
CHAPTER 3

MATERIALS AND METHODS

3. Materials and Methods:

3.1. Tools and Techniques:

Molecular dynamics (MD) simulation was used to investigate the structural dynamics, aggregation mechanism, dimerization, and interaction of the A β ₁₋₄₂ peptide and α S protein with various small molecules and peptides. The principle and theory of the MD simulation are discussed below.

3.1.1. Molecular dynamics (MD) simulation:

To comprehend the underlying physics of the composition and functioning of biological macromolecules, molecular dynamics simulations are crucial tools. The link between theory and experimentation is created by MD simulations. Experiments are replicated, unseen microscopic details are clarified, and experiments are further explained using MD simulations. The classical equations of motion are numerically solved step by step in an MD simulation, allowing for a detailed examination of molecular interactions. A popular method for examining the time evolution of a system of "particles", usually atoms or molecules with known properties, is molecular dynamics (MD) simulation. Examples of applications include simulating the interactions between molecules in the body and therapeutic drugs, simulating the microscopic mechanism of water droplet freezing, calculating the thermodynamic and rheological properties of various hydrocarbon mixtures and simulating the effects of a particular combination of organic molecules on properties that affect solar cell efficiency. The key benefit of MD simulation is its capacity to increase the complexity horizon that distinguishes "solvable" from "unsolvable" problems.

3.1.1.1. History of MD Simulation:

Alder and Wainwright were the first people to introduce the molecular dynamics simulation approach in the late 1950s [470, 471]. This method was developed in order to study the interactions that occur between hard spheres. Their research has yielded a number of discoveries that provide new perspectives on the behavior of simple liquids. After this, the subsequent significant step forward for the MD simulation occurred in 1964 when Rahman carried out the first simulation with a realistic potential for liquid

argon [472]. This was an important step forward for the MD simulation. The first example of an MD simulation being used to model a realistic system was Rahman and Stillinger's simulation of liquid water, which was published in 1974 [473]. 1977 was the year that saw the debut of the very first protein simulations, which were published alongside the modeling of the bovine pancreatic trypsin inhibitor (BPTI) [474]. In the current scientific literature, we may find MD simulations of solvated proteins, protein-DNA complexes, and lipid systems. These simulations address a number of challenges, including the thermodynamics of ligand binding and the folding of tiny proteins, among other things. The simulation techniques have undergone a significant expansion, which has resulted in the development of a wide range of specialized techniques for specific problems. These techniques consist of mixed quantum mechanical-classical simulations, which are of significant significance for enzymatic reactions within the context of full proteins. MD simulation approaches are utilized in a variety of experimental processes, such as x-ray crystallography and nuclear magnetic resonance (NMR) structure determination.

3.1.1.2. Theory of MD Simulation:

MD simulation is essentially based on Newton's second law of motion, often known as the equation of motion, which states that $\mathbf{F}=\mathbf{ma}$, where \mathbf{F} is the force exerted on the particle, \mathbf{m} is the particle's mass, and \mathbf{a} is the particle's acceleration. On the other hand, if the force acting on each atom is known, it is easy to determine the acceleration that each atom in the system experiences. When the equations of motion are integrated, a trajectory is generated that describes the positions, velocities, and accelerations of the particles in relation to time. This information can be used to forecast the system's behavior. These trajectories can be used to derive the properties of the average values and apply them accordingly. This technique is deterministic, which implies that the system's state can be predicted at any time in the past, present or future once the velocities and positions of each atom are known. However, the MD simulation techniques may be time-consuming and costly in terms of computer resources. Despite this, computer prices are decreasing as their processing speeds increase. In order to achieve the most precise findings, simulations of solvated proteins are conducted at the millisecond scale. In a handful of the investigations, simulations that stretch into the millisecond regime have also been described.

Newton's equation of motion is given by:

$$F_i = m_i a_i \dots \dots \dots (3.1)$$

In this equation, F_i denotes the force exerted on particle i , m_i denotes the mass of particle i , and a_i denotes the acceleration of particle i .

