conformation of the DFG loop, sampling for the RETI simulations starting from the 1m52 crystal structure is more important than the sampling achieved in the RETI simulations starting from the 2f4j crystal structure.



(a)

(b)

(c)

(d)

Figure 7.7: RMSDs of the DFG loop for the RETI simulations starting from the 2f4j(left) and 1m52 (right) crystal structures at $\lambda = 1$. Top row represents the RMSDs with all the atoms and the bottom row represents the RMSDs for the atoms of the backbone only.

Figure 7.7 shows good sampling of the DFG loop. However, the sampling of the DFG loop is not sufficient to achieve the necessary interconversion. The RMSD between the two crystal structures for the DFG loop is 5.52 Å for all the atoms and 2.99 Å for the backbone atoms only.

Figure 7.7(b) shows that the RMSD for one of the RETI simulations is quite different from the others (green curve figure 7.7(b)). This is mainly due to changes in the conformation of the side chain. Figure 7.7(d) shows the RMSD for the atoms of the backbone only and the RMSD (green curve) is not that dissimilar to the other simulation being however different. The green conformation in figure 7.6(b) shows that the side chain of the phenylalanine in green (simulation corresponding to the RMSD plotted in green figure 7.7(b) and figure 7.7(d)) adopts a different conformation than for the others simulations, explaining the increase in the RMSD.

The use of the CRA algorithm has shown significant increase in the sampling of the activation loop of the Bcr-Abl kinases. Nevertheless, the increase of sampling is not sufficient to sample the complete opening of the loop or the interconversion of the DFG loop during free energy calculations. However, such changes in conformations can not be observed using MD methods either. The size of the change and the resources available seem to draw a limit to the use of the CRA algorithm.

Next we have have applied the CRA algorithm to the PDE5 class of phosphodiesterase to try to compute accurate relative binding free energy between the commercial drugs viagra and cyalis.

7.2 PDE5

Phosphodiesterases are a large class of enzymes mediating a number of physiological processes ranging from immune response to platelet aggregation to cardiac and smooth muscle relaxation. In particular, phosphodiesterase 5 (PDE5) plays an important role in mediating sexual arousal, and it is the central molecular target in treatments of erectile dysfunction.

7.2.1 Protein function and structure

Phosphodiesterases usually hydrolyse the second messengers cyclic guanosine monophosphate (cGMP)and cyclic adenosine monophosphate (cAMP) which are key components in the transduction cascades. By reducing the cellular level of cGMP and cAMP, phosphodiesterases regulate the mechanisms described above[144].

There are 11 classes of phosphodiesterases; the class 5 (PDE5) is involved in mediating sexual response. Several drugs are known to bind to PDE5, most famous being sildenafil (viagra) and vardenafil (cyalis) (both represented in figure 7.8).

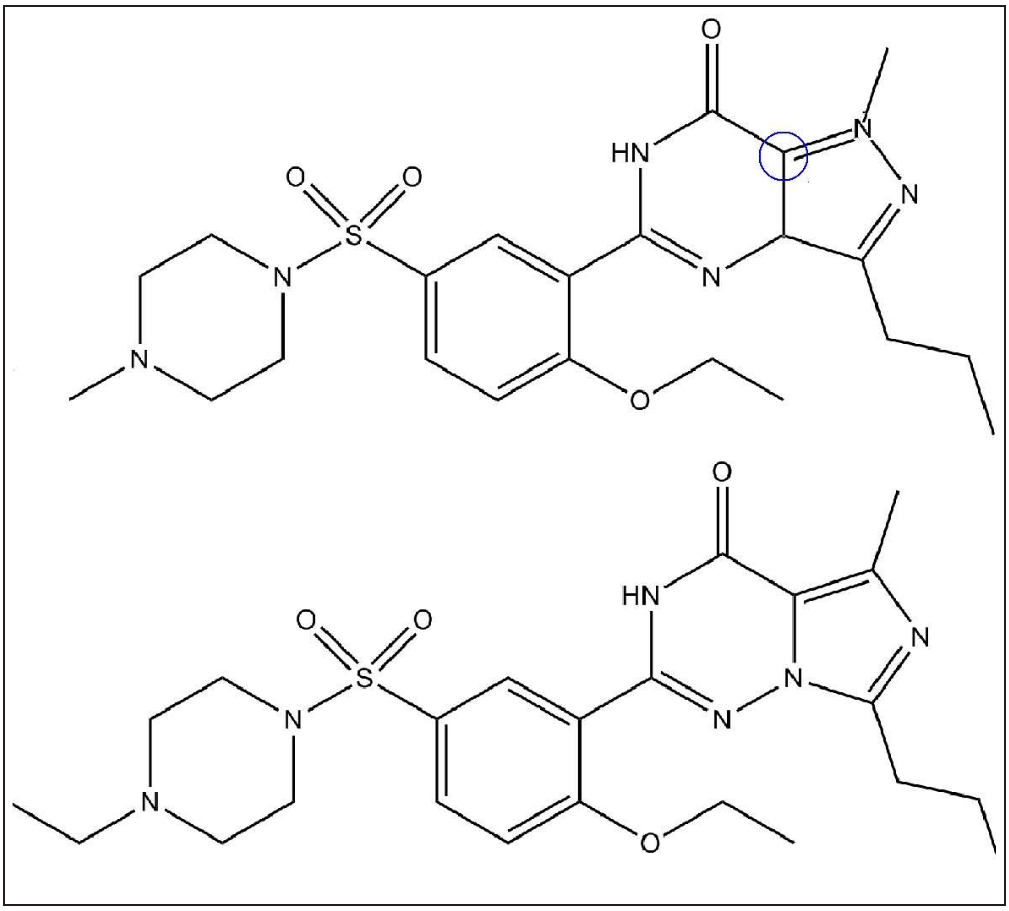


Figure 7.8: Sildenafil (top) and vardenafil (bottom).

These drugs have been designed so that the cross-reactivity with other families of phosphodiesterases is very low so they mostly target the PDE5 proteins[145–147] and both drugs have similar structures and the conformation of the bound state of the protein with both drugs is very similar.

