MOLECULAR DYNAMICS SIMULTATION OF ANTIFREEZE PROTEIN (T4- LYSOZYME) USING GROMACS

BY

EGBA CHUKWUEMEKA MICHAEL

PSC1205337

DEPARTMENT OF PHYSICS,

FACULTY OF PHYSICAL SCIENCES,

UNIVERSITY OF BENIN,

BENIN CITY.

July, 2016

MOLECULAR DYNAMICS SIMULTATION OF ANTIFREEZE PROTEIN (T4- LYSOZYME) USING GROMACS

BY

EGBA CHUKWUEMEKA MICHAEL

PSC1205337

A PROJECT SUBMITTED TO THE DEPARTMENT OF PHYSICS, FACULTY OF PHYSICAL SCIENCES, IN PARTIAL FUFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE (B.Sc.) OF THE UNIVERSITY OF BENIN, BENIN CITY, EDO STATE, NIGERIA.

July, 2016

CERTIFICATION

This is to certify that the work was carried out under my supervision by **EGBA CHUKWUEMEKA MICHAEL**, a final year student of Department of Physics, Faculty of Physical Sciences, University of Benin, Benin City, Edo State, Nigeria.

---------- ----------

DR, ARTHUR I. I. EJERE Date

SUPERVISOR

---------- ----------

DR. F.O EZOMO Date

(Head of Physics Dept.)

---------- ----------

(EXTERNAL SUPERVISOR) Date

DEDICATION

This work is dedicated to God Almighty, the author and finisher of my faith, the custodian of my life for his protection, guidance throughout my entire stay in the university and for giving me the wisdom and knowledge to carry out this study and making this project a reality.

I also dedicated this project to my loving parents Mr and Mrs. M. EGBA

ACKNOWLEDGEMENT

I wish to first and foremost acknowledge the Almighty God for seeing me through during my research work and giving me the strength, wisdom, and understanding to embark on this research.

My sincere gratitude goes to my supervisor DR ARTUR I. I. EJERE for his invaluable assistance in carrying out this project. Sir, I am highly grateful.

My profound gratitude goes to DR F.O EZOMO, the head of physics department, university of Benin, and all lecturers and staff of the department physics for their continuous assistance and support.

I wish to thank my friends Emmanuel, Felix, Kenneth, Paul, Jonah, Winner, Chigozie, kponke, Osas, Metonde and everyone else who have been there in one way or the other in making my stay in school an awesome experience.

Finally, my heart felt gratitude goes to my family for all the immeasurable love, care, support and prayers. Thank you

TABLE OF CONTENTS

Pages

ASBTRACT

Proteins are one of the most important families of biological macromolecules. Proteins can assume many different structures. Adopting different computational methods many protein functions and structure related problems can be explored.

This thesis focuses on three different protein issues. The structural changes induced by high temperature on a large enzyme were investigated simulating the denaturation of glucose oxidase. Molecular dynamics (MD) simulations at different high temperatures were performed. The transition state of the denaturation process was found and the relative ensemble of structures characterized. Different protein properties were analyzed and found in agreement with experimental and theoretical data. Moreover the breaking points of the protein were localized and point mutations on the protein sequence were suggested.

Antifreeze proteins (AFP) allow different organisms to survive in subzero environments. These proteins lower the freezing point of physiological fluids.

MD simulations of the snow flea AFP (sfAFP) in water have shown the partial instability of the protein structure. When attached to different ice planes at the ice/water interface, the sfAFP induces local ice melting. AFPs are divided into two categories: hyperactive and moderately active depending on their antifreeze

power. The water diffusion profile of ice/water systems containing one protein from each family were compared.

CHATER ONE

1.0 Introduction

1.1 Temperature and Antifreeze Proteins

Temperature is an important factor for survival of living organisms and when water the biological solvent freezes to ice, living species are met with a major challenge. Poikilothermic species exposed to subzero temperatures, have two main strategies to protect themselves from cold damage; they can either avoid cold by preventing intra–and extracellular ice formation in their bodies (freezing intolerance) or they can survive with the formation of extracellular ice in their bodies(freezing tolerance). The discovery of antifreeze proteins started out as a curiosity among scientist, about how the survival of fish in cold climates was possible. In the 1950's, it was discovered that the survival of Antarctic teleost fish at subzero temperatures was caused by the presence of specialized antifreeze substances in their blood, rather than the presence of salts or additional substances. Studies revealed that these antifreeze substances were proteins, which could lower the freezing temperature in Arctic fish, thus protecting the fish from cold damage. Following studies revealed similarly acting proteins – hence giving reason to the overall name, antifreeze proteins.

Antifreeze proteins (AFPs) is a particular class of proteins that suppress ice growth in organisms and thereby enable their survival in cold and subfreezing

conditions. These proteins provide protection for these organisms by depressing the freezing point of local water bodies, reacting in a non-colligative manner. The addition of AFPs at the interface between solid ice and liquid water inhibits the thermodynamically favored growth of the ice crystal. Ice growth is kinetically inhibited by AFPs covering the water-accessible surfaces of ice. This difference between the freezing and melting temperature is referred to as thermal hysteresis. It is used as the characteristic measure of the antifreeze activity of an AFP. Both glycoproteins and AFPs which are extracted from the blood of polar fish, usually exhibit up to 2 degrees Celsius of thermal hysteresis. These types are called moderate AFPs whereas certain insects' exhibit up to 5 degrees Celsius of thermal hysteresis, and these are classified as hyperactive antifreeze proteins. Thermal hysteresis can be measured experimentally using a custom made Nano liter osmometer. Early experimental models suggested that hydrogen bonding of AFP threonine residues to ice was the major driving force of AFP-ice association (De Vries 1974, Raymond and De Vries 1977) with AFP binding to ice via the threonine face. However, mutation experiments revealed that hydrogen bonding was not the driving mechanism and that Van der Waals and hydrophobic interactions were suggested to play important roles in AFP binding. The actual mechanism of AFP binding is analyzed using atomistic the molecular dynamics simulation, and then compared with the experimental results (Havenith and

Leitner2003). Havenith experimentally studied the dynamics using terahertz spectroscopy and predicted that antifreeze activity was directly correlated with long range collective hydration dynamics and provided evidence of a new model for the proteins to survive under extreme conditions. The results suggest that the protein perturbs the aqueous solvent over long distances and retarded hydration dynamics in the large hydration shell does not favor freezing. There are two reasons why many types of AFPs are able to perform the same function despite their diversity. Although ice is uniformly composed of oxygen and hydrogen, it has many different surfaces exposed for binding. Different types of AFPs may interact with different surfaces. Although the four types of AFPs differ in their primary sequence of amino acids, when each folds into a functioning protein, they may share similarities in their three dimensional or tertiary structure that facilitates the same interactions with ice. The preliminary proposed mechanism in which freezing point depression is achieved by an adsorption-inhibition mechanism in which the proteins recognize and bind quasireversibly to an ice surface, thereby preventing growth of ice crystals. In this thesis we used an AFP-water model instead of creating an ice water interface, since it is difficult to simulate ice using the molecular dynamics simulation. The PRODRG server is used to create the topology and coordinate files. The antifreeze proteins (AFPs) are also called ice structuring proteins (ISPs) are a class of proteins produced in many vertebrates,

plants, fungi and bacteria that permit their survival in subzero environments. AFPs bind to small ice crystals to inhibit growth and recrystallization of ice that would otherwise be fatal for the organism. In comparison to widely used antifreeze substances like ethylene glycol, AFPs do not have lower freezing point in proportion to concentration. So this allows them to act as antifreeze at very less concentrations (about 1/300) of those of other dissolved solutes, hence minimizes their effect on osmotic pressure. The binding capabilities of AFPs has been studied and attributed to their binding ability at specific ice crystal surfaces. Many types of antifreeze glycoprotein or AFGPs are also widely found in fishes of Antarctic regions having molecular mass range around 2.6-3.3 KD. There are basically four classes of antifreeze proteins Type I, II, III and type IV. The Type II AFPs are mostly studied and these are cysteine-rich globular proteins containing five disulfide bonds. The inhibition of freezing in case of the AFPs are thought to by an adsorption–inhibition mechanism as they adsorb to non-basal planes of ice, inhibiting thermodynamically favored ice growth. There are many applications of antifreeze proteins, these proteins are used in increasing freeze tolerance of crop plants and extending the harvest season in cooler climate improving farm fish production in cooler climates, lengthening shelf life of frozen foods ,enhancing preservation of tissues for transplant or transfusion in medicine, therapy for hypothermia as cryoprotective agents for organ and cell cryopreservation and also

as chemical adjuvants to cancer cryosurgery and in development of transgenic plants and animals with increased tolerance to cold environment. As the function of these proteins resides on stability of the 3D structure hence, molecular dynamics simulation based analysis is essential to study the structural stability in different physiological environment. PH dependent molecular dynamics simulation methods are used to predict the substrate specificity and selection of enzymes thereby provides a deep insight to study about structural and functional aspect. The ice binding capability in case of antifreeze protein has been studied in different environmental conditions like gas phase, solvated by water, adsorbed on the ice crystal plane in the gas phase and in aqueous solution by molecular dynamics simulation method. Also the temperature dependent unfolding pathway has been resolved for type III antifreeze protein by GROMACS (Groningen Machine for Advanced Chemical Simulation) software package. The prime objective of the study is to evaluate the effect of physiological pH on antifreeze protein. The antifreeze protein type II is considered here. The molecular dynamics simulation was performed in water in series of different pH and the structural basis was evaluated by calculating energy, root mean square deviation, radius of gyration, residue wise root mean square fluctuation, distribution of hydrogen bonds etc. The study is performed to establish the relationship between the physiological pH with the structural and functional aspect of the antifreeze protein.

CHAPTER 2

2.0 PROTEINS

Proteins are one of the most important families of biological macromolecules. Proteins can assume many different structures. This make them perfect to serve a wide range of functions in all organisms. In the last decades, molecular modelling has become an important and powerful tool in the investigation of biological systems. Adopting different computational methods many protein functions and structure related problems can be explored. This thesis focuses on three different protein issues. The structural changes induced by high temperature on a large enzyme were investigated simulating the denaturation of glucose oxidase. Molecular dynamics (MD) simulations at different high temperatures were performed. The transition state of the denaturation process was found and the relative ensemble of structures characterized. Different protein properties were analyzed and found in agreement with experimental and theoretical data. Moreover the breaking points of the protein were localized and point mutations on the protein sequence were suggested. Antifreeze proteins (AFP) allow different organisms to survive in subzero environments. These proteins lower the freezing point of physiological fluids.

MD simulations of the snow flea AFP (sfAFP) in water have shown the partial instability of the protein structure. When attached to different ice planes at the

ice/water interface, the sfAFP induces local ice melting. AFPs are divided into two categories: hyperactive and moderately active depending on their antifreeze power. The water diffusion profile of ice/water systems containing one protein from each family were compared. The ice/water interface width was found to be broadened to different extent by the two proteins, while a control protein (lysozyme) did not affect the interface thickness.

Hemoglobin is the oxygen carrier in all vertebrates. Mutation along the protein sequence can alter the protein functionality and its capability of binding molecular oxygen. Density Functional Theory methods were applied in the calculation of the oxygen binding energy of the wild type hemoglobin and four other variants. Evaluations on the electronic structures and on the binding energies of the different hemoglobin variants suggest that perhaps none of the mutated hemoglobin's efficiently bind oxygen.

Proteins are one of the most important families of macromolecules. They are found in all living organisms. They are essential for life and moreover for life variety. Proteins are decoded from the DNA and can have very different three-dimensional structures. The enormous number of structures assumed by proteins makes them perfect to serve different functions. Proteins structures and functions have been of great interest for scientists for a long time. Nevertheless the first protein structure was solved only in the middle of the last century. In less than ten

years the first protein was sequenced [1] by Frederick Sanger and the first 3D proteins structures [2] were solved by Max Perut and John C. Kendrew. Their discoveries led to entire new fields of research.

In the 70s computers became powerful enough to investigate proteins using mathematical methods that were created some years before. These methods, including Quantum Mechanics (QM), Molecular Dynamics (MD) and Monte Carlo (MC) are nowadays widely used in the study of proteins and many other kinds of molecules. Computational chemistry has become a useful tool in understanding protein interactions and functions. Coupled with experiments, computers simulations can be used to either understand or predict the behavior of proteins. Moreover computer simulations allow us to look at the problem at atomic resolution. In the last decades, adopting different methods, simulations have been used to understand the functions of many different proteins.

Proteins can interact with other macromolecules like DNA or RNA or other proteins as well. Simulations can be used to understand the nature of the interaction as well as the modification induced on the protein structure by the interaction. When proteins work as enzymes they usually interact with small molecules. Here computational methods can provide detailed description of the reaction from both energetic and structural points of view. Proteins assume their characteristic shape by a process called folding. The protein folding issue has been

challenging for scientists for many years. Computers have provided a better understanding of the pathways followed by proteins to assume and loose their 3D conformations [3]. Many other aspects of the protein world are and can be investigated, applying computational methods.