Newton's force, F_i can also be expressed as the potential energy gradient,

$$F_i = -\nabla_i V \dots \dots \dots (3.2)$$

By combining the above two equations, the following equation is obtained:

$$-\frac{dV}{dr_i} = m_i \frac{d^2 r_i}{dt^2} \dots \dots \dots (3.3)$$

Here V is defined as the potential energy of the system. The *equation of motion of Newton* can be connected to the derivative of potential energy meant for position changes with regard to time.

The main purpose of numerical integration for Newton's equation of motion is finding an expression that may specify the position $r_i(t+\Delta t)$ at time $t+\Delta t$ in relation to the previously determined locations at time t . To determine the new positions at time $t+\Delta t$, the *Velocity Verlet* method uses both the locations and accelerations at time t and the positions from time $(t+\Delta t)$. The *Velocity Verlet* algorithm does not use specified speeds. Using the Verlet algorithm is significant for the following reasons: i) straight-forwardness and ii) storage requirements are modest. But the main drawback of employing this approach is its moderate precision.

The variant of the *Velocity Verlet* algorithm is the *leap-frog algorithm*, in which the velocities are either explicitly propagated or derived from the positions. The velocities are determined at time $(t + 1/2\Delta t)$ in the leap-frog procedure, and they are then used to calculate the locations, r , at time $(t + \Delta t)$. This causes the velocities and positions to leapfrog one another. The main benefit of the leap-frog method is that speeds are calculated explicitly, while the main drawback is that speeds are not calculated at the same time as positions.

In the leapfrog algorithm, the velocity is used as a **half-time step**:

$$\dot{r}_i \left(t + \frac{\Delta t}{2} \right) = \dot{r}_i \left(t - \frac{\Delta t}{2} \right) + \ddot{r}_i(t) \Delta t \dots \dots \dots (3.4)$$

At the time t , the velocities can be computed from

$$\dot{r}_i(t) = \frac{\dot{r}_i(t+\frac{\Delta t}{2}) + \dot{r}_i(t-\frac{\Delta t}{2})}{2} \dots\dots\dots (3.5)$$

This becomes significant when kinetic energy is required at time t , such as when velocity rescaling is required. The atomic positions needed are then attained from:

$$r_i(t+\Delta t) = r_i(t) + \dot{r}_i(t + \frac{\Delta t}{2}) \Delta t \dots\dots\dots (3.6)$$

A force field is used to depict the time development of bond lengths, bond angles, and torsion, as well as non-bonding van der Waals and electrostatic interactions between atoms. This force field is a set of constants and equations that are meant to copy the shape of molecules and other features of tested structures.

3.1.1.3. Force field (FF):

A force field (FF) is a mathematical expression that shows how the energy of a system changes depending on where the particles are. It comprises an analytical form of inter-atomic potential energy, $\mu(r^N)$, where $r^N = (r^1, r^2 \dots r^N)$, and a set of parameters that enter into this analytical form. Most of the time, these parameters are found through semi-empirical quantum mechanical calculations, ab-initio calculations, or fitting to experimental data like neutron, x-ray, and electron diffraction, NMR, infrared, Raman, and neutron spectroscopy, and so on. The molecules are described as a group of atoms held together via simple elastic (harmonic) forces, and these FF interchange the true potential with a simplified model that is valid in the region being simulated. Preferably, it is easy to evaluate quickly but difficult to repeat the system's interest to be studied. There are many different types of force fields with different levels of complexity that are used to treat different kinds of systems. A classical FF expression might look like this:

$$V(r^N) = \sum_{\text{bonds}} \frac{k_i}{2} (l_i - l_{i,0})^2 + \sum_{\text{angles}} \frac{k_i}{2} (\theta_i - \theta_{i,0})^2 + \sum_{\text{torsions}} \frac{V_n}{2} (1 + \cos(n\phi - \phi_o)) + \sum_{i=1}^N \sum_{j=i+1}^N (4\epsilon_{ij} [(\frac{\sigma_{ij}}{r_{ij}})^{12} - (\frac{\sigma_{ij}}{r_{ij}})^6] + \frac{q_{iq_j}}{4\pi\epsilon_o\epsilon_r r_{ij}}) \dots\dots\dots (3.7)$$

Here, $V(r^N)$: potential energy as a function of the positions (r) of N atoms;

k_i : force constant;

l, l_0 : current and reference bond lengths;

θ, θ_0 : current and reference valence angle;

V_n : barrier height of rotation;

ϕ : torsion angle;

n : multiplicity that determines the number of energy minima during a full rotation;

σ_{ij} : collision diameter for the interaction between two atoms i and j ;

ϵ_{ij} : well depth of the Lennard-Jones potential for the i - j interaction;

q_i, q_j : partial atomic charges on the atoms i and j ;

r_{ij} : current distance between the atoms i and j ;

ϵ_0, ϵ_r : permittivity of the vacuum and relative permittivity of the environment respectively;

ϕ_0 : phase factor that determines where the torsion angle passes through its energy minima.