However the vardenafil binds the PDE5 protein about 30 times tighter

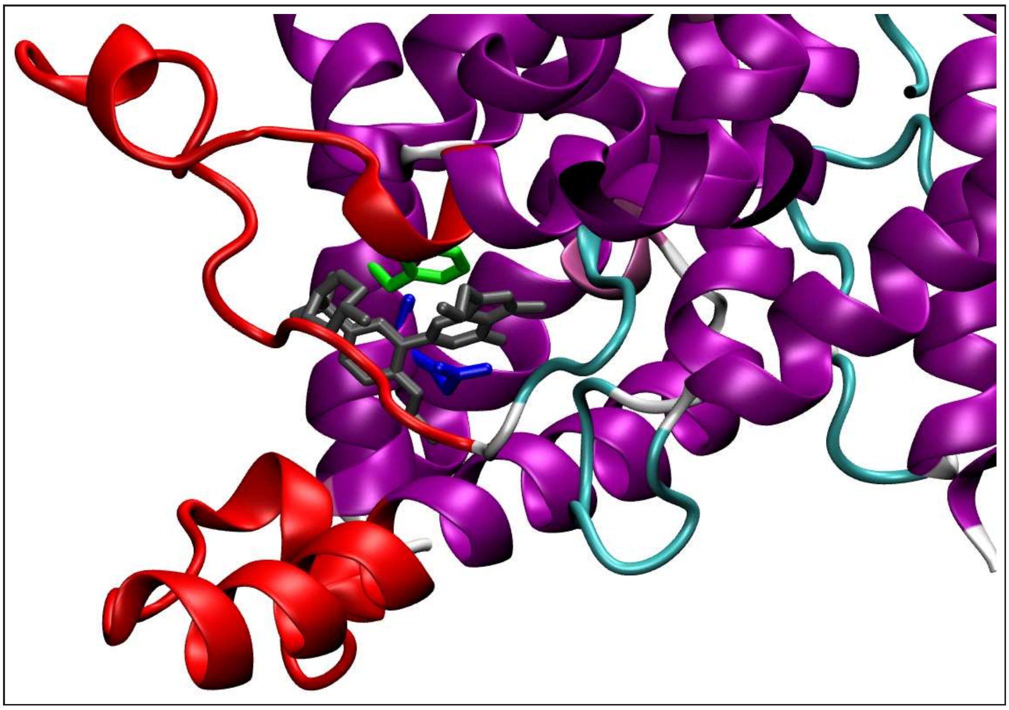


Figure 7.9: Structures involved in the binding mechanisms of the sildenafil (grey). Glutamide switch (green), hydrophobic clamp (blue) and loop clamp (red).

than the vardenafil to the PDE5 protein catalytic domain. Two binding interactions have been reported in the literature; the glutamide switch[148] and the hydrophobic clamp[149]. Furthermore to the existing binding modes, Zagrovic *et al.* quotes a binding mechanisms in which both the H and M loops of the protein execute sizable conformational changes[150] called the ”*loop clamp*”. Figure 7.9 shows a representation of the various binding modes for the sildenafil in the PDE5 protein.

The PDE5 protein have been previously studied using MD methods[150]. Zagrovic *et al.* has performed several simulations on this system. Simulations where run on the 1udt crystal structure. The missing part of the H-loop where added using the modeller package[143] and ten MD simulations of 3 ns each using different starting velocities from the Maxwell-Boltzmann distribution at 300 K were run. Simulations were run using the GROMOS 45A3 forcefield[151] and the GROMOS package[141]. Protein was solvated using a truncated octahedron box filled with SCP[152] water molecules. Thermodynamical integration was performed using 26 values of λ equidistant between 0 and 1. For each λ 500 ps of simulation was carried out, the fist 100 ps used to equilibrate the system. Summary of the results of this work can be found section 7.2.3

The crystal structures of the PDE5 catalytic domain bound to both sildenafil[153] and vardenafil[154] are available in the PDB database under the references 2h42 and 1uho respectively.

Insight of the binding pocket of the protein illustrates the complexity of the system. The binding pocket contains the ligand, two divalent cations (Mg^{2+} and Zn^{2+}) and some crystallographic water molecules involved in ion coordination (see figure 7.10).

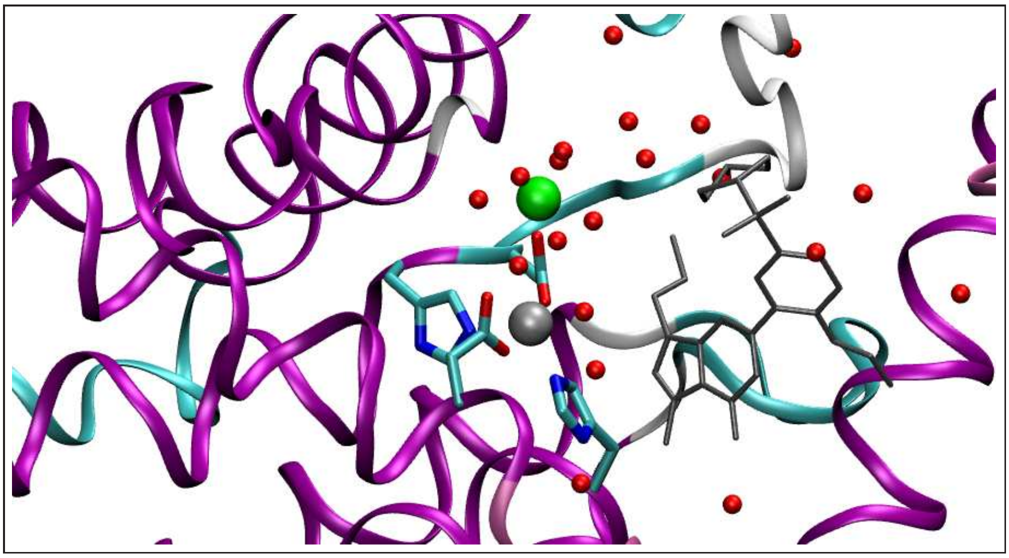


Figure 7.10: Binding pocket of the 2h42 crystal structure. The ligand is represented in grey, the oxygen of the water molecules in red, the zinc in grey and the magnesium in green. Also represented, the residues involved in the metal coordination.

The ions are bound to waters and several side-chains of the protein[154]. The zinc is coordinated to the side chains of His 617, Asp 654, Asp 764, His 653 and two water molecules. The magnesium is bound to five water molecules and the side chain of Asp 654. Three of the water molecules bridging the metal to the His 657, Asp 682 and His 685.

Such complexity raises several issues to assess during the equilibration of the system.

The next section will describe the parametrisation of the system in ProtoMS[51] to integrate all the variables relevant to the bodies present in the binding pocket.

7.2.2 Parametrisation

To study the PDE5 protein two crystal structures have been used, 1tbf and 2h42, both structure have been prepared in the same way. All the MC simulations have been run using GBSA solvation. However, GBSA in ProtoMS[51] is not parametrised to deal with cations such as magnesium or zinc. So the force field has to be parametrised as one cannot simply ignore the presence of the ions.

To make sure the correct parameters are chosen, literature was gathered to select the parameters to use with both ions[155–160]. Determining the Born radius for the zinc is not straight forward. The Born radius changes with the coordination state of the zinc. The value of the free energy of solvation for a single ion in a solvent continuum is given by:

$$\Delta G_{Born} = -\frac{q^2}{8\pi r}(\frac{1}{\epsilon_0} - \frac{1}{\epsilon}) \tag{7.1}$$

where q is the charge of the ion, r the Born radius and ϵ_0 and ϵ the vacuum and continuum permittivity.

The value of the Born radius as well as some scaling parameters are used by ProtoMS[51] to compute the energy of solvation for an ion. The Born radius values for both the zinc and the magnesium have been chosen from Babu *et al.*[155] (see table 7.4).