Computational methods have been applied to different protein issues. The most used methods are Molecular Dynamics and Density Functional Theory, which are described in the following sections. First, high temperature MD simulations were performed on a large enzyme (Glucose oxidase) to investigate its denaturation process. The goal was to find and describe the transition state of the denaturation process of the enzyme. A Root Mean Square Deviation (RMSD) based clustering method was successfully applied in the research of the transition state. The studied protein is the largest protein so far where this method has been applied. Antifreeze proteins (AFP) are the next subject of study. These proteins, found in organisms that live in subzero environments, have been shown to prevent freezing of the physiological fluids. MD simulations of a newly discovered AFP were performed in both water and at different ice/water interfaces. The work aims to describe the dynamics of the protein in water at room temperature and its interactions with the ice surface. Different subzero temperatures and ice planes as well as different protein orientations were investigated. In a second investigation on this subject, the influence of two AFPs on the ice/water interface were studied.

Diffusion profiles data were used to measure the variation of the ice/water interface thickness when AFPs are bound to ice. A non-AFP protein was also simulated as control. In the last work the binding energy of five hemoglobin variants were investigated using DFT methods.

Mutations can alter the oxygen binding site of hemoglobin, leading in some cases to serious diseases. Five hemoglobin variants were compared in terms of binding energy and geometry of their complexes with O2.

2.1 MOLECULES OF LIFE

Many different molecules are crucial for life. Water is absolutely one of them.

It is thought that life developed first in the oceans. Our body is mainly made of water as well as the body of all living organisms. Moreover, most processes and chemical reactions take place in water solution. Among the one hundred elements of the Periodic Table, very few are basic constituents of life. These are hydrogen, carbon, nitrogen, oxygen, sulfur and phosphorus. These elements are the building blocks of the three most important macromolecules: nucleic acids (DNA and RNA), proteins and carbohydrates, which are all essential for all living organisms. The most important carbohydrate is cellulose, a polymer of glucose, which is the main constituent of all plants. The deoxyribonucleic

acid (DNA) is the molecule where the genetic information is stored. Also proteins are polymers, namely of smaller units called amino acids. Proteins are found in all living things and all viruses. Proteins are responsible for virus's structure, while in the other organisms they exert a wide range of functions. DNA existence was known since the 19th century, even if its structure was still unknown. In the middle of the 20th century, physicist Erwin Schrödinger in his book What Is Life? The Physical Aspect of the Living Cell [4], introduced the idea of an "aperiodic crystal" that contained all genetic information for all chemical bonds. Schrödinger postulated the existence of a molecule that could store the basic information necessary for life. This idea inspired two other scientists, J. D. Watson and F. Crick, that ten years later discovered the DNA double helix structure. DNA is not only essential for life and its replication, but it is also the source of life diversity. The physical characteristics of living things are stored in its code and expressed into proteins.

2.2 PROTEIN STRUCTURE

The structure of a protein can be divided into four different levels called primary, secondary, tertiary and quaternary structure. The primary structure of a protein is the simple sequence of amino acids that form the protein. The primary

structure of a protein drives the so called folding of proteins. By folding is intended how the long polypeptide chain is shaped in the three-dimensional space. The main chain, or backbone, of a protein is formed by four atoms from each amino acid. These four atoms form two dihedral angles called φ and ψ which are in principle free to rotate. The dihedral angles give to the protein backbone the flexibility necessary for the protein to fold. The rotational freedom of the φ and ψ angles should give to each protein the possibility to fold in an enormous number of conformations. However for each protein in its biological environment, only one of those conformations is adopted. The reason is that all the amino acids side chains interact with each other establishing weak non-covalent bonds (like hydrogen bonds and van der Waals interactions) which drive the protein folding. Another factor that governs protein folding is the distribution of the two main class of amino acids, polar and hydrophobic. During the folding process the majority of hydrophobic side chains tend to be buried inside the protein forming the protein core. In this way they can avoid contacts with the surrounding water molecules. The polar residues instead tend to be on the exterior of the protein, mainly interacting with water. When the protein folds there are some particular shapes that are adopted by the backbone. These patterns represent the second level of the protein structure, or as it is most commonly called the secondary structure. The two most common patterns found in proteins structures are the α-helix [5] and the β –

strand. The α-helix is generated when the backbone of the protein turns around itself in a regular trend (see Fig. 2.1). In this way the backbone atoms form a hydrogen-bonding scheme where the oxygen of one amino acid is hydrogen bonded to the nitrogen of the other amino acid four residues away in the protein sequence. Some geometrical parameters are typical for a α-helix: there are 3.6 residues per helix turn, the distance between two helical turns (Pitch) is equal to 5.4 Å, every amino acid takes 1.5 Å along the helix the β -strand is instead an extended segment of a polypeptide chain which is the unit of the β -sheet. Depending on how two or more β -strands interact with each other it is possible to form two different kinds of β -sheet (see Fig. 2.2). If the two strands run in the same direction they form a parallel β -sheet, while when they run in opposite directions an anti-parallel β -sheet is formed. Another fundamental difference between parallel and anti-parallel β -sheets is the hydrogen bonding pattern. In the anti-parallel arrangement every N-H group of one strand is hydrogen bonded to the C=O group of another strand and the hydrogen bonds are planar. In the parallel conformation the hydrogen bonds are instead non-planar, which make the parallel conformation less stable than the anti-parallel one. It is also possible to find in some proteins extended β -sheet with mixed conformations, where some strands connect in parallel and some other strands connect in anti-parallel way. The protein backbone can also fold following other patterns. For example other helical forms

are possible, like the 310 -helix or the π-helix. The former has 3 residues per turn while the latter has 5 residues per turn. Then there are the turns that connect one secondary structure element to another. In order to assign the secondary structure to all amino acids of a protein is commonly used the Dictionary of Proteins Secondary Structure (DSSP) [6].

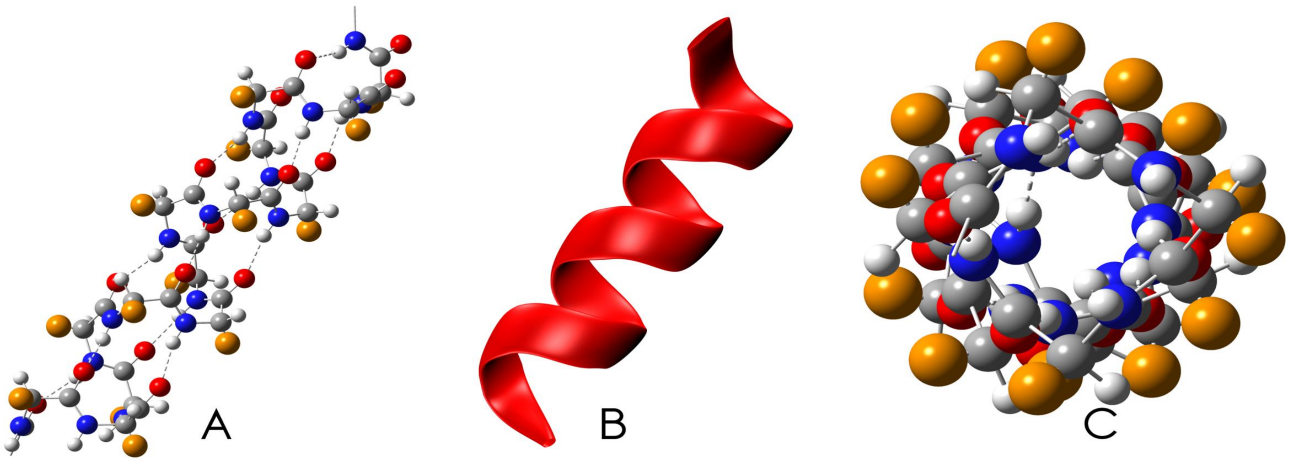


Figure 2.1: Secondary structure: α-helix. A): hydrogen bonding pattern. B) Cartoon representation. C) Top view.

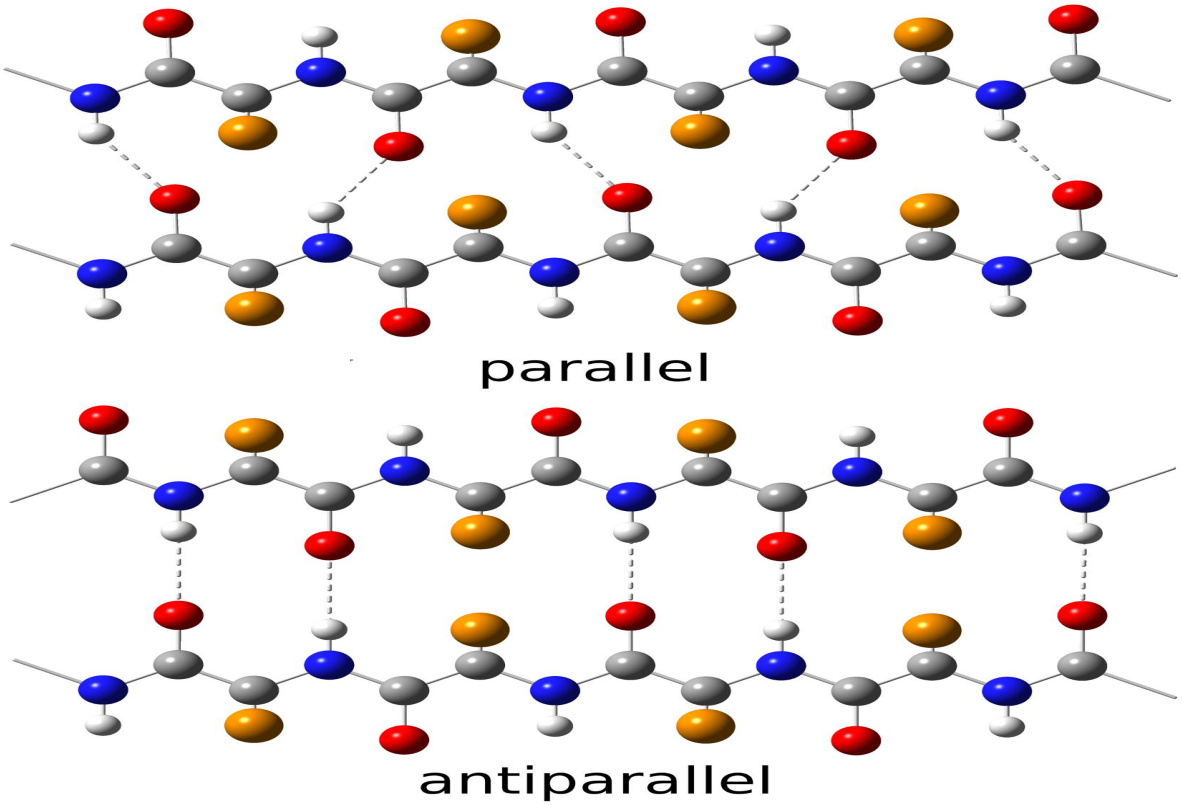


Figure 2.2: Secondary structure: β -sheet. Parallel (top) and anti-parallel arrangement.

DSSP assigns the secondary structure according to the hydrogen-bond patterns and geometrical features of a protein structure. It defines eight different secondary structure elements: α-helix, 3_{10} -helix, π-helix, β -sheet, β -bridge, turn, bend and coil. B-bridges are formed by only two β -strands, bends and turns are connectors of other secondary structure elements, coils or random coils refer to residues in none of the above conformations. Usually the protein structure is also represented as distribution of the dihedral angles φ and ψ in the Ramachandran plot [7]. This map shows the amino acids population in different conformational areas

that correspond to the secondary structure elements. A single polypeptide chain, with more than one secondary structure element, represent the tertiary structure of a protein (see Fig. 2.3). The tertiary structure of a protein is described by its atomic coordinates in the three-dimensional space. The process of finding these atomic coordinates is called solution of the structure of the protein. Mainly two experimental techniques are used to solve the protein structure: X-ray crystallography and Nuclear Magnetic Resonance (NMR). The atomic coordinates of all known proteins structures are deposited into a data bank: the Protein Data Bank (PDB).