Bonded interactions, such as bond lengths, angles, and bond rotations, and non-bonded interactions, i.e. van der Waals and electrostatic interactions, make up the majority of the potential energy function. The types of interactions are schematically presented in **Figure 3.1**.


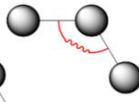
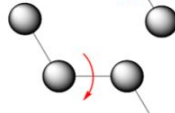
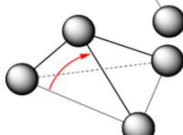

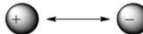
$U(R) = \sum_{\text{bonds}} k_r (r - r_{eq})^2$	<i>bond</i>	
$+ \sum_{\text{angles}} k_\theta (\theta - \theta_{eq})^2$	<i>angle</i>	
$+ \sum_{\text{dihedrals}} k_\phi (1 + \cos[n\phi - \gamma])$	<i>dihedral</i>	
$+ \sum_{\text{impropers}} k_\omega (\omega - \omega_{eq})^2$	<i>improper</i>	
$+ \sum_{i < j}^{\text{atoms}} \epsilon_{ij} \left[\left(\frac{r_m}{r_{ij}} \right)^{12} - 2 \left(\frac{r_m}{r_{ij}} \right)^6 \right]$	<i>van der Waals</i>	
$+ \sum_{i < j}^{\text{atoms}} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$	<i>electrostatic</i>	

Figure 3.1. Schematic illustration of the main contribution to the potential energy function (Taken from [481]).

The first four terms of the equation represent intra-molecular or local contributions to the total energy (bond stretching, angle bending, dihedral, and improper torsions). The last two terms of the equation describe the repulsive and van der Waals interactions (represented by a 12-6 Lennard-Jones potential) as well as the Coulombic interactions.

3.1.1.4. Long-range interactions: Ewald sum:

The Ewald Summation is a widely used method for estimating the electrostatic interactions in computer simulations of condensed-matter systems [475]. The errors that result from truncating the infinite real and Fourier-space lattice sums in the formulation of Ewald Sum are analyzed. For the Fourier-space cutoff, an optimal choice with a screening parameter of 7 is extracted. Normally, it is noticed that the number of Fourier space vectors required to achieve a particular level of precision scales with 7/3. Nonetheless, this proposed method can be used to regulate the efficient computing parameters for Ewald sums, thereby evaluating the quality of Ewald-sum implementations and comparing the various implementations. This is perhaps the most frequent method for evaluating long-range interactions in MD simulations. The fundamental concept underlying the Ewald sum is to analyze a charge distribution for each site's opposite sign. The distribution of additional charges reveals the interactions between nearby atoms. However, the visible interactions are short-range and might be precisely managed by the cut-off scheme. To compensate for the additional charge distribution, the equal charge distribution with the opposite sign and short-range interaction is averaged in the reciprocal space. As the electrostatic potential due to the screened charge is a quickly decreasing function of \mathbf{r} , it is straightforward to calculate the contribution of a group of screened charges to the electrostatic potential at a specific location \mathbf{r}_i using direct summation. The total potential energy for the long-range Coulomb interaction is given by the expression:

$$\mu_c = \mu_q(\alpha) - \mu_{self}(\alpha) + \Delta\mu(\alpha) \dots\dots\dots (3.8)$$

In the equation, the greater the value of α , the sharper the distribution; consequently, a large number of \mathbf{K} summations are included to improve precision. However, a higher value narrows the screened potential range, allowing us to use a smaller cutoff radius. In

order to improve accuracy and efficiency, the value of α is therefore subject to optimization between the two factors. In the above scales, the Ewald summation is only shown as N^2 . Finchman was able to optimize the summation that scales as $N^{3/2}$ by selecting the appropriate α and k-space summation cut-off \mathbf{K} . In addition, this Ewald summation approach can be optimised through the use of the Fast Fourier Transform (FFT) in the reciprocal summation evaluation process. The Particle Mesh-based technique, on the other hand, employs a constant cutoff for the direct space sum and an FFT-based approximation that scales as $N \log(N)$ for the reciprocal space sum.