The values in the `gborn.parameter` file are not the values of the Born radii, but the values used by ProtoMS[51] with a scaling factor to compute the correct Born radii and hydration energies. To use the correct value in the simulation, code has been modified so the value of the Born radius and the

M^{2+}	$-\Delta G_{hyd}$(kcal/mol)		R_{BORN}(Å)	
Zn^{2+}	-467_{exp}	-477^b	1.40^a	1.4^b
Mg^{2+}	-437_{exp}	-433^b	1.50^a	1.5^b

Table 7.4: Absolute hydration free energy and Born radii for the zinc and magnesium cations; from the literature (*a*) and used in ProtoMS[51] (*b*).

absolute hydration energy were printed in the output files. Then trial and error for values of the parameters has been applied until the single point energy of one ion in the Born continuum was close enough to the experimental value (typically a value of the energy within 10 kcal/mol from the experimental value[69]). The Born radius used in ProtoMS[51] and the respective hydration energy for both ions can be found table 7.4.

7.2.3 Simulations

Once the parameters for both zinc and magnesium have been set to the correct value, the question of the crystallographic water remained. Some of the water molecules play an important part in the coordination of the cations[154]. These waters cannot be removed from the binding pocket, so a script has been written to build water shells using the position of the oxygens from the crystal structure. The water shell containing all the oxygens within a 5 Å radius centred around the ions and the ligand (sildenafil) has been built.

Then the oxygens were transformed into TIP3P[61] water molecules using the xleap tool from the amber[109] package. For a water molecule to be used in ProtoMS[51], its geometry has to be exactly a TIP3P or TIP4P one; no variation in the bond length or bond angle of the water are expected. One drawback of using xleap is that this module does not favour any hydrogen bonding. All the TIP3P molecules are orientated in the same way. However a short minimisation of the structure would correct this problem. At this point, the sander module cannot be used to minimise the structure as this would change the geometry of the water molecules. ProtoMS[51] has been used to minimise the system. A careful minimisation of the system (2h42 protein) was performed as follow:

- First 15 blocks of 10000 MC steps at 0 K have been performed sampling the whole system.
- Then 5 blocks of 10000 MC steps at 298 K have been performed sampling only the water molecules.
- And finally 5 blocks of 10000 MC at 298 K have been performed sampling the whole system.

The figure 7.11 show the total energy during the minimisation process.

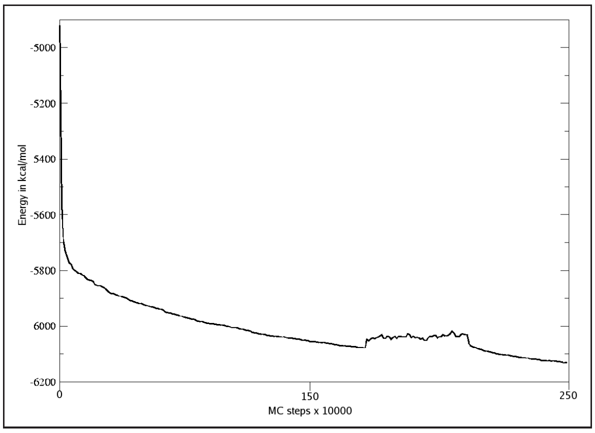


Figure 7.11: Energy in kcal/mol of the system during the minimisation process for the 1tbf protein.

Then a scoop of the protein has been created. The scoop consist of two spheres (inner and outer) of respective radius 12 and 17 Å from the sildenafil and the two ions. Residues 93, 84, 201, 203 and 204 were removed from the scoop, and the lysine 85 was deprotonated, so the total charge of the protein was brought to zero.

Then three dummy atoms were added to one hydrogen of methyl group of the sildenafil for the mutation sildenafil to vardenafil. The bonds between the dummies and the hydrogen were set to 0.2 Å. Then the system was minimised to make sure the added dummies were not a source of any steric clashes. Three short MC simulations of ten thousand steps at 0 K each were run. First, a simulation with only the protein allowed to move, second a simulation with the protein and the ligand allowed to move and the third simulation with the whole system allowed to move. The final energy of the system was -6806 kcal/mol. To finish the equilibration process, 10000 equilibration steps were performed.

The coordinates of the protein, the ions, the ligand and the water molecules were saved and used as initial structure for the simulations. Several simulations where run.

All the simulations were performed using constant temperature and number of molecules, using GBSA solvation and the CRA algorithm between the scoop residues 39 and 70 (H-loop). Both the ions and the crystallographic water molecules involved in the binding were conserved. The water shell was centre around a 14 Å spheres, and a of 0.1 kcal/mol.Å^2 was applied at the boundaries. For the 2h42, protein simulations for both the bound the unliganded (unbound) structures were performed each composed of two million MC steps. Acceptance rates for both simulations are plotted in table 7.5

System	Zn	Mg	Protein	CRA
Bound	11.3 %	8.9 %	28.8 %	27.8 %
Unbound	21.1 %	16.2 %	28.4 %	27.8 %

Table 7.5: Acceptance rates of zinc (Zn) and magnesium (Mg) ions and the protein during the two simulations (bound and unbound).

Acceptance rates for both simulations are very similar. The ions seem to achieve better sampling in the unbound protein rather than in the bound protein. Figure 7.12 shows the binding pocket at different stages of the simulation (initial in dark blue, final in red and intermediate in cyan).

Figure 7.12 shows several points. Both ions retain their coordination states (not represented in figure 7.12 are the side chains involved in the coordination). The position of the zinc remains very close to its initial position while the magnesium moves further away from the initial position. The behaviour of the ions is in good agreement with the literature[150]. The evolution