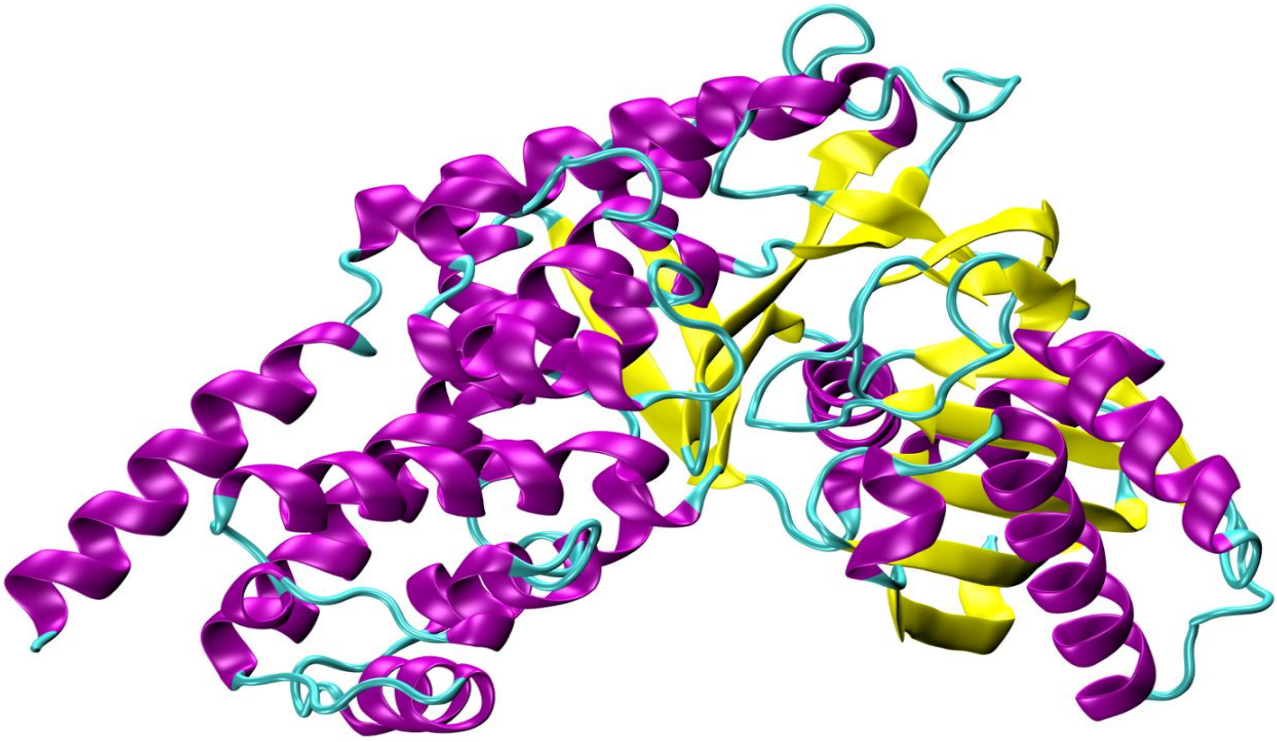


Figure 2.3: The tertiary structure. Cartoon representation of the human glucokinase.

The highest level of protein organization is the quaternary structure. A protein has a quaternary structure when two or more chains or subunits interact with each other to form a single biological entity. The single subunits that form the quaternary structure of a protein can be identical or different. The most common example of a protein that has a quaternary structure is hemoglobin. The hemoglobin structure together with myoglobin (a single chain protein related to hemoglobin) were the first proteins to be solved in the late 50's. The hemoglobin is composed of four subunits (see Fig. 2.4), two called α and two called β. The two α subunits are identical as well as the two β, while the α and β subunits are different. The different subunits are held together by non-covalent interactions usually referred as salt bridges. The salt bridges are the combination of two types of interactions: hydrogen bonds and electrostatic interactions. Hemoglobin, like many other proteins, has attached by non-covalent interactions a non-protein molecule, the heme group.

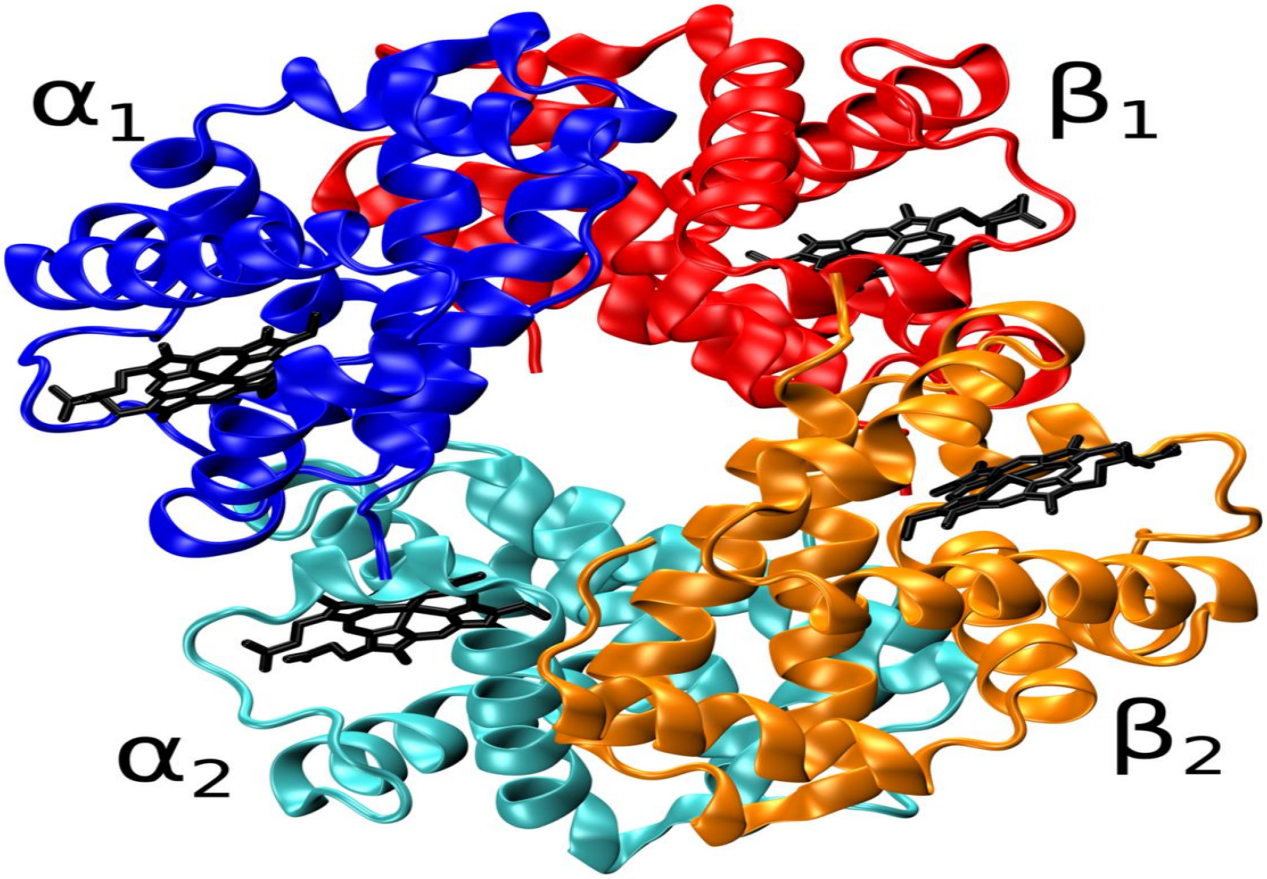


Figure 2.4: The quaternary structure. Cartoon representation of the human Hemoglobin.

2.3 PROTEIN FUNCTIONS

As we have seen proteins have the possibility to fold into a great variety of structures. This gives them the possibility to serve for a very large number of different functions. Within the cell, proteins perform many different functions ranging from structural to transport of signals or small molecules. Protein functions can be split into seven main different categories: antibodies, movement, enzymes,

hormones, structural, storage and transport. Antibodies or immunoglobulin are generally glycoproteins that can interact or bind to bacteria and viruses in order to neutralize them and defend the organism. They are produced by the plasma cells and are the main component of the immune system. Proteins like actin or myosin are responsible for body movement. They form the muscle microfilaments and are responsible for muscles contraction allowing organisms to move. Hormones are proteins involved in the regulation of organs activity and metabolism. Insulin for example is the protein which regulates glucose metabolism. Proteins like collagen and elastin are fibrous proteins which provide support to connective tissues. Tubulins are the basic component of microtubulus which provide structure to the cell. Moreover viruses shape is mainly determined by proteins which are used as shell to protect their genetic material. Ferritin is an example of storage proteins. It is responsible for storing iron and release it where is required. Another example of storage proteins is casein which is found in the milk and provides amino acids, calcium and phosphorus. Hemoglobin is probably the most important transport protein. It carries molecular oxygen from the lungs to the rest of the body. Acting as enzymes is one of the most important function of proteins. They are responsible for a countless number of processes and chemical reactions that take place in all living organisms. Enzymes are biological catalysts that accelerate reaction rates and lower the energetic barriers of metabolic reactions. Protein structures create

specific places on the protein surface where an external molecule, called substrate can bind and react. This specific moiety of the protein is usually called binding site. It can be either a "pocket" created on the protein surface by a specific arrangement of amino acids or by a coenzyme. A coenzyme is a specific ligand bound non-covalently to the protein which perform the catalytic reaction. An example of coenzyme is the heme group (see Fig 2.4) that allows hemoglobin to bind and release oxygen.

CHAPTER THREE

3.0 MOLECULAR DYNAMICS

In the 1920s and 30s a great revolution took place in physics. Great scientists, among which Werner Heisenberg, Wolfgang Pauli, Niels Bohr, Max Plank, Erwin Schrödinger gave birth to a new interpretation of the atomic world: quantum mechanics. With the advent of the new theory, Newton's mechanics was thought to be off the table in understanding the motion of small particles like electrons and atoms. Since science is in constant evolution, it was just a matter of time before Newton's mechanics was reintroduced in the discussion. Some 20 or 30 years after the "quantum revolution", Newton's mechanics become the central part of computational techniques that are widely used nowadays: Monte Carlo and Molecular Dynamics simulations. The technique named Monte Carlo is due to Nicholas Metropolis along with Stanislaw Ulam and John von Neumann during their work at Los Alamos. Metropolis and his coauthors published the first article based on the MC method in 1953 [8]. The first works based on the MD technique appeared some years later. Alder and Wainwirght published in 1959 [9], while Aneesur Rahman in 1964 published a study on liquid argon [10]. All these pioneering studies were done at a very early stage of computers development. When in the 70s computer power increased enough to study more complex systems like DNA and proteins, Michael Levitt and Arieh Warshel started simulation

studies of these macromolecules [11]. Today scientists have the possibility to use much more powerful machines and even clusters of machines, where thousands of single computers can perform large computations running in parallel. One of the most adopted technique used in the study of macromolecules is Molecular Dynamics, and it is also the most applied method in this thesis.

Molecular dynamic simulation is a computational method that simulates the motion of a system of particles. McCammon introduced the first protein simulations in 1977, and since then this method has been widely used in the theoretical study of biological molecules including proteins and nucleic acids because it can provide molecular change information by calculating the time dependent behavior of a molecular system. For example, GROMACS is a package that carries out molecular dynamic simulations, and generates a trajectory of the molecule. GROMACS's high performance draws a lot of interest from researchers looking to develop their own tools to analyze the GROMACS trajectories. JGromacs, one of many applications written in the different languages from that used by GROMACS, analyzes the trajectories generated by GROMACS. JGromacs does not work with large molecules due to its huge memory consumption. In this project, we attempt to simplify the GROMACS steps, and develop our own analysis tool that works well with large molecules. The goal of a molecular dynamics simulation is to predict macroscopic properties such as

pressure, energy, heat capacities, etc. from the microscopic properties including atomic positions and velocities generated by molecular dynamic simulations. The bridge between macroscopic properties and microscopic properties is statistical mechanics using the time independent statistical average. A molecular dynamics simulation generates a sequence of points in a multidimensional space as a function of time, where the point belong to the same collection of all possible systems which have different mechanical states such as positions or coordinates, and have the same thermodynamic state such as temperature, volume, pressure.

In statistical mechanics, an ensemble averages corresponding to experimental observables, and this means the molecular dynamics simulations must calculate all possible states of the system to get the ensemble averages. "The Ergodic hypothesis, which states that the time average equals the ensemble average," allows the molecular dynamics simulation with enough representative conformations to calculate information on macroscopic properties using a feasible amount of computer resources.

Molecular dynamics, as mentioned above, is based on the integration of Newton's law of motion, in particular on the integration of Newton's second law. If we consider a particle of mass m and a force F_{xi} acting on it along the coordinate x, we can get the particle position at the time t, solving the following equation:

$$\frac{d^2x_i}{dt^2} = \frac{F_{xi}}{m_i} \tag{1.0}$$

Since the modeled systems contain several thousands of particles (in some cases even millions), the force that act on a single particle changes every time that it interacts with the other particles. Unfortunately, this kind of problems cannot be solved analytically for more than two particles, as shown by Henri Poincaré. Hence the equations have to be solved numerically applying some assumptions. In order to calculate the evolution of a system of N interacting particles, we have to sum over finite time intervals the forces acting on a given particle. In this way the force acting on the i-th particle with coordinates r_i at time t is obtained as vectorial sum of all forces due to the other particles:

$$F_i(\{r_i(t)\}) = \sum_{j \neq i}^{N} F_{ij} \tag{1.1}$$

Knowing the force acting on the i-th particle, it is possible to calculate the velocity and then the acceleration of the particle at time t. In this way the particle position at time t + Δt can be obtained assuming that the force is constant during the interval

Δt. So, the forces acting on the i-th particle in the new configuration can be computed obtaining new position and velocity for a new configuration. Repeating this process a large number of times allows us to get the evolution of the simulated system, or as it is commonly called, the trajectory.