3.1.1.5. Dealing with molecules: SHAKE algorithm:

The choice of time step in a molecular system is constrained by the different time scales associated with vibrational degrees of freedom, such as bond vibration, angle stretching, and torsional mode. Hydrogen-containing bonds have a faster vibrational mode, which limits the integration time step to 1 fs. However, if a longer time period is used, these fast degrees of freedom can be restrained while the unconstrained degrees of freedom are solved. Hydrogen bonds have the highest frequency, so they could be constrained during dynamics using The SHAKE algorithm developed by Ryckaert et. al. [476]. The first step of the SHAKE algorithm is to send the equations of motion for an atomic system that are not limited in any way. In addition, the SHAKE algorithm relies on the Lagrange multiplier formalism to maintain constant bond distances. Assuming N_c , the constraint is given by:

$$\alpha_k = r_{k_1 k_2}^2 - R_{k_1 k_2}^2 = 0, \text{ Where } k = 1, 2, 3, \dots, N_c \dots \dots \dots (3.9)$$

The term $R_{k_1 k_2}$ is considered a constrained distance between the \mathbf{k}_1 and \mathbf{k}_2 atoms. The modified constrained equation of motion is defined as:

$$m_i \frac{d^2 r_i(t)}{dt^2} = - \frac{\partial}{\partial r_i} [V(r_1 \dots r_N) + \sum_{k=1}^{N_c} \tau_k(t) \alpha_k(r_1 \dots r_N)] \dots \dots \dots (3.10)$$

In this case, m_i is referred to as the mass of the i^{th} particle and τ_k is the Lagrange multiplier (unknown) for the k^{th} constraint. However, by resolving N_c quadratic linked equations, the unknown multiplier in this modified constrained equation of motion can be determined. Finally, we have discovered the motion equation shown below:

$$r_{k1}(t + \Delta t) = r_{k1}^{uc}(t + \Delta t) - 2(\Delta t)^2 m_{k1}^{-1} \tau_k(t) r_{k1k2}(t) \dots \dots \dots (3.11)$$

In the equation, r^{uc} is the position updates with unconstrained force only. This method is however repetitive till the defined tolerance is specified.

By iteratively modifying particle coordinates, the SHAKE algorithm approach avoids the explicit matrix inversion up until the system satisfies all constraints to within a specified tolerance. In addition to preserving the rigid bonds, constraint algorithms also need to account for *constraint decay*, which is the rise in deviation from the ideal lengths brought on by the accumulation of numerical mistakes. However, the iterative algorithms implicitly provide precise constraint decay by requiring convergence within a given tolerance at each time step. Frequent corrections and checks are made to the confined distance deviations originating from the original values. However, because there is no built-in feedback system for noticing changes in distance, the non-iterative algorithms required an explicit method that can counteract the constraint degradation.

3.1.1.6. Periodic Boundary conditions:

In order to comprehend periodic boundary conditions, we will imagine a system with N particles interacting at a temperature T and a volume V . The periodic boundary conditions identical to the *2D Ising system* must be applied such that the system is surrounded by copies of itself. This establishes that, for a system of particles, when a particle exits the centre box on one side, it enters the central box on the opposite side. As seen in **Figure 3.2**, the atoms for the molecules are positioned in a box comprised of translated copies of atom coordinates. From **Figure 3.2**, it can be observed that particle 1 in the centre box has the potential to interact with many copies of particle 3 existing in the central box. In addition, it is appropriate to evaluate a distinct interaction between particles 1 and 3, with the interaction resulting in the shortest inter-atomic distance being the obvious option. Consequently, this procedure is known as the **nearest image convention**. A known periodic three-dimensional array surrounds the inner cell. When an atom crosses the barrier and enters the opposite side with the same velocity, it is replaced by an image atom. In the future, the number of particles within the core box will remain constant. Nonetheless, a non-bonded cutoff is largely employed to manage the

non-bonded interactions such that each atom interacts with just one image of every other atom in the system.