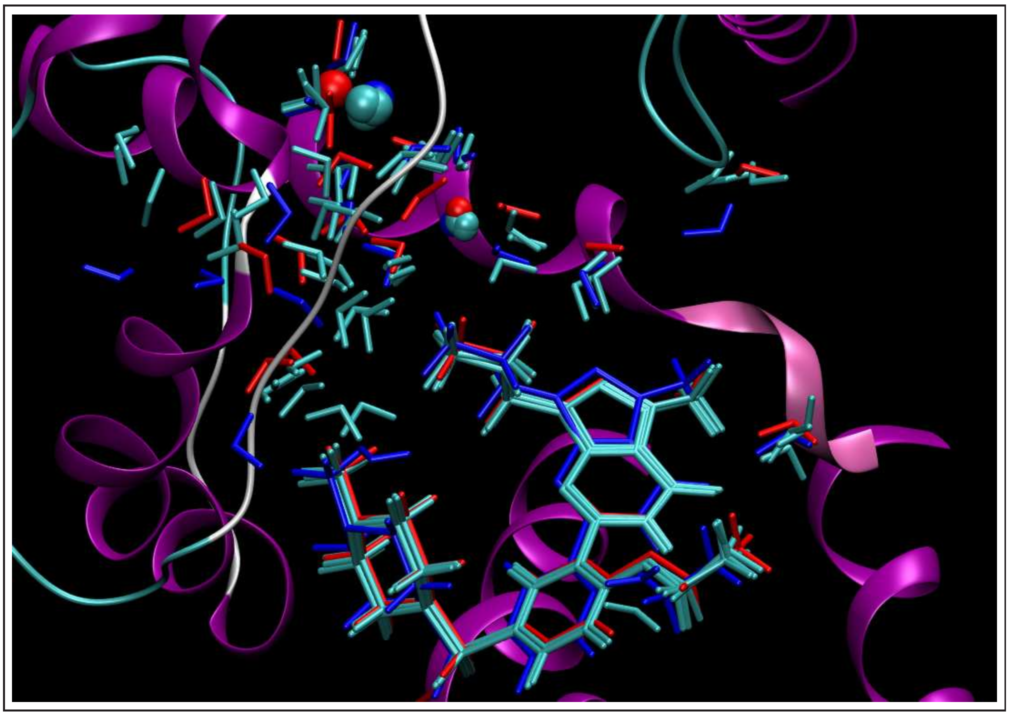
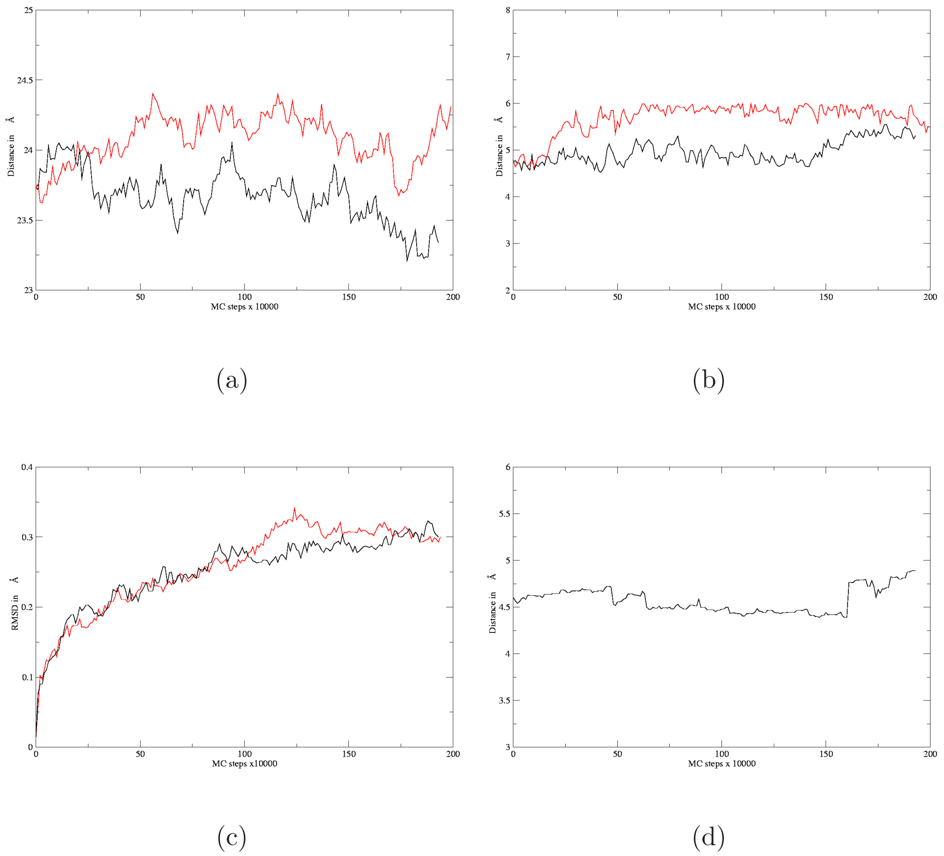


Figure 7.12: Binding pocket at different stages of the simulation for the bound protein. Sildenafil(sticks), water molecules (sticks) and zinc (small sphere) and magnesium (big sphere) ions are represented at the different stages. The initial structure of the protein is also represented.

of some intermolecular features have been plotted figure 7.13, and compared to the values from the literature[150].

The distance between the two ion and the distance between the sildenafil and the valine 157 plotted in figure 7.13(b) and 7.13(c) are close to the results by Zagrovic *et al.* for both the simulations with the bound and the holo structures. However, the RMSD is about one order of magnitude smaller. This can be explained by several facts. First, the CRA have only be used on the H-loop, the other residues have been sampled using standard ProtoMS[51] moves. Second, it is likely that the minimisation process undergone by the protein has biased the conformation of the protein toward a low energy state, where the ligand is tightly bound to the protein. And third, in the work from Zagrovic *et al.*, part of the H-loop was not present in the crystal structure and modelled using the modeller package[143]. The modelled H-loop undergoes



(a) (b)

(c) (d)

Figure 7.13: Inter-ions(b) and H670 to N789(a) distances. RMSD during the simulation(c) and distance between the C_α of the valine 157 and the C24 of the sildenafil (in blue in figure 7.8) (d). RMSD are calculated on the backbone atoms only. Values for the simulation without sildenafil are plotted in red. Values for the simulation with the sildenafil are plotted in black. all distances are in Å

large scale motion (up to 9 Å) and such motions can bias the value of the RMSD. Figure 7.13(a) shows the distance between the histidine 670 and the glutamine 789. The distance between the two residues during the bound state simulation is consistent with the value from the literature[150] (22 ± 6 Å). The distance for the holo simulation is different from the value of the literature[150] (29 ± 4 Å). However, figure 7.13(a) clearly shows that both curves are drifting away from the initial value and the MD simulations were run for a relatively long time (3 ns). The distance between the two residues for the bound protein is getting lower, whereas the distance for the unbound protein is getting bigger. This behaviour is again in accordance this the work of Zagrovic *et al.*.

So MC simulations using the CRA algorithm were able to reproduce with accuracy the results of the MD simulations. Such simulations were used to try to reproduce the experimental and theoretical value of the relative binding free energy between the sildenafil and vardenafil. Results of the RETI simulations are summarised table 7.6.

	experiment[a]	experiment[b]	MD[150]	MC/CRA
$\Delta\Delta G^{\circ}_{bind}$	-2.2 to -1.1	-1.4 to -1.3	-0.6	0.2±0.9

Table 7.6: Experimental and theoretical relative binding free energies between vardenafil and sildenafil in kcal/mol. Calculated from IC_{50} values [a] and from K_D values [b]

RETI simulations were run using the dual topology method in the GBSA continuum at constant temperature on the scoop of the 2h42 PDE5. CRA moves were performed between the scoop residues 39 to 70 (H-loop). The values of λ are similar to the values used in the lysozyme and kinase simulations. For each λ, 50 RETI moves of 30000 MC steps was run mutating the sildenafil into vardenafil. Owing the high similarity between the two ligands no equilibration at $\lambda = 0.5$ was run.

Results in table 7.6 shows that both MD and MC simulations overestimate the binding affinity of the sildenafil. However, the MC simulations using the CRA algorithm on the H-loop overestimate the binding energy of the sildenafil from around 1 kcal/mol, but this value is within the margin of the standard error. So the computed relative binding free energy using MC method is very close to the computed relative biding free energy using MD method.

7.3 Concluding remarks

Biological targets such as Bcr-Abl kinases and PDE5 phosphodiesterase have been investigated using MC simulations and the CRA algorithm. Trying to reproduce the experimental observations on the kinase conformations is still beyond the capabilities of the CRA. However, although the use of MC simulation and CRA algorithms did not managed to inter-convert the DFG loop of the kinase or fully open the activation loop after removing the ligands, the MC simulations using the CRA algorithm of comparable quantity to the MD simulations previously run *in-situ*.