All integrators used in MD are based on Taylor series expansion. Given a particle with mass m and coordinates r, at time t, the new coordinates after a discrete Δt (time step) can be obtained from:

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \frac{1}{6}b(t)\Delta t^3 + \ldots \qquad (1.2)$$

Dropping all terms higher than Δt^3 and applying Newton's second law a(t)=F(t)/m, Loup Verlet suggested one of the first integrators which is based on the expansion of a particle's position on the previous and following time step:

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{F(t)}{2m}\Delta t^2 \qquad (1.3)$$

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{F(t)}{2m}\Delta t^2 \qquad (1.4)$$

Summing up equations 1.3 and 1.4, the new coordinates can be obtained from position and acceleration at time t and the position of the previous time step:

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + \frac{F(t)}{2m}\Delta t^2 \qquad (1.5)$$

Since all third and higher order terms are dropped, numerical errors might be introduced into the update scheme. Velocities are not calculated, so if they are needed they have to be computed separately. To overcome these problems the so called leap-frog algorithm is usually preferred. The algorithm uses positions r, at time t and velocities at half-time step:

$$v(t+\frac{1}{2}\Delta t) = v(t-\frac{1}{2}\Delta t) + \frac{F(t)}{m}\Delta t \tag{1.6}$$

$$r(t+\Delta t) = r(t) + v\Delta t(t+\frac{1}{2}\Delta t) \tag{1.7}$$

3.0.1 Force Field

The term force field refers to a set of parameters and a mathematical function which describe the potential energy of the studied system of particles. The potential energy is related to the force acting on a particle by the equation:

$$F_i = -\frac{\delta U_i}{\delta r_i} \tag{1.8}$$

The potential energy U can be obtained as a sum of different contributions which

include bonded and non-bonded interactions:

$$U = U_{bonded} + U_{non-bonded} \quad (1.9)$$

The bonded interactions are divided into three different contributions, due to bond stretching, angle bending and torsional angle along the bond:

$$U_{bonded} = U_{bond} + U_{angle} + U_{torsion} \quad (1.10)$$

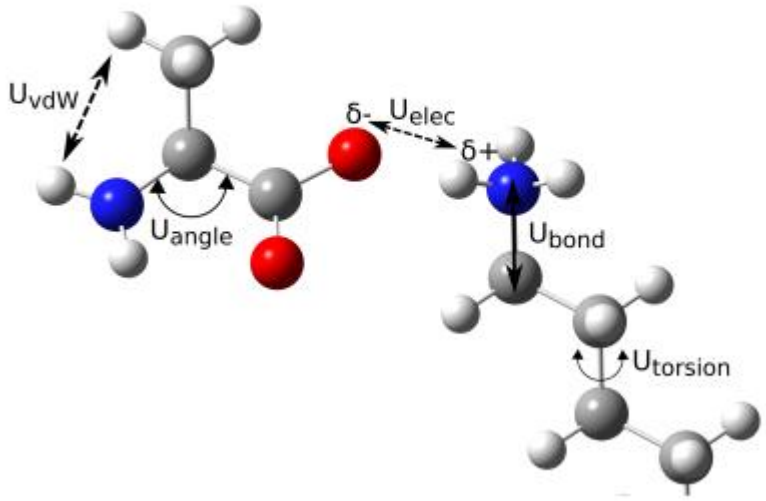


Figure 3.1 the five different interactions contributing to the total potential energy

The non-bonded interactions are instead divided into two contributions, due to pairwise electrostatic (Coulomb) and van der Waals interactions:

$$U_{non-bonded} = U_{elec} + U_{vdW} \quad (1.11)$$

The five different terms that contribute to the total potential energy are shown in Fig. 3.1. All five contributions can be described by different potential forms.

The most common way of describing the bonded interactions is through a harmonic potential, as shown in equations 1.12, 1.13 and 1.14 for bond, angle and torsion respectively:

$$U_{bond} = \sum_{bond} k(r - r_0)^2 \tag{1.12}$$

$$U_{angle} = \sum_{angle} k_\theta(\theta - \theta_0)^2 \tag{1.13}$$

$$U_{torsion} = \sum_{torsion} k_\phi(1 + cos(n\phi - \phi_p)) \tag{1.14}$$

Other potential forms can be used to describe the bonded interactions. In the simulation software package GROMACS [12] (used in this thesis to perform all MD simulations) for example are also available the Morse potential [13] for bond stretching, the Urey-Bradley potential [14] for angle bending, Ryckaert-Bellemans potential and Fourier function [15] for dihedral angles. The non-bonded interactions are usually described by the Coulomb potential for the electrostatic interactions and a 6-12 Lennard-Jones the repulsion and dispersion terms as shown in equations 1.15 and 1.16 respectively:

$$U_{elec} = \sum_{elec} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \tag{1.15}$$

$$U_{I.J} = \sum_{I.J} 4\varepsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right) \tag{1.16}$$

3.0.2 Periodic Boundary Conditions and Long-Range Interactions

In computer simulations the studied systems have to have limited dimensions, defined in a simulation box. The simulation boxes can have different shapes, from the simplest cubic to more sophisticated like rectangular, rhombic dodecahedron or truncated octahedron. In all the above cases, particles close to the walls of the simulation box would experience different forces from those particles in the middle of the box. To overcome this issue Periodic Boundary Conditions (PBC) are applied in three dimensions. An illustration of PBC in two dimensions is given in Fig. 3.2. In the two-dimensional example, when a molecule approaches the wall of the box and crosses it going out of the box, another molecule in one of the box replicas enters from the opposite wall. In this way the number of molecules is kept constant in all box replicas. One problem can rise from the replicas of the central box. In systems containing a single macromolecule like a protein or a DNA fragment, the box dimensions have to be long enough to avoid interactions of the

solute with its replicas in the neighbor boxes.

Both electrostatic and van der Waals interactions are long-range interactions. In systems with several thousands of particles the amount of long-range interactions represent the main part of the computation during a MD simulation. Hence the computation of long-range interactions is limited in order to perform long MD simulations in reasonable amount of time. The simplest way to limit the amount of interaction to be computed is applying a cut-off distance. In this way all the particles farther than a certain distance are not considered interacting with each other. Since this can create artifacts in the neighborhood of the cut-off region, the potential energy can be switched or shifted towards zero by a function which gradually reduces the interactions to zero. This method works well for van der Waals interactions which decay as r^{-6}, so they decay very fast. Electrostatics interactions have a major long-range nature (decay as r^{-1}), so a simple cut-off would introduce severe artifacts in the interaction energy. A common way to compute the electrostatic interactions is the Ewald summation method. This method was first introduced to calculate long-range interactions of the periodic images in crystals.

The electrostatic potential of N particles interacting with their periodic images is given by:

(1.17)

$$U_{elec} = \frac{1}{2}\sum_{n_x}\sum_{n_y}\sum_{n_z*}\sum_{i}^{N}\sum_{j}^{N}\frac{q_i q_j}{4\pi\varepsilon_0 r_{i.j,n}}$$

where n x , n y and n z are the box vectors and the star (*) indicates that terms with i = j should be omitted when (n x ,n y ,n z)=(0,0,0). This method is anyway quite slow, so in order to improve its performance the Particle Mesh Ewald (PME) method was introduced by Tom Darden and coworkers.

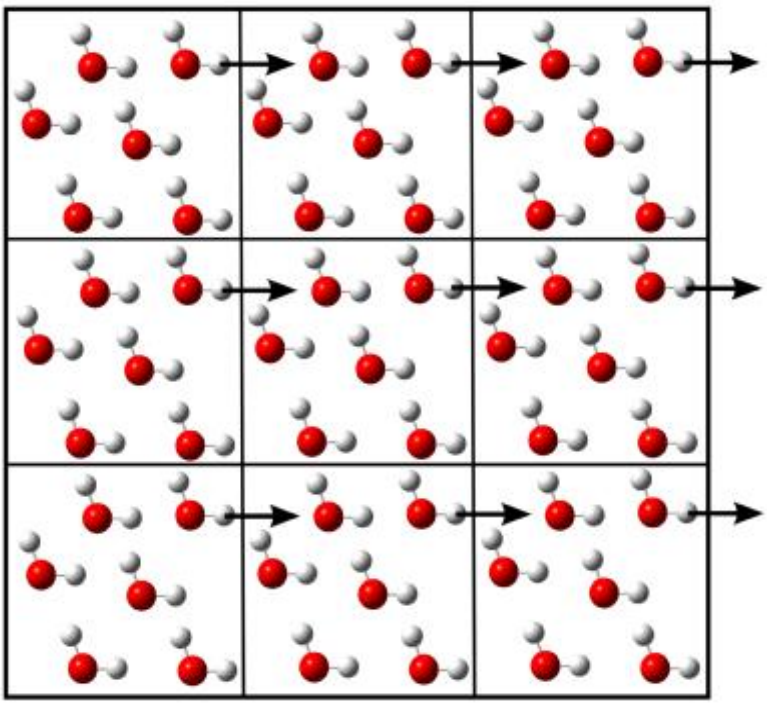


Fig 3.2 Schematic representation of periodic boundary conditions in two dimensions.

3.0.3 Control of Temperature and Pressure

Molecular dynamics simulations can be performed in different ensembles. This means that certain state functions have to be constant. Generally three different ensembles are used in MD simulations: NVE, NVT and NPT. In NVE ensemble (micro canonical) the number of particles, volume and energy are kept constant. Since not many real systems are thermally isolated, it is common to simulate at constant temperature or pressure. In an NVT ensemble (canonical) the temperature is kept constant along with the volume and the number of particles. In the NPT ensemble (isobaric and isothermal) both temperature and pressure are maintained constant.

In order to control pressure and temperature different algorithms have been developed. Thermostats and barostats are applied to couple temperature and pressure respectively. Berendsen thermostat couple the system temperature to an external heat bath. Any deviation from the desired temperature is corrected according to:

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau} \qquad (1.18)$$

Where τ is the time constant representing how often the system and the heat bath are coupled. This thermostat anyway suppresses the kinetic energy fluctuations, so

it does not generate a proper canonical ensemble. The velocity rescaling thermostat, which is based on the same idea of the Berendsen thermostat, produces a proper canonical ensemble with a correct kinetic energy distribution:

$$dK = (K_0 - K)\frac{dt}{\tau_T} + 2\sqrt{\frac{KK_0}{N_f}}\frac{dW}{\sqrt{\tau_T}} \quad (1.19)$$

Where K is the kinetic energy, N_f is the number of degrees of freedom and τ_T is related to τ by:

$$\tau = 2C_v\,\tau_T/N_f k \quad (1.20)$$

Where C_v the total heat capacity, k is is Boltzmann's constant. Other thermostats are also available like the Nosé-Hover thermostat.

Pressure can also be controlled by different algorithms. Berendsen barostat is very similar to the temperature coupling. Coordinates and box vectors are rescaled with a matrix μ, coupling the pressure of the system to an imaginary pressure bath:

$$\frac{dP}{dt} = \frac{P_0 - P}{\tau_P} \quad (1.21)$$

Where τp is the time constant representing how often the system and the pressure bath are coupled and it is related to the scaling matrix by:

$$\mu_{ij} = \delta_{ij} - \frac{n_{PC}\Delta t}{3\tau_p}\beta_{ij}\{P_{0ij} - P_{ij}(t)\} \quad (1.22)$$

The scaling can also be computed in just one or two box dimensions. Again the Berendsen barostat produces the correct average pressure but not the correct NPT ensemble. The Parrinello-Rahman pressure coupling, which is similar to the Nosé-Hoover temperature coupling, gives the correct NPT ensemble. In the Parrinello-Rahman barostat the box vectors are represented by a matrix b and they change according to:

$$\frac{db^2}{dt^2} = VW^{-1}b'^{-1}\left(P - P_{ref}\right) \quad (1.23)$$

3.0.4 Trajectory Analyses

A fundamental step in MD simulations is the analysis of the trajectories. The nature of the investigation and the peculiar properties of the molecules involved in the study affect very much the kind of analyses needed to extract relevant information from the trajectories. The analyses of the data can vary very much

depending on the studied system. However some analyses are almost always performed in protein simulations. Analyses like root mean square deviation (RMSD), root mean square fluctuation (RMSF), number of hydrogen bonds (H-bonds), solvent accessible surface (SAS), radius of gyration (RoG) and secondary structure are usually always computed when a protein is the main body of the investigation.

The RMSD gives the average distance between the protein and a reference structure. The reference structure is usually the starting configuration of the production run or the X-ray structure. It can be calculated either on all or on a specific selection of atoms of the protein (usually the backbone atoms N, O, C and Cα).

$$RMSD(t_1,t_2) = \left[\frac{1}{M}\sum_{i=1}^{N} m_i \|r_i(t_1) - r_i(t_0)\|^2\right]^{\frac{1}{2}} \qquad (1.24)$$

Where r_i(t0) are the positions of the selected atoms of the reference structure.
The RMSF represents a measure of the deviation of atoms from reference positions. The difference with RMSD is that in RMSF the average is computed over time while in RMSD is computed over atoms. Moreover RMSD gives a values for every time step while RMSF gives a value for every atom.

$$RMSF(t_1,t_2) = \left[\frac{1}{T}\sum_{t=1}^{T}(r_i(t_1) - r_i(t_0))^2\right]^{\frac{1}{2}} \qquad (1.25)$$

The SAS is a measure of the protein surface accessible to the solvent. The idea was first proposed by Lee and Richards. The SAS is usually computed by an algorithm which uses a small sphere to probe the protein surface. Different solvents can be investigated adjusting the radius of the probe molecule.