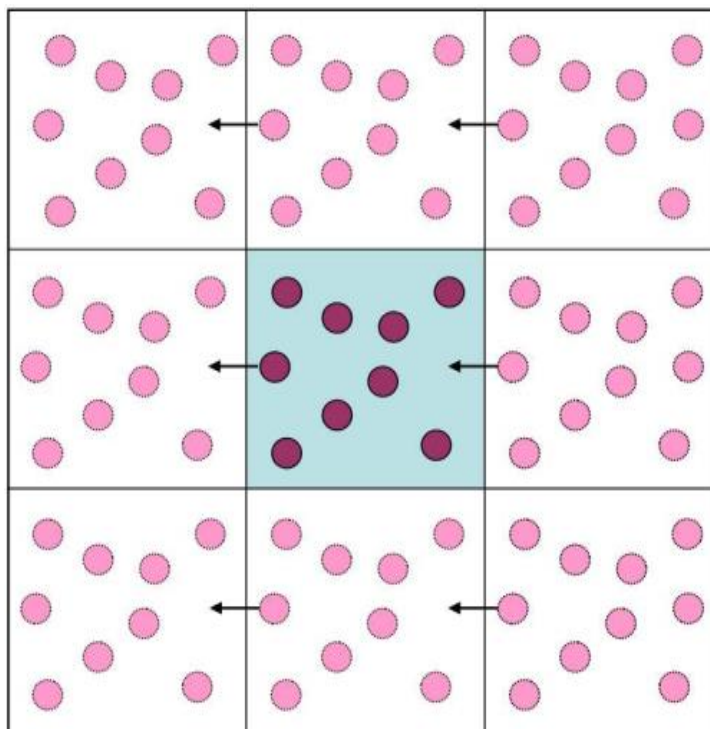


Figure 3.2. Periodic boundary conditions in two dimensions. The simulation cell (dark color) is surrounded by translated copies of itself (light color) (Taken from [481]).

3.1.1.7. Temperature and Pressure Computation and Control:

The initial temperature of the system is computed by coupling to a Berendsen thermal bath [477]. The bath supply or remove heat from the system as appropriate, thereby acts as a source of thermal energy. The system temperature $T(t)$ that deviates from the bath temperature T_0 is corrected according to:

$$\frac{dT(t)}{dt} = \frac{1}{\tau} \{T_0 - T(t)\} \dots\dots\dots(3.12)$$

where τ (time constant) determines the strength of the coupling between the bath and the system. The temperature of the system is corrected by scaling the atom velocities at each step by a factor χ , given by:

$$\chi = \left[1 + \frac{\Delta t}{\tau_T} \left(\frac{T_0}{T(t)} - 1 \right) \right] \dots\dots\dots(3.13)$$

The strength of the coupling can be varied by changing the time constant τ .

The pressure control method is similar to the temperature control method. The system can be coupled to a barostat, and the pressure can be maintained at a constant value by periodic scaling of the simulation cell size and atomic positions with a factor μ :

$$\mu = 1 - \omega \frac{\Delta t}{\tau_p} (P - P_0) \dots \dots \dots (3.14)$$

where ω represents the isothermal compressibility, τ_p represents the relaxation constant, P_0 is the pressure of the barostat, P , the momentary pressure at time t and Δt is the time of step. The standard simulation package AMBER14 is used in the present work [478, 479]. PMEMD, one of the AMBER modules, carries out the molecular dynamics simulation. The various steps involved in setting up and running an MD simulation are discussed below in detail and shown in the form of a schematic representation as depicted in the **Figure 3.3**.