In the case of the PDE5 protein, the use of of the CRA algorithm leads to the same level of sampling as MD simulations, both from a conformational and an energetic point of view. The relative binding free energy between the vardenafil and the sildenafil computed using MC with the CRA algorithm leads to the same results as the computed relative binding free energy obtained with MD method.

The CRA has proved once again to be a useful tool to sample proteins using statistical mechanics. The next chapter will conclude the work done and open new perspectives on the use of the CRA algorithm.

Chapter 8

Concluding remarks and perspectives

8.1 Concluding remarks

This research set out with the aim of implementing a novel method to sample large backbone moves for proteins, that provides enhanced sampling of protein loops and is still fast enough to be used in pharmaceutical drug design. Several algorithms have been reviewed in chapter 4 and the choice has been made to implement the CRA algorithm. To satisfy this aim, chapter 5 describes the implementation and testing of the CRA algorithm in the ProtoMS[51] package.

The CRA has been implemented as a new move in ProtoMS[51]. Two types of moves for the CRA are available, one as described in the literature[83] and one where the length of the prerotation move can be chosen to fit the requirement of the user. The user also has the possibility to choose between different parameters by changing the value of few simple key words in the input files. The CRA has provided significant enhancement of the sampling of the backbone of the protein. The speed of the CRA moves has been tested against the speed of rigid-unit backbone moves and the penalty in speed (one CRA move is about twice as slow as a rigid unit backbone move in ProtoMS[51]) is regarded to be negligible with respect to the increase of acceptance rate and sampling provided by the CRA algorithm.

The increase in the sampling enables ProtoMS[51] to achieve the same level of sampling as classical MD simulations. Once the CRA was successfully implemented and thoroughly tested, it was used to try to solve biological problems that involve large changes in the backbone conformation of the protein.

The CRA algorithm has been applied to the several biological targets to try to reproduce or better MD results. Systems have been chosen of biological interest, the lysozyme protein, the Bcr-Abl kinase and the PDE5 phosphodiesterase. For the three systems, MC simulations with CRA moves have been use to sample conformational loop problems, respectively the change in conformation of the F-loop, the switch in the DFG/activation loop and the change of configuration of the H-loop.

The use of the CRA algorithm has enhanced the sampling for the three systems. The trajectory of the simulations shows that the use of the CRA allows the loops to sample different conformations. Other technique such as parallel tempering have been used to enhance the sampling of side chains and have led to very good results in terms of sampling.

RETI simulations have been computed using the CRA moves to try to compute precise relative binding free energies. Although the use of the CRA has increased the sampling, in most cases, the computed relative binding free energies could not reproduce the experimental results.

It is not clear if this issue is only related to the sampling of the backbone, or of the sides chains, the size of the ligand/binding pocket and the accuracy of the forcefield. However the CRA algorithm could be improved in a few ways.

8.2 Future work and perspectives

The capabilities of the CRA algorithm have been barely scratched during this research and a lot more can be done. The combined use of the CRA algorithm, the definition of moves through ProtoMS[51] and the use of methods such as RETI and PT may lead to great discoveries in the field of molecular modelling.

The two parameters controlling the force of the bias and the acceptance rate in the CRA moves are those used in the literature. A first implementation could be to let the user choose and optimise which parameters to use (but the optimisation is likely to be case dependent). However changing the parameters would require extreme caution as the wrong parameters would lead to poor sampling or poor acceptance rate.

The parallel tempering method has proved to give good results in the case of the valine 111 of the lysozyme, allowing the sampling of both position of the angle of the rotamer χ. It would be interesting to sample a protein using different probabilities of move types for each temperatures. At higher temperatures, the loop involved in large scale (or slow motion) conformational change would be sampled with the highest probability (for example ten CRA move every eleven ProtoMS[51] moves at 500 K) whereas at standard temperature (298 K), the ratio CRA moves per total moves would be lowered (one CRA move every four moves). Such an approach would not brake detailed balance and would provide greater sampling of the protein loop. This approach would be very similar to the TEE-REX algorithm[161,162] from Kubitziki *et al.* where at higher temperature only the slow degrees of freedom are sampled.

It could be useful to implement more features into the CRA algorithm. If a protein undergoes a large change of conformation in more than one loop, the only way to sample all the loops so far is to allow CRA moves on all the residues located within the loop boundaries. This would lead to sampling problems if a scoop of the protein was to be used were some residues are missing or have to be kept fixed.

One possible solution would be to have the option to use the CRA move on more than one loop. Using an array to store the number of loops, and the residue number of both ends of each loops, the code would pick up one loop randomly and perform a CRA move within this particular loop boundary.

Given the good results provided by parallel tempering, it would be of great interest to be able to run both PT and RETI simulations in the same simulations (so each λ value could exchange configurations with the PT simulations). A method to couple protein change with λ to capture large scale rearrangemnets as we mutate the ligand may lead to better accuracy in the computed relative binding free energy. However such a method would be extremely costly.

With a little work to add a few extra options, the CRA could became an even more powerful tool to use to sample large backbone moves using MC simulations.

List of Figures

List of Tables

References

[1] Woldawe. D, *Annu. Rev. Med.*, **2002**, *53*, 595–614.

[2] M. J. Forster, *Micron*, **2002**, *33*, 365–384.

[3] G.E. Moore, *Electronics*, **1965**, *38*.

[4] S. J. Zhong, A. T. Macias, and A. D. Mackerell, *Curr. Top. Med. Chem.*, **2007**, *7*, 63–82.

[5] G. F. Yang and X. Q. Huang, *Curr. Pharm. Des.*, **2006**, *12*, 4601–4611.

[6] H. Alonso, A. A. Bliznyuk, and J. E. Gready, *Med. Res. Rev.*, **2006**, *26*, 531–568.

[7] D. J. Wales and H. A. Scheraga, *Science*, **1999**, *285*, 1368–1372.

[8] J. Lee, A. Liwo, and H. A. Scheraga, *Proc. Natl. Acad. Sci. U. S. A.*, **1999**, *96*, 2025–2030.

[9] J. Y. Lee and H. A. Scheraga, *Int. J. Quant. Chem.*, **1999**, *75*, 255–265.

[10] N. A. Alves and U. H. E. Hansmann, *J. Chem. Phys.*, **2002**, *117*, 2337–2343.

[11] A. R. Dinner, T. Lazaridis, and M. Karplus, *Proc. Natl. Acad. Sci. U. S. A.*, **1999**, *96*, 9068–9073.

[12] R. H. Zhou, B. J. Berne, and R. Germain, *Proc. Natl. Acad. Sci. U. S. A.*, **2001**, *98*, 14931–14936.

[13] S. C. Phillips, M. T. Swain, A. P. Wiley, J. W. Essex, and C. M. Edge, *J. Phys. Chem. B*, **2003**, *107*, 2098–2110.

[14] S. C. Phillips, J. W. Essex, and C. M. Edge, *J. Chem. Phys*, **2000**, *112*, 2586.

[15] V. S. Pande, I. Baker, J. Chapman, S. P. Elmer, S. Khaliq, S. M. Larson, Y. M. Rhee, M. R. Shirts, C. D. Snow, E. J. Sorin, and B. Zagrovic, *Biopolymers*, **2003**, *68*, 91–109.