The RoG gives a rough measure of the compactness of a protein. It is useful to characterize the structure of a protein during its denaturation process.

$$R_0G = \left(\frac{\sum_i ||r_i||^2 m_i}{\sum_i m_i}\right)^{\frac{1}{2}} \qquad (1.26)$$

Where m_i is the mass of atom *i* and r_i is the position of atom *i* with respect to the center of mass of the protein.

The number of H-bonds is a simple and important parameter for protein molecules. In all H-bonds there is always a group which donates (donor, D) and a group that accepts (acceptor, A) the hydrogen bond. In proteins the donor is always an OH or an NH2, while the acceptor is always a C=O or a N. It is usually computed using a geometrical criterion. Two parameters have to be set for the

calculation of the H-bonds: the distance between the donor and the acceptor atoms and the angle DHA. The former is usually set between 2.9 and 3.4 Å while the former with a cut-off between 30° and 45°.

The secondary structure can be generally computed in two ways: through the DSSP algorithm or mapping the torsional angles with a Ramachandran plot.

The length and the size (in terms of Gb) of trajectories can represent a problem during the analysis. Depending on the number of particles and the trajectory length, files of several hundreds of Gb can be easily produced. Single analyses can require considerable amount of time and produce large amount of files. In these cases homemade scripts represent a smart way to handle the enormous number of files generated by the single analyses.

3.1 GROMACS OVERVIEW

GROMACS is an acronym for Groningen Machine for Chemical Simulation. It was developed at the University of Groningen, The Netherlands, in the early 1990s.This open-source project is written in ANSIC, and contains about 100 utility and analysis programs which allow users to perform molecular simulations and energy minimization (EM) for biological molecules. It is one of the most commonly used molecular dynamics simulation packages. The following is a list of the main features the GROMACS has.

1. Features for generating topologies and coordinates

   pdb2gmx – converts pdb files to topology and coordinate files.

   editconf – edits the box and writes subgroups

   genbox – solvates a system

   genion – generates mono atomic ions on energetically favorable positions

2. Features for running a simulation.

   grompp – makes a run input file

   mdrun – performs a simulation, does a normal mode analysis or an EM.

3. Features for processing properties

   g_energy – writes energies to xvg files and displays averages

   g_gyrate – calculates the radius of gyration

   g_potential – calculates the electrostatic potential across the box

   g_density – calculates the density of the system

4. Features for processing files.

   trjconv – converts and manipulates trajectory files.

5. Analysis tools.

   g_rms – calculates rmsd's with a reference structure and rmsd matrices

   g_rmsf – calculates atomic fluctuations.

The available potential energy functions such as the AMBER, CHARMM, GROMOS, OPLS / AMBER, etc. provide reasonably good accuracy with reasonably good computational efficiency. Therefore, we have the needed information to calculate the trajectory that describes the positions, velocities and acceleration of the particles at different time, and we can determine the detailed information about the molecule. In general, there are three stages in molecular dynamics simulation: preparation of the input, production molecular dynamics, and analysis of the result.

Stage I: Preparation

This stage has multiple steps including generating the topology file; defining a box and filling it with solvent, and adding any counter-ions to neutralize the system; performing energy minimization to provide stable simulation; performing equilibration for sufficient time to get stable pressure, temperature and energy.

Stage II: Production

This stage is the longest stage resulting in a trajectory containing coordinates and velocities of the system.

Stage III: Analysis

The last stage includes analysis of the resulting trajectory and data files to

obtain information on the property of the molecule. Some important quantities calculated in this stage include RMS difference between two structures, RMS fluctuations, and rigidity or constant force, etc

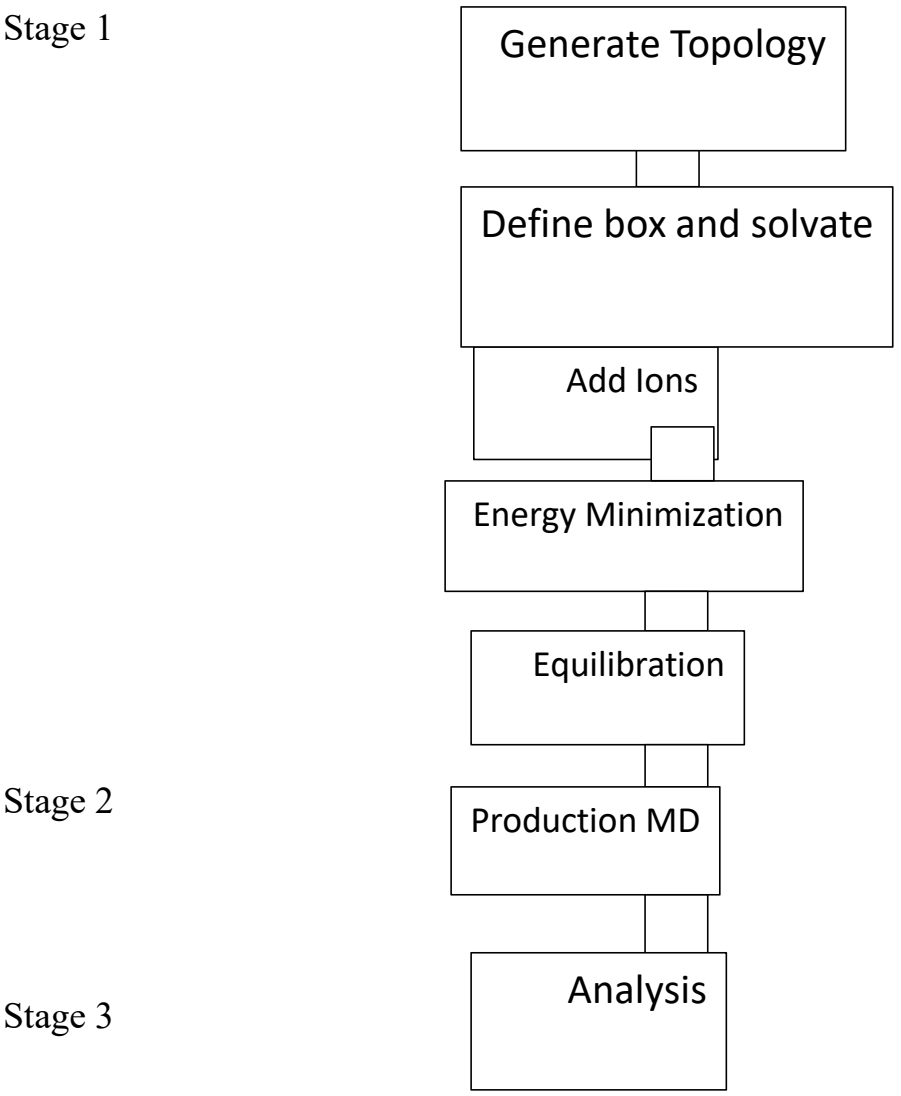


Figure 3.3. Three stages in molecular dynamic simulation: preparation of the input, production molecular dynamic and analysis of the result.

Among many molecular dynamics simulation packages, GROMACS, CHARMM, AMBER, and NAMD are most commonly used. We will use GROMACS in our study because it is open-source, and popular in the study of protein. Before we develop a tool to analyze the GROMACS data, we need to understand GROMACS.

3.1.1 Gromacs Installation on Linux Ubuntu 14.04

Before Gromacs can be installed, a pre-requisite compiler like g++ must be installed, also with cmake and Fast Fourier Transform in the West (FFTW) must be properly installed. Making sure your system is updated open a terminal and do the command:

sudo add-apt-repository

ppa:ubuntu-toolchain-r/test

sudo apt-get update

Next thing is to install g++ from the terminal with the command:

sudo apt-get install g++

After going through a download process a cmake latest version [16] should be downloaded from web browser which can be gotten from cmake website and its source code should be downloaded. FFTW should also be downloaded [17] and also Gromacs source code from the Gromacs website [18].

3.1.2 Installing Cmake-3.3.4

Cmake can be installed through terminal, getting to the file directory through the terminal with the command:

ls

cd Downloads

ls

Unzip the shown cmake file with the command:

tar xzvf cmake-3.0.1.tar.gz

or

tar xzvf cmake and press the tab button

After unzipping the cmake file, enter the command to open the unzipped folder:

cd cmake-3.3.4

ls

Run the configuration by entering the command:

./configure

Then enter the command below to continue:

make –j**

Note that ** in the above command means the number of cords or threads of the PC.

This is will take a little while to run. After running, enter the command:

sudo make install –j**

After the above install process, go to computer, open system file, type cmake in the search tab, select and copy cmake binary, go back to computer, open file system, open usr, open bin as root and paste the cmake binary. The cmake is fully installed.

3.1.3 Installing Fftw-3.3.4 (Fast Fourier Transform In the West)

Type cd.. (This is just like going backwards i.e to Downloads).

Enter the command:

ls

tar xzvf fftw-3.3.4.tar.gz

After unzipping, type:

ls

cd fftw-3.3.4

ls

./configure

This is going to run for a while, when it's done type the command:

make -j**

When it is done running, type:

make install -j**

FFTW is installed.

3.1.4 Installing Gromacs-4.6.5

Type cd.. (This is just like going backwards i.e. to Downloads).

ls

tar xzvf gromacs.4.6.5.tar.gz

ls

cd gromacs-4.6.5

Next thing here is to make an mkdir directory, type the command:

mkdir build

ls

cd build

ls

Now direct cmake to the main gromacs folder which is in downloads.

Open downloads, do not open the gromacs folder, click and drag into the terminal right after the cmake.

cmake gromacsfolder –DGMX_BUILD_OWN_FFTW=ON

Type the command:

make -j**

This will take a while, after it's done you type:

sudo make install -j**

Next,

source/usr/local/gromacs/bin/GMXRC

Gromacs is installed.

CHAPTER 4

4.1 Inputs and Outputs/Results

4.1.1 T4-Lysozyme Protein

Lysozyme is a small, stable enzyme, making ideal for research into protein structure and function.

Alexander Fleming discovered lysozyme during a deliberate search for medical antibiotics. Over a period of years, he added everything that he could think of to bacterial cultures, looking for anything that would slow their growth. He discovered lysozyme by chance. One day, when he had a cold, he added a drop of mucus to the culture and, much to his surprise, it killed the bacteria. He had discovered one of our own natural defenses against infection. Unfortunately, lysozyme is a large molecule that is not particularly useful as a drug. It can be applied topically, but cannot rid the entire body of disease, because it is too large to travel between cells. Fortunately, Fleming continued his search, finding a true antibiotic drug five years later: penicillin.

Lysozyme is a 164-residue protein with antibiotic effect. There are plenty of lysozyme structures in the Protein Data Bank [19] but many are bound to special

compounds or determined at special conditions such as high pressure. Choose the entry 1LYD with 2°A resolution, and download it as 1LYD.pdb.

STRUCTURE

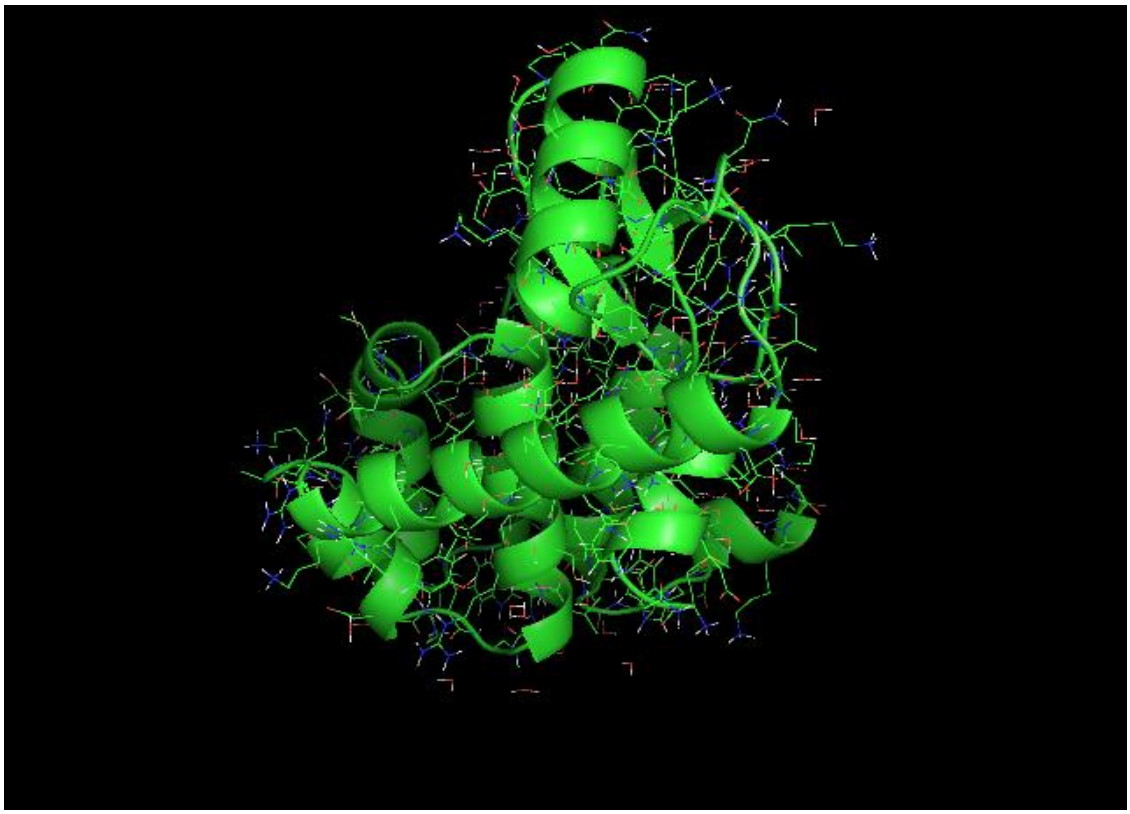
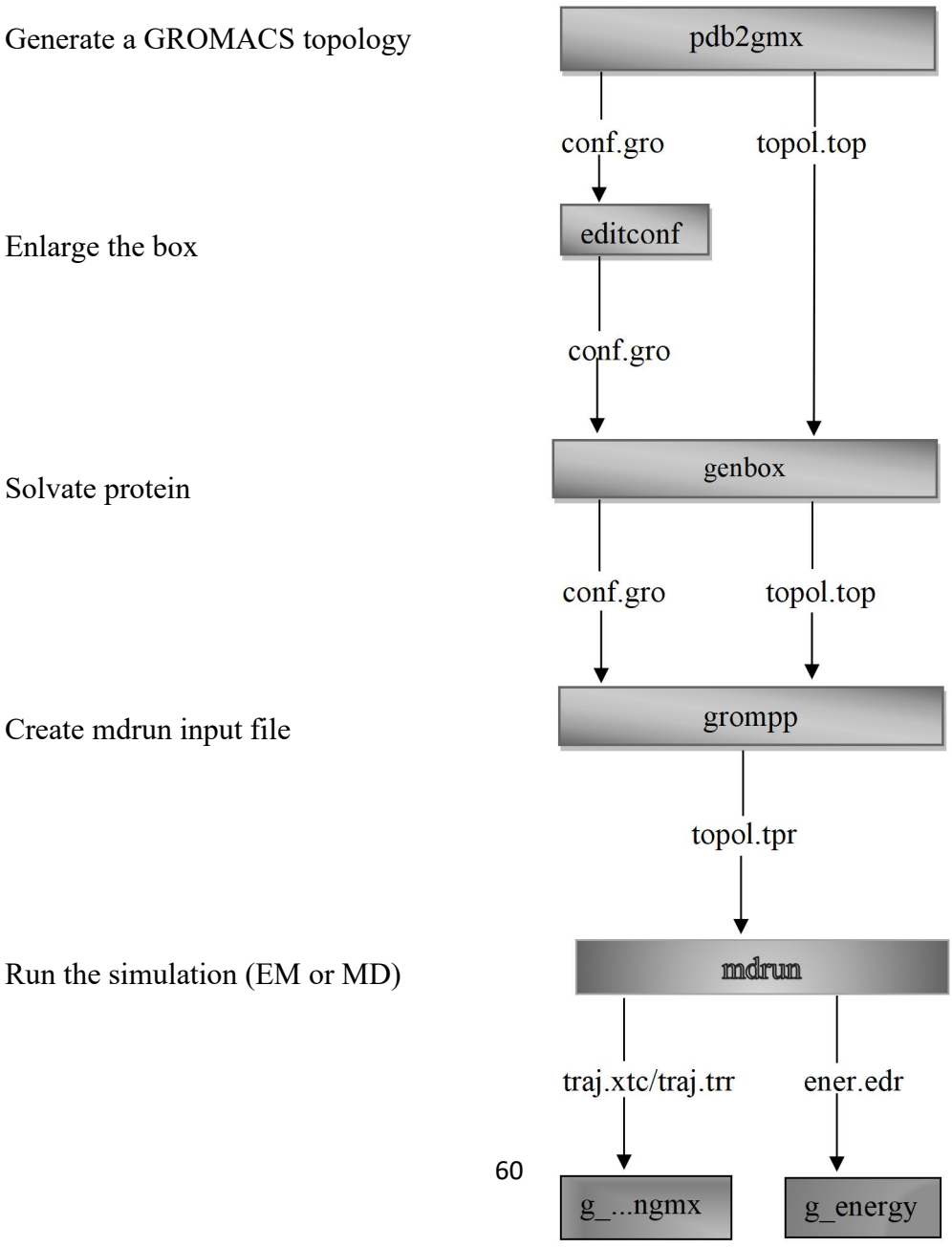


Figure 4.1 shows a cartoon representation of the lysozyme structure 1LYD from PDB, with side chains shown as sticks. Including Hydrogen, the protein contains almost 2,900 atoms. Ray-traced image generated with PyMOL

4.1.2 Preparation of Input/output Data

GROMACS flowchart



4.2 Generate a Protein Topology

This is one of the most delicate parts of the MD input preparation since the behavior of the molecule will depend critically on this topology.

In addition to the coordinates and velocities that change each step, simulations also need a static description of all atoms and interactions in the system, called topology. In GROMACS, this is created from the PDB structure by the program pdb2gmx, which also adds all of the hydrogen atoms that are not present in x-ray structures. We will work with the OPLS-AA force field, the TIP3P water model and accept the default choices for all residue protonation states, termini, disulfide bridges, etc. The command to use is then:

```
pdb2gmx -f 1yd.pdb -water tip3p
```

The structure will be processed by pdb2gmx, and will be prompted to choose a force field:

Select the Force Field:

The force field will contain the information that will be written to the topology. Read thoroughly about each force field and decide which is most applicable. Here, **the option 13 that corresponds to the 53a6 parameterization of the GROMOS 96 force field** at the command prompt, followed by 'Enter'.

The new files been generated: **topol.top**, and **posre.itp** is a GROMACS-formatted structure file that contains all the atoms defined within the force field (i.e., The 20 natural amino acids form part of this database, as well as many other molecular groups including several solvent molecules and lipids, but unfortunately not all the molecules can be handled in this way.). The **topol.top** file is the system topology. The **posre.itp** file contains information used to restrain the positions of heavy atoms.

4.3 Creating a Simulation Box

The default box is taken from the PDB crystal cell, but a simulation in water requires something larger. The box size is a trade-off, however: volume is proportional to the box side cubed, and more water means the simulation is slower. The easiest option is to place the solute in the center of a cube, with greater than 0.8 nm to the box sides. The drawback with this is that a cube wastes volume in the

corners—the ideal case would a sphere, but, as mentioned in the theory section, we also require periodic boundary conditions, which excludes spheres. There are, however, periodic cells, such as a 'truncated octahedron or rhombic dodecahedron' that are more spherical than a cube. This is far from trivial to see in three dimensions, (very useful for membrane simulations). The box creation is accomplished with:

```
editconf -f 1lyd.pdb -bt dodecahedron -d 0.8 -o 1lyd.pdb
```

The above command centers the protein in the box (-c), and places it at least 0.8 nm from the box edge (-d 0.8)). The box type is defined as a cube (-bt dodecahedron). The distance to the edge of the box is an important parameter. The new conformation is written to the file **box.gro.** Since periodic boundary conditions will be used, the minimum image convention must be satisfied. That is, a protein should never see its periodic image, otherwise the forces calculated will be spurious. Specifying a solute-box distance of 0.8 nm will mean that there are at least 1.0 nm between any two periodic images of a protein.

4.4 Adding Solvent Water

The last step before the simulation is to add water in the box to solvate the protein. This is performed by using a small pre-equilibrated system of water

coordinates that is repeated over the box, with overlapping water molecules removed. The lysozyme system will require roughly 6,000 water molecules, which increases the number of atoms significantly (from 2,900 to more than 20,000). GROMACS does not use a special pre-equilibrated system for TIP3P water because water coordinates can be used with any model—the actual parameters are stored in the topology and force field. In GROMACS, a suitable command to solvate the new box would be:

```
genbox -cp 1lyd.pdb -cs -p topol.top -o 1lydpdb
```

Solvent coordinates (-cs) are taken from an SPC water system, and the -p flag adds the new water to the topology file. The resulting system is illustrated in Fig. 4.2. The output is called "lysozyme_w.pdb". Take a look to the output file "lysozyme_w.pdb" using PyMOL. Probably the view of the system is a bit strange with the protein in a corner of the simulation box. This is due to the geometry of the box. The coordinates can be modified to produce a better view with your preferred molecular viewer using the following command:

```
echo 0 | trjconv -s 1lyd.pdb -f 1lyd.pdb -o lyso_view.pdb -pbc atom -ur compact
```

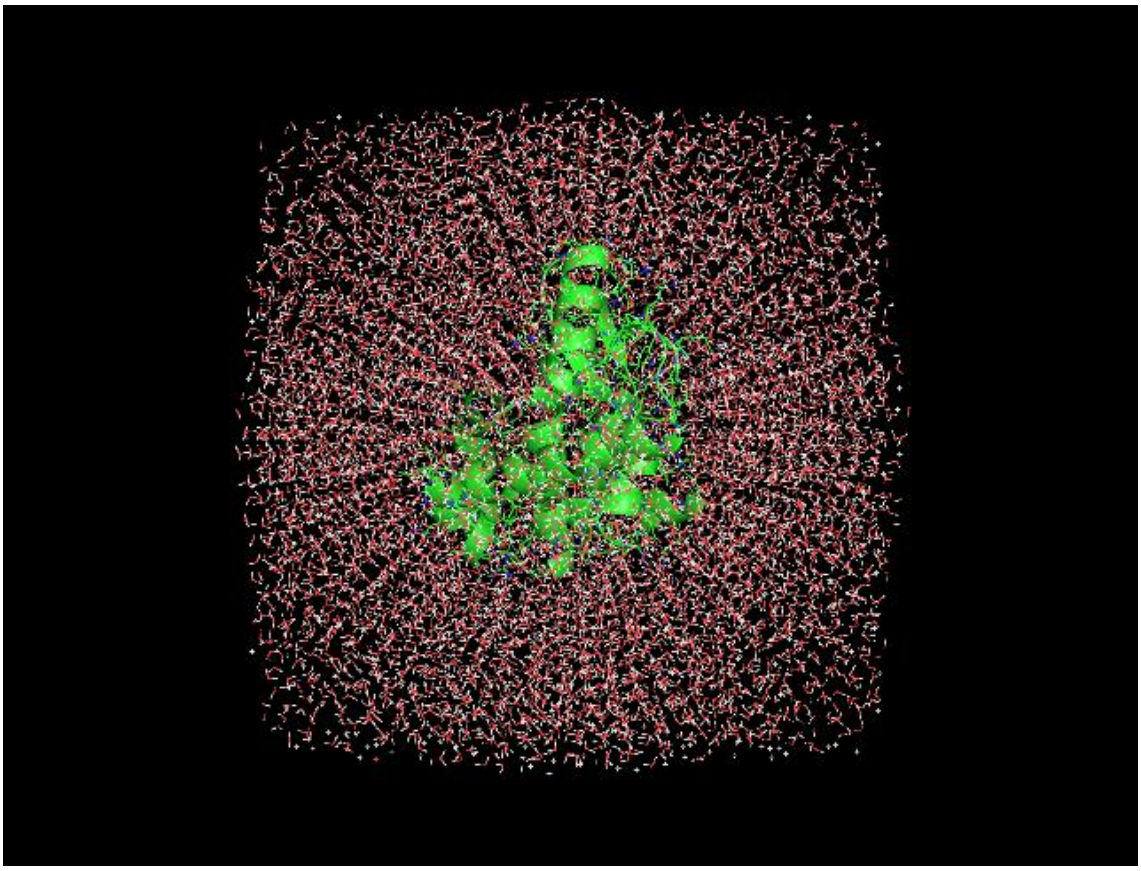


Fig 4.2 Lysozyme solvated in water in a triclinic box representing a rhombic dodecahedron (30% lower volume than a cube)

4.5 Energy Minimization

The added hydrogen and broken hydrogen bond network in water would lead to very large forces and structure distortion if molecular dynamics was started immediately. To remove these forces, it is necessary to first run a short energy minimization. The aim is not to reach any local energy minimum, therefore, 500

steps of steepest descent (as mentioned in the theory section) works very well as a stable rather than maximally efficient minimization. Non-bonded interactions and other settings are specified in a parameter file (**em.mdp**); it is only necessary to specify parameters where we deviate from the default value.