[16] B. Zagrovic, E. J. Sorin, and V. Pande, *J. Mol. Biol.*, **2001**, *313*, 151–169.

[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, *J. Chem. Phys.*, **1953**, *21*, 1087–1092.

[18] M. Pretuz, *Protein Structure New Approaches to disease and therapy;* W.H. Freeman and Company, 1992.

[19] http://mod.life.nthu.edu.tw/protein/ptrc/teaching/protein/down.htm.

[20] L. Pauling, R. B. Corey, and H. R. Branson, *Proc. Natl. Acad. Sci. U. S. A.*, **1951**, *37*, 205–211.

[21] L. Pauling and R. B. Corey, *Proc. Natl. Acad. Sci. U. S. A.*, **1951**, *37*, 235–240.

[22] L. Pauling and R. B. Corey, *Proc. Natl. Acad. Sci. U. S. A.*, **1951**, *37*, 251–256.

[23] C. A. Pickover, *Science*, **1984**, *223*, 181–182.

[24] http://www.elmhurst.edu/ chm/vchembook/571lockkey.html.

[25] Y. Q. Deng and B. Roux, *J. Chem. Theory Comput.*, **2006**, *2*, 1255–1273.

[26] G. N. Ramachandran and V. Sassiekharan, *Adv. Prot. Chem.*, **1968**, *28*, 283–437.

[27] O. Herzberg, C. C. H. Chen, G. Kapadia, M. Mcguire, L. J. Carroll, S. J. Noh, and D. Dunawaymariano, *Proc. Natl. Acad. Sci. U. S. A.*, **1996**, *93*, 2652–2657.

[28] M. Gerstein, A. M. Lesk, and C. Chothia, *Biochemistry*, **1994**, *33*, 6739–6749.

[29] http://moose.bio.ucalgary.ca.

[30] J. W. Essex, D. L. Severance, J. Tirado-Rives, and W. L. Jorgensen, *J. Phys. Chem. B*, **1997**, *101*, 9663–9669.

[31] J. T. Kim, A. D. Hamilton, C. M. Bailey, R. A. Domoal, L. G. Wang, K. S. Anderson, and W. L. Jorgensen, *J. Am. Chem. Soc.*, **2006**, *128*, 15372–15373.

[32] R. Malham, S. Johnstone, R. J. Bingham, E. Barratt, S. E. V. Phillips, C. A. Laughton, , and S. W. Homans, *J. Am. Chem. Soc.*, **2005**, *127*, 17061–17067.

[33] E. Barratt, R. J. Bingham, D. J. Warner, C. A. Laughton, S. E. V. Phillips, and S. W. Homans, *J. Am. Chem. Soc.*, **2005**, *127*, 11827–11834.

[34] N. Shimokhina, A. Bronowska, , and S. W. Homans, *Angew. Chem. Int. Ed.*, **2006**, *45*, 6374–6376.

[35] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids;* Oxford Science Publications, 1987.

[36] D Frenkel and B Smit, *Understanding molecular simulation;* Academic Press, 2002.

[37] A. R. Leach, *Molecular modelling: principles and applications.;* Pearson Education Limited, Harlow, 1996.

[38] N. Metropolis and S. Ulam, *J. Am. Stat. Ass.*, **1949**, *44*, 335–341.

[39] W. D. Cornell, P. Cieplack, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, *J. Am. Chem. Soc.*, **1995**, *117*, 5179–5197.

[40] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner, *J. Am. Chem. Soc.*, **1984**, *106*, 765–784.

[41] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. M. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. M. Wang, and P. Kollman, *J. Comp. Chem*, **2003**, *24*, 1999–2012.

[42] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comp. Chem*, **1983**, *4*, 187–217.

[43] A. D. Mackerell, *Abstracts of Papers of the American Chemical Society*, **1998**, *216*, U696–U696.

[44] W. R. P. Scott, P. H. Hunenberger, I. G. Tironi, A. E. Mark, S. R. Billeter, J. Fennen, A. E. Torda, T. Huber, P. Kruger, and W. F. Van Gunsteren, *J. Phys. Chem A*, **1999**, *103*, 3596–3607.

[45] W. L. Jorgensen and J. Tirado-Rives, *J. Am. Chem. Soc.*, **1988**, *110*, 1657–1666.

[46] B. Alder and T. E. Wainwright, *J. Chem. Phys.*, **1957**, *27*, 1208–1209.

[47] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.*, **1982**, *76*, 637–649.

[48] D. A. Pearlman and P. A. Kollman, *J. Chem. Phys.*, **1991**, *94*, 4532–4545.

[49] J.W. Gibbs, *Elementary principles in statistical mechanics;* Dover publications Inc, New York, 1902.

[50] W. L. Jorgensen, MCPRO 1.4, Yale University, New Haven, CT, **1996**.

[51] C. Woods; *ProtoMS manual.*

[52] J. Jorgensen, W. L. Tirado-Rives, *J. Comp. Chem.*, **2005**, *26*, 1689–1700.

[53] R.W. Zwanzig, *J. Chem. Phys.*, **1954**, *22*, 1420–1426.

[54] P. A. Bash, U. C. Singh, R. Langridge, and P. A. Kollman, *Science*, **1987**, *236*, 564–568.

[55] G. L. Seibel and P. A. Kollman, *Abstracts of Papers of the American Chemical Society*, **1987**, *193*, 215–PHYS.

[56] T. P. Straatsma and J. A. Mccammon, *Ann. Rev. Phys. Chem.*, **1992**, *43*, 407–435.

[57] J. Michel, M. L. Verdonk, and J. W. Essex, *J. Med. Chem*, **2006**, *49*, 7427–7439.

[58] J. Michel, M. L. Verdon, and J. W. Essex, *J. Chem. Theory Comput.*, **2007**, *3*, 1645–1655.

[59] A. Schug, T. Herges, A. Verma, and W. Wenzel, *Journal of Physics-Condensed Matter*, **2005**, *17*, S1641–S1650.

[60] K. Hukushima and K. Nemoto, *J. Phys. Soc. Jpn.*, **1996**, *65*, 1604–1608.

[61] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.*, **1983**, *79*, 926–935.

[62] M. Ferrario and A. Tani, *Chem. Phys. Lett.*, **1985**, *121*, 182–186.

[63] W. L. Jorgensen and J. D. Madura, *Molecular Physics*, **1985**, *56*, 1381–1392.

[64] S. M. Tschampel, M. R. Kennerty, and R. J. Woods, *J. Chem. Theory Comput.*, **2007**, *3*, 1721–1733.

[65] M.Z. Born, *Phys.*, **1920**, *1*, 45.

[66] W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, *J. Am. Chem. Soc.*, **1990**, *112*, 6127–6129.

[67] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *Chem. Phys. Lett.*, **1995**, *246*, 122–129.

[68] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem.*, **1996**, *100*, 19824–19839.

[69] J. Michel, R.D. Taylor, and J.W. Essex, *J. Comput. Chem.*, **2004**, *25*, 1760–1770.

[70] RD Taylor, PJ Jewsbury, and JW. Essex, *J. Comput. Aided Mol. Des.*, **2002**, *16*, 151–166.