GROMACS uses a separate preprocessing program, grompp, to collect parameters, topology, and coordinates into a single run input file (**em.tpr**) from which the simulation is started (this makes it easier to move it to a separate supercomputer). These commands are:

grompp -f em.mdp -p topol.top -c 1lyd_w.pdb -o em.tpr

From the output of this program the charge of the system is +8. It is not realistic to have a system with net charge, so some ions will be introduced. At least 8 negative ions are needed to compensate for this charge. Add 4 Na + and 12 Cl – by:

genion -s em.tpr -o 1lyd_ions.pdb -p - -nn 12 -pname NA -nname CL

Select the option **13,**

Build a binary file using the command:

grompp -f em.mdp -p topol.top -c 1lyd_ions.pdb -o em.tpr

Start the minimization with the command:

```
mdrun -v -deffnm em
```

The -deffnm is a smart shortcut that uses “em” as the base filename for all options but with different extensions.

The minimization should finish after 1561 steps, as indicated in the "em.mdp" file. As a result several files should be created, including "confout.gro" that contains the coordinates of all the atoms of the system -the same information that the pdb file.

The potential energy was calculated for after the complete energy-minimization using the command:

```
g_energy -f em.edr -o potential.xvg
```

Select the option **9.**

A graph was plotted directly from the terminal using the command:

```
xmgrace potential.xvg
```

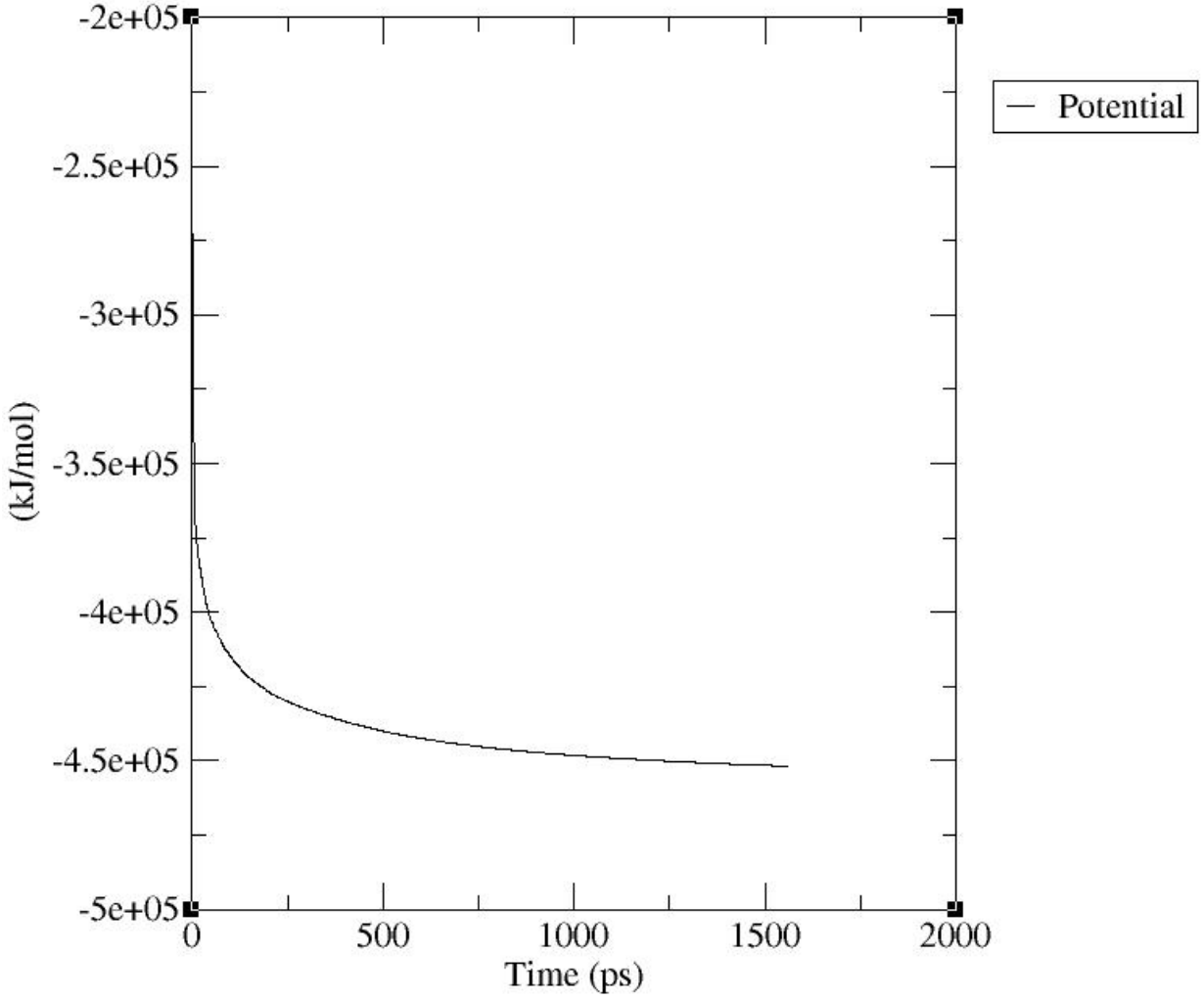


Fig 4.3 Graph of the potential

4.6 Equilibrate the Water around the Protein

Before starting the simulation it is convenient to equilibrate the water molecules by performing a short MD simulation with the position of the heavy protein atoms restrained. For this we will need to create a file (call it "eqwat.mdp").

Again the coordinates, topology and simulation parameters files need to be assembled into a binary file. The coordinates file employed is the output file "confout.gro" obtained from the minimization:

```
grompp -f eqwat.mdp -p topol.top -c confout.gro -o eqwat.tpr
```

```
mdrun -v -s eqwat.tpr
```

Again, after a couple of minutes, the short equilibration finish and several new files appear.

4.7 Production Run

The difference between equilibration and production run is minimal: the position restraints and pressure coupling are turned off, we decide how often to write output coordinates to analyze (say, every 1,000 steps), and start a significantly longer simulation. How long depends on what is been studied, and that should be decided before starting any simulations. We will perform a 10-ns simulation (500 steps).

Introduce this information in a file called "run.mdp". The binary file is created by combining again the coordinates, topology and simulation parameters files. Perform the production run as:

```
grompp -f run.mdp -p topol.top -c confout.gro -o run.tpr
```

```
mdrun -v -s run.tpr
```

4.8 Trajectory Analysis

4.8.1 Deviation from X-Ray Structure

One of the most important fundamental properties to analyze is whether the protein is stable and close to the experimental structure. The standard way to measure this is the root mean square displacement (RMSD) of all heavy atoms with respect to the x-ray structure. The first is **trjconv**, which is used as a post-processing tool to strip out coordinates, correct for periodicity, or manually alter the trajectory (time units, frame frequency, etc.). The protein will diffuse through the unit cell, and may appear to "jump" across to the other side of the box. GROMACS has a finished program to do this, as:

```
trjconv -s run.tpr -f traj.xtc -o traj_noPBC.xtc -pbc mol -ur compact
```

Select **4** ("**Backbone**") for output

GROMACS has a built-in utility for RMSD calculations called rms. To use **rms**, issue this command:

```
g_rms -s em.tpr -f traj_noPBC.xtc -o rmsd_xtal.xvg -tu ns
```

The **-tu** flag will output the results in terms of ns, even though the trajectory was written in ps. This is done for clarity of the output (especially for long simulation - 1e+05 ps does not look as nice as 100 ns). The output plot will show the RMSD relative to the structure present in the minimized, equilibrated system:

```
xmgrace rmsd_xtal.xvg
```

RMSD

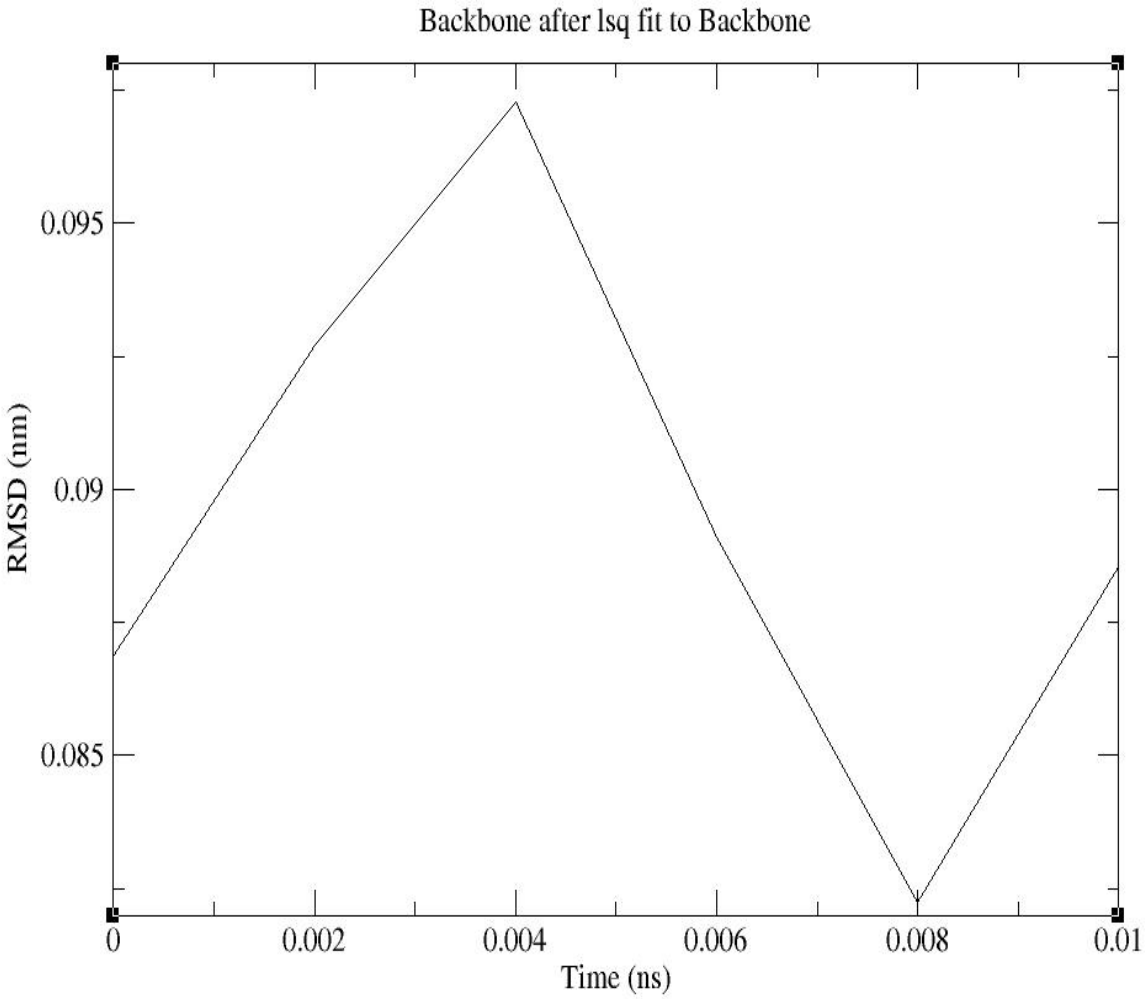


Fig 4.4 Graph of rmsd

The RMSD illustrated in Fig. 4.4 indicates that the structure is stable. It increases rapidly in the first part of the simulation and decreases to the minimum as the simulation continued, roughly the resolution of the x-ray structure. The difference is partly caused by limitations in the force field, but also because atoms in the simulation are moving and vibrating around an equilibrium structure.