[71] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J.Olson, *J. Comput. Chem.*, **1998**, *19*, 1639–1662.

[72] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor, *J. Mol. Biol.*, **1997**, *267*, 727–748.

[73] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe, *J. Mol. Biol.*, **1996**, *261*, 470–489.

[74] H. Clauben, M. Rarey C. Buning, and T. Lengauer, *J. Mol. Biol.*, **2001**, *308*, 377–395.

[75] P. J. Flory, *Statistical Mechanics of chain Molecules;* Wiley, 1969.

[76] P. H. Verdier and W. H. Stockmayer, *J. Am. Chem. Soc.*, **1962**, *36*, 227–235.

[77] S. K. Kumar, M. Vacatello, and D. Y. Yoon, *J. Chem. Phys.*, **1988**, *89*, 5206–5215.

[78] B. J. Banaszak and J. J. De Pablo, *J. Chem. Phys.*, **2003**, *119*, 2456–2462.

[79] M. N Rosenbluth and A. W. Rosenbluth, *J. Chem. Phys.*, **1955**, *23*, 356–359.

[80] D. Frenkel and B. Smit, *Mol. Phys.*, **1992**, *75*, 983–988.

[81] K. Kremer and K. Binder, *Comp. Phys. Rep.*, **1988**, *7*, 259–310.

[82] L. R. Dodd, T. D. Boone, and D. N. Theodorou, *Mol. Phys.*, **1993**, *78*, 961–996.

[83] J. P. Ulmschneider and W. L. Jorgensen, *J. Chem. Phys.*, **2003**, *118*, 4261–4271.

[84] J. P. Ulmschneider and W. L. Jorgensen, *J. Phys. Chem. B*, **2004**, *108*, 16883–16892.

[85] J. P. Ulmschneider and W. L. Jorgensen, *J. Am. Chem. Soc.*, **2004**, *126*, 1849–1857.

[86] W. Boomsma and T. Hamelryck, *Bmc Bioinformatics*, **2005**, *6.*

[87] D. Frenkel and B. Smit, *J. Phys.: Condens. Matter.*, **1992**, *4*, 3053–3076.

[88] http://blender.doc.fr.free.fr.

[89] N. Go and H. A. Scheraga, *Macromolecules*, **1970**, *3*, 178–187.

[90] G. Favrin, A. Irback, and F. Sjunnesson, *J. Chem. Phys.*, **2001**, *114*, 8154–8158.

[91] R. A. Da Silva, L. Degreve, and A. Caliri, *Biophys. J.*, **2004**, *87*, 1567–1577.

[92] S. Cahill, M. Cahill, and K. Cahill, *J. Comput. Chem.*, **2003**, *24*, 1364–1370.

[93] M. Cahill, S. Cahill, and K. Cahill, *Biophys. J.*, **2002**, *82*, 2665–2670.

[94] E. W. Knapp, *J. Comput. Chem.*, **1992**, *13*, 793–798.

[95] E. W. Knapp and A. Irgensdefregger, *J. Comput. Chem.*, **1993**, *14*, 19–29.

[96] A. Lee, I. Streinu, and O. Brock, *Phys. Biol.*, **2005**, *2*, 108–115.

[97] E. A. Coutsias, C. Seok, M. P. Jacobson, and K. A. Dill, *J. Comput. Chem.*, **2004**, *25*, 510–528.

[98] S. M. Lavalle and J. J Knuffer, *Rapidly-exploring random trees: progress ans prospects. Algorithmic and Computational Robotics: New Directions;* A.K. Peters, Boston, 2001.

[99] J. Cortes, T. Simeon, M. Remaud-Simeon, and V. Tran, *J. Comp. Chem.*, **2004**, *25*, 956–967.

[100] J. Cortes, T. Simeon, V. R. De Angulo, A. D. Guieysse, M. Remaud-Simeon, and V. Tran, *Bioinformatics*, **2005**, *21*, I116–I125.

[101] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, *Ieee Transactions On Robotics and Automation*, **1996**, *12*, 566–580.

[102] S. Santos, U. W. Suter, M. Muller, and J. Nievergelt, *J. Chem. Phys.*, **2001**, *114*, 9772–9779.

[103] S. Kirillova, J. Cortes, A. Stefaniu, and T. Simeon, *Proteins: Struct., Funct., Bioinf.*, **2008**, *70*, 131–143.

[104] A. A. Canutescu and R. L. Dunbrack, *Protein Sci.*, **2003**, *12*, 963–972.

[105] M. H. G. Wu and M. W. Deem, *J. Chem. Phys.*, **1999**, *111*, 6625–6632.

[106] A. R. Dinner, *J. Comput. Chem.*, **2000**, *21*, 1132–1144.

[107] R. H. Henchman, J. A. Kilburn, D. L. Turner, and J. W. Essex, *J. Phys. Chem. B*, **2004**, *108*, 17571–17582.

[108] J. P. Ulmschneider, M. B. Ulmschneider, and A. Di Nola, *J. Phys. Chem. B*, **2006**, *110*, 16733–16742.

[109] D.A. Case, T.A. Darden, T.E. Cheatham III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, K.M. Merz, B. Wang, D.A. Pearlman, M. Crowley, S. Brozell, V. Tsui, H. Gohlke, J. Mongan, V. Hornak, G. Cui, P. Beroza, C. Schafmeister, J.W. Caldwell, W.S. Ross, and P.A. Kollman, Amber 8, university of california, san fransisco., **2004**.

[110] G. Schaftenaar and J.H Noordik, *J. Comput.-Aided Mol. Design*, **2000**, *14*, 123–134.

[111] Nobel lecture, *Physiology or Medecine;* Elsevier Publishing Company, Amesterdam, 1942-1962.

[112] http://lysozyme.co.uk/lysozyme-enzyme.php.

[113] N.C.J. Strynadka and M.N.G. James, *J. Mol. Biol.*, **1991**, *220*, 401–424.

[114] R. Diamond, *J. Mol. Biol.*, **1974**, *82*, 371–391.

[115] A. E. Eriksson, W. A. Baase, X. J. Zhang, D. W. Heinz, M. Blaber, E. P. Baldwin, and B. W. Matthews, *Science*, **1992**, *255*, 178–183.

[116] A. E. Eriksson, W. A. Baase, J. A. Wozniak, and B. W. Matthews, *Nature*, **1992**, *355*, 371–373.

[117] A. Morton, W. A. Baase, and B. W. Matthews, *Biochemistry*, **1995**, *34*, 8564–8575.

[118] A. Morton and B. W. Matthews, *Biochemistry*, **1995**, *34*, 8576–8588.

[119] D. L. Mobley, A. P. Graves, J. D. Chodera, B. K. Shoichet, and K. A. Dill, *Biophys. J.*, **2007**, pages 368A–368A.

[120] D. L. Mobley, J. D. Chodera, and K. A. Dill, *J. Chem. Theory Comput.*, **2007**, *3*, 1231–1235.

[121] D. L. Mobley, J. D. Chodera, and K. A. Dill, *J. Chem. Phys.*, **2006**, *125*, 368A–368A.

[122] D. L. Mobley, J. Chodera, and K. Dill, *Biophys. J.*, **2005**, *88*, 332A–332A.

[123] G. Mann and J. Hermans, *J. Mol. Biol.*, **2000**, *302*, 979–989.

[124] X. J. Zhang, J. A. Wozniak, and B. W. Matthews, *J.Mol. Biol.*, **1995**, *250*, 527–552.

[125] S. Boresch, F. Tettinger, M. Leitgeb, and M. Karplus, *J. Phys. Chem. B*, **2003**, *107*, 9535–9551.

[126] I. Stoica, *J. Mol. Mod.*, **2005**, *11*, 210–225.

[127] McCormick Fabbro. D, *Protein Tyrosine Kinases. From Inhibitors to Useful Drugs;* Humana Press, 2006.

[128] S. Taylor and G Ghosh, *Curr. Opin. Struct. Biol*, **2006**, *16*, 665–667.

[129] M.M.E. Noble, J.A. Endicott, and L.N Johnson, *Drug. Disc. Rev.*, **2004**, *303*, 1800–1805.

[130] B. Nagar, W. Bornman, P. Pellicena, T. Schindler, D.R. Veach, W.T. Miller, B. Clarkson, and J. Kuriyan, *Cancer*, **2002**, *62*, 4236–4243.

[131] B. Nagar, W. Bornman, P. Pellicena, T. Schindler, D.R. Veach, W.T. Miller, B. Clarkson, and J. Kuriyan, *Cancer*, **2002**, *62*, 4236–4243.

[132] M.A. Young, N.P. Shah, L.H. Chao, M. Seeliger, Z.V. Milanov, W.H. Biggs, D.K. Treiber, H.K. Patel, P. Zarrinkar, D.J. Lockhart, P. Sawyers, and J. Kuriyan, *Cancer. RES.*, **2006**, *66*, 1007–1014.

[133] B. Nagar, O. Hantschel, M.A. Young, K. Scheffzek, W. Bornman, P. Pellicena, T. Schindler, D.R. Veach, W.T. Miller, B. Clarkson, and J. Kuriyan, *Cell*, **2003**, *112*, 859–871.

[134] T Schindler, W. Bornman, P. Pellicena, W.T. Miller, B. Clarkson, and J. Kuriyan, *Science*, **2002**, *289*, 1938–1942.

[135] C.L. Sawyers, *Cancer. Cell.*, **2002**, *1*, 13–15.

[136] J. Michel, M. Orsi, and J. W. Essex, *J. Phys. Chem. B.*, **2008**, *112*, 657–660.

[137] M. Orsi, D. Y. Haubertin, W. Sanderson, and J. W. Essex, *J. Phys. Chem. B.*, **2008**, *112*, 802–815.

[138] M. Muller, K. Katsov, and M. Schick, *Phys. Rep.*, **2006**, *434*, 113–176.

[139] S. O. Nielsen, C. F. Lopez, G. Srinivas, and M. L. Klein, *J. Phys.: Condens. Matter.*, **2004**, *16*, R481–R512.

[140] J. Michel, R. D. Taylor, and J. W. Essex, *J. Chem. Theory Comput*, **2006**, *2*, 732–739.

[141] W. F. van Gunsteren, S. R. Billeter, A. A. Eising, P. H. Hunenberger, P. Kruger, A. E. Mark, W. R. P. Scott, P., and I. G. Tironi, Biomolecular simulation: The gromos96 manual and user guide, **1996**.

[142] D. Eisenberg and A. D. McLachlan, *Nature*, **1986**, *319*, 199–203.

[143] A. Fiser, R.K Do, and A. Sali, *Prot. Sci.*, **2000**, *9*, 1753–1773.

[144] R. Zoraghi, J. D. Corbin, and S.H. Francis, *J. Biol. Chem.*, **2006**, *281*, 5553–5559.

[145] J. D. Corbin and S. H. Francis, *J. Biol. Chem*, **1999**, *274*, 13729–13732.

[146] T. L. Fink, S. H. Francis, A. Beasley, K. A. Grimes, and J. D. Corbin, *J. Biol. Chem.*, **1999**, *274*, 34613–34620.

[147] I. V. Turko, S. H. Francis, and J. D. Corbin, *J.Biol. Chem.*, **1999**, *274*, 29038–29041.

[148] K. Y. J. Zhang, G. L. Card, Y. Suzuki, D. R. Artis, D. Fong, S. Gillette, D. Hsieh, J. Neiman, B. L. West, C. Zhang, M. V. Milburn, S. H. Kim, J. Schlessinger, and G. Bollag, *Mol. Cell*, **2004**, *15*, 279–286.

[149] G. L. Card, B. P. England, Y. Suzuki, D. Fong, B. Powell, B. Lee, C. Luu, M. Tabrizizad, S. Gillette, P. N. Ibrahim, D. R. Artis, G. Bollag, M. V. Milburn, S. H. Kim, J. Schlessinger, and K. Y. J. Zhang, *Structure*, **2004**, *12*, 2233–2247.

[150] B. Zagrovic and W. F. Van Gunsteren, *J. Chem. Theory. Comput.*, **2007**, *3*, 301–311.

[151] L. D. Schuler, X. Daura, and W. F. van Gunsteren, *J. Comput. Chem.*, **2001**, *22*, 1205–1218.

[152] H. J. Berendsen, J. P. Postma, W. F. van Gunsteren, and J. Hermans, *Intermolecular Forces;* Reidel, 1981.

[153] H. C. Wang, Y. D. Liu, Q. Huai, J. W. Cai, R. Zoraghi, S. H. Francis, J. D. Corbin, H. Robinson, Z. C. Xin, G. T. Lin, and H. Ke, *J. Biol. Chem.*, **2006**, *281*, 21469–21479.

[154] B. J. Sung, K. Y. Hwang, Y. H. Jeon, J. I. Lee, Y. S. Heo, J. H. Kim, J. Moon, J. M. Yoon, Y. L. Hyun, E. Kim, S. J. Eum, S. Y. Park, J. O. Lee, T. G. Lee, S. Ro, and J. M. Cho, *Nature*, **2003**, *425*, 98–102.

[155] C. S. Babu and C. Lim, *J. Phys. Chem. A.*, **2006**, *110*, 691–699.

[156] M. Elstner, Q. Cui, P. Munih, E. Kaxiras, T. Frauenheim, and M. Karplus, *J. Comp. Chem.*, **2003**, *24*, 565–581.

[157] C. S. Babu and C. Lim, *J. Chem. Phys.*, **2001**, *114*, 889–898.

[158] C. S. Babu and C. Lim, *J. Phys. Chem. B.*, **1999**, *103*, 7958–7968.

[159] R. H. Stote and M. Karplus, *Prot. Struct. Funct. and Gen.*, **1995**, *23*, 12–31.

[160] S. C. Hoops, K. W. Anderson, and K. M. Merz, *J. Am. Chem. Soc.*, **1991**, *113*, 8262–8270.

[161] M. B. Kubitzki and B. L. De Groot, *Biophy. J.*, **2007**, *92*, 4262–4270.

[162] M. B. Kubitzki and B. L. De Groot, *Structure*, **2008**, *16*, 1175–1182.