4.8.2 Comparing Fluctuations with Temperature Factors

Vibrations around the equilibrium are not random, but depend on local structure flexibility. The root mean square fluctuation (RMSF) of each residue is straightforward to calculate over the trajectory, but, more important, they can be converted to temperature factors that are also present for each atom in a PDB file. To do this, enter the command:

```
g_rmsf -s run.tpr -f run.xtc -o rmsf.xvg -oq bfac.pdb
```

Choose the group “C-alpha” to get one value per residue. Figure 4.5 displays both the residue RMSF from the simulation as well as the calculated and experimental temperature factors. Using:

```
xmgrace rmsf.xvg
```

RMS FLUNCTUTAION

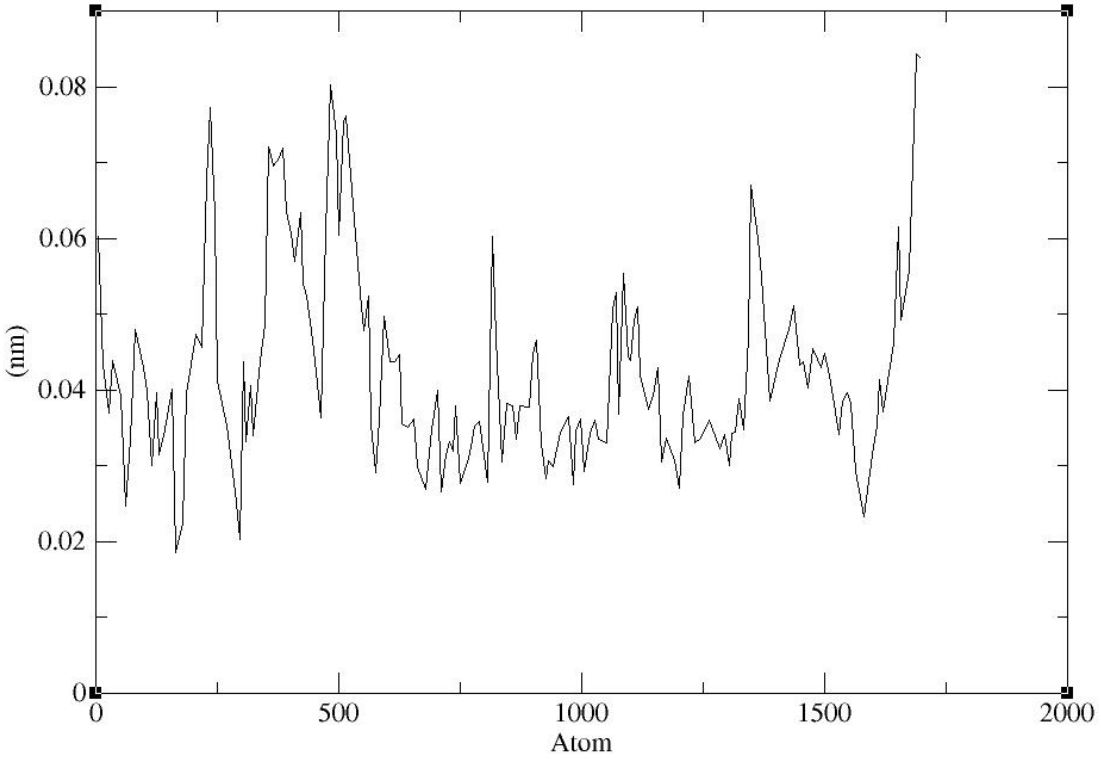


Fig 4.6 Graph of rmsf

4.8.3 Radius of gyration

The radius of gyration of a protein is a measure of its compactness. If a protein is stably folded, it will likely maintain a relatively steady value of *Rg*. If a protein unfolds, its *Rg* will change over time. To analyze the radius of gyration for lysozyme in the simulation use the command:

g_gyrate -s run.tpr -f traj_noPBC.xtc -o gyrate.xvg

Select the option **4**,

xmgrace gyrate.xvg

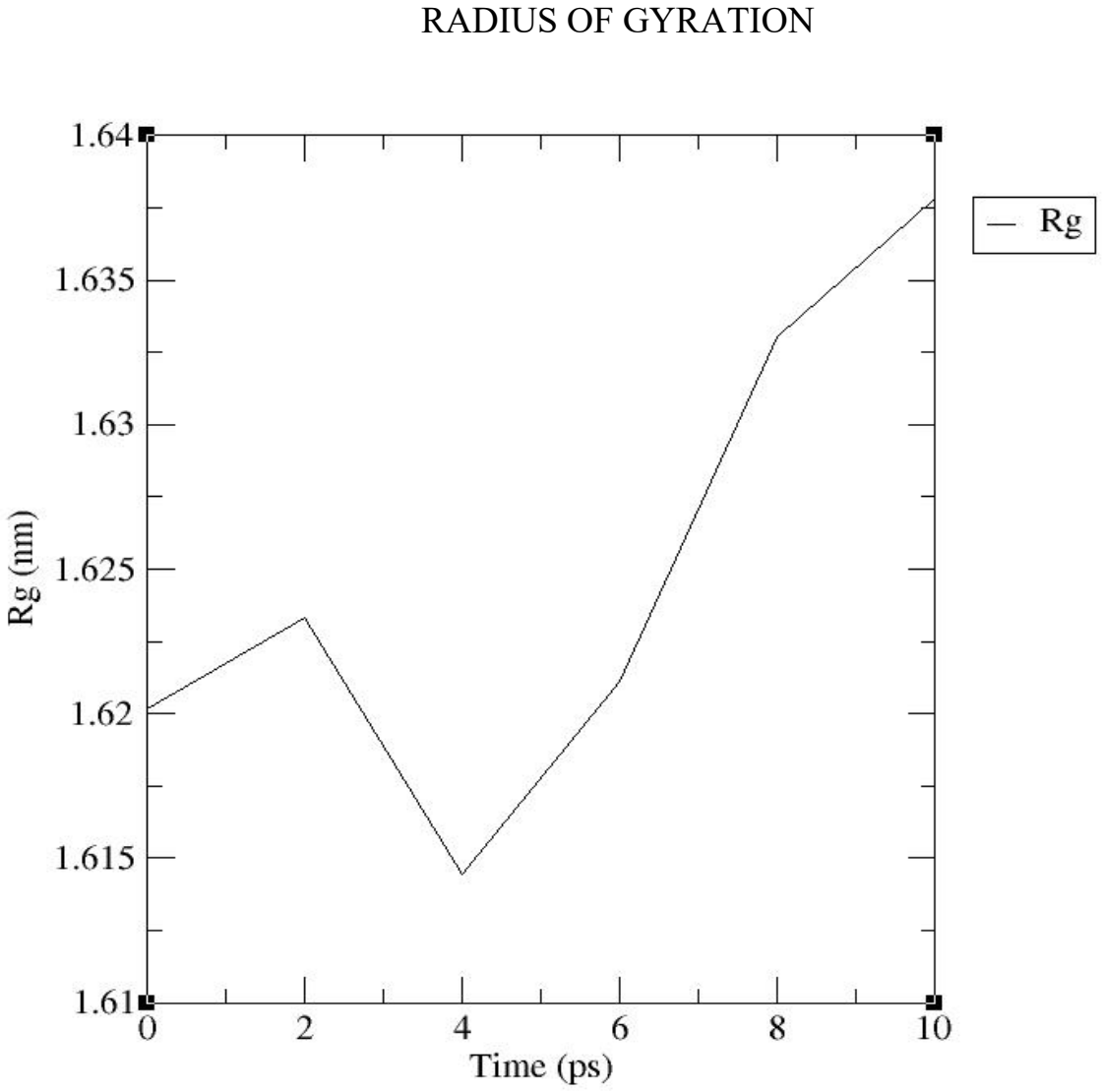


Fig 4.7 Graph of radius of gyration

From the reasonably invariant *Rg* values, the protein is stable in its compact (folded) form over the course of 10 ps at 300 K. This result is not unexpected, but illustrates an advanced capacity of GROMACS analysis that comes built-in.

4.8.4 Secondary Structure

Another measure of stability is the protein secondary structure. This can be calculated for each frame with a program such as DSSP. The GROMACS program "do_dssp" can create time-resolved secondary structure plots.Because the program writes output in a special xpm (X pixmap) format. Enter the command:

```
do_dssp -s run.tpr -f run.xtc
xpm2ps -f ss.xpm -o ss.eps
```

Use the group "protein" for the calculation.

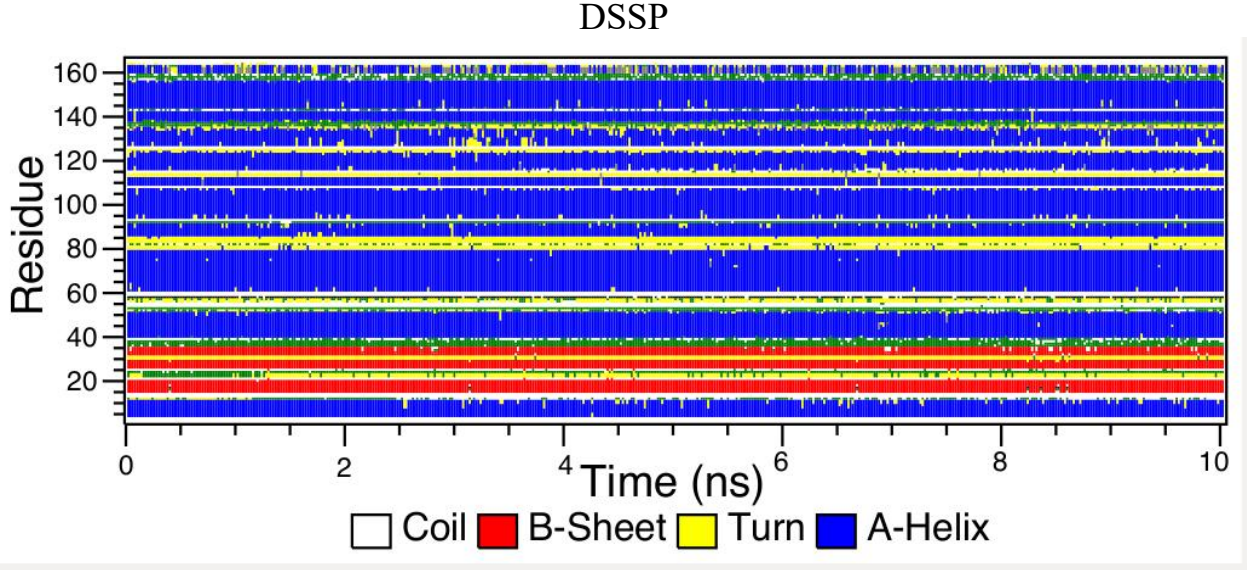


Fig. 4.8 Local secondary structure in lysozyme as a function of time during the simulation, according to the DSSP definition. Note how helices sometimes are unrolled slightly at the start and end but the overall structure is very stable over 10 ns

CHAPTHER FIVE

5.0 Conclusion

A molecular dynamics simulation of antifreeze protein was successfully ran using GROMACS. Simulations require a lot of care from the user just as with experimental techniques.

Simulations using empirical force fields are still very limited in the range of timescales accessible, but recent techniques based on distributed computing and Markovian state models have been able to probe dynamics in the millisecond range without extending individual simulations to those scales.

REFERENCE

[1] ALDER B. J. AND WAINWRIGHT T. E. (1959). Studies in Molecular Dynamics. I. General Method. J. Chem. Phys., 31:459–466.

[2] BERENDSEN H.J.C., VAN DER SPOEL D., AND VAN DRUNEN R. (1995). GROMACS: a Message-Passing Parallel Molecular Dynamics Implementation. Comput. Phys. Commun., 91:43–56.

[3] BROOKS B. R., *et al*. (1983). CHARMM: a Program for Macromolecular Energy, Minimization, and Dynamics Calculation. J. Comp. Chem., 4:187–217.

[4] JORGENSEN W. L. AND TIRADO-RIVES J. (2005).Potential Energy Function for Atomic-Level Simulations of Water and Organic and Biomolecular Systems. Proc. Natl. Acad. Sci. USA, 102:6665–6670.

[5] KABSCH W. AND SANDER C. (1983). Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Biopolymers, 22(12):2577–2637.

[6] LEVITT M. AND WARSHEL A. (1975). Computer Simulations of Protein Folding. Nature, 253:694–698.

[7] METROPOLIS N., *et al*. (1953). Equation of State Calculations by Fast Computing Machines. J. Chem. Phys., 21:1087–1092.

[8] MORSE P. M. Diatomic Molecules According to the Wave Mechanics.

(1929). II. Vibrational Levels. Phys. Rev., 34:57–64.

[9] PAULING L., COREY R. B., AND BRANSON H. R. (1951). The Structure of Proteins: Two Hydrogen-Bonded Helical Configurations of the Polypeptide chain. 205–211.

[10] PERUTZ M. F., *et al*. (1960). Structure of Haemoglobin. A Three-Dimensional Fourier Synthesis at 5.5 Å Resolution, Obtained by X-ray Analysis. Nature, 185:416–422.

[11] SANGER F. AND TUPPY H. (1951). The Amino-Acid Sequence in the PhenylalanylChain of Insulin 1. The Identification of Lower Peptiedes from Partial Hydrolysates. Biochem. J., 49:463–481.

[12] SCHRÖDINGER E. (1944). What is Life? The Physical Aspect of the Living Cell. Cambridge University Press.

[13] SIMMERLING C., STROCKBINE B., AND ROITBERG A. E. (2002). All-Atom Structure Prediction and Folding Simulations of a Stable Protein. J. Am. Chem. Soc., 124:11258–11259.

[14] RAMACHANDRAN G. N., *et al.* (1963). Stereochemistry of Polypeptide Chain Configurations. J. Mol. Biol., 7:95–99.

[15] RAHMAN A. (1964). Correlations in the Motion of Atoms in Liquid Argon. Phys. Rev., 136:405–411.

[16] http//www.cmake.org

[17] http//www.fftw.org

[18] http//www.gromacs.org

[19] http//www.pdb.org

Appendix

Potential

Statistics over 1561 steps [0.0000 through 1560.0000 ps], 1 data sets

All statistics are over 1 points

Energy	Average	Err.Est.	RMSD	Tot-Drift
Potential	-439681	6700	-nan	-43503.1 (kJ/mol)

Rmsd

Selected 4: 'Backbone'

Last frame 5 time 0.010

Rmsf

Selected 3: 'C-alpha'

Last frame 5 time 10.000

Radius of Gyration

Selected 4: 'Backbone'

Reading frame 0 time 0.000

Last frame 5 time 10.000