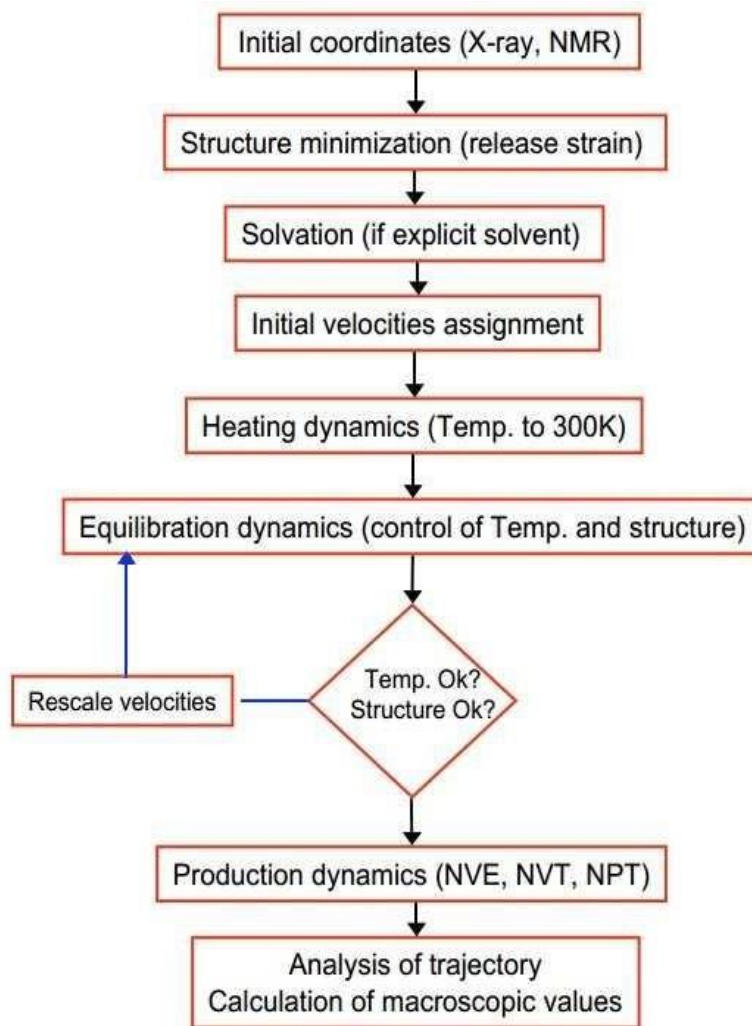


Figure 3.3. Schematic flowchart of steps involved in MD Simulation (Taken from [481]).

3.1.1.8. Water molecule models:

Research is greatly aided by computer simulations of biomolecular systems because they provide information about the structure, dynamics, and energetics of biomolecules that is not accessible to experimental measurement techniques. However, other molecular models are presented that give information for water in MD simulation. These models are described by site count, polarization effects, and model structure (rigid or flexible). The importance of water models is demonstrated by the fact that the (known but hypothetical) model (i.e., computer water) can precisely predict the physical properties of liquid water. This is due to the fact that it displays the liquid water's (unknown) structure. The system's computational sophistication, the size that can be computed in an acceptable length of time, and all three of these parameters are in the trade-off. The limits set by the size of the system, the time restrictions, and the complexity of the models are being tested even if computational power increases significantly year after year. 3-site water models are the ones that are most frequently employed in MD simulations due to their simplicity, thermodynamic explanations, computational efficiency, and logical structure.

A water molecule's three atoms can interact with these models through three different areas. The point charge of each atom is specific to that atom. Out of all the atoms, oxygen is the only one with Lennard-Jones characteristics that allow for interaction. The models composed of Lennard-Jones sites with orienting electrostatic effects may or may not cover one or more of the charged sites. To determine the molecular size, Lennard-Jones interactions are crucial. This contact is considered to be repulsive at very close distances, proving that electrostatic interactions keep the structure from completely collapsing. It is extremely attractive yet non-directional at intermediate distances and competes with directionally attractive electrostatic interactions.

A few well-known 3-site models include the simple point charge (SPC), extended simple point charge (SPC/E), and transferable intermolecular potential three-point (TIP3P) models [480]. However, each of these models employs a geometry that is consistent with the known form of the water molecule. In this simulation, the TIP3P water model is employed. The O-H bond length (r_{OH}) and H-O-H bond angle (θ_{HOH}) of the TIP3P water model used in this study are determined to be comparable to

experimental gas-phase values of 0.9572 Å and 104.52°, respectively. **Figure 3.4** depicts the structure of a simple TIP3P water model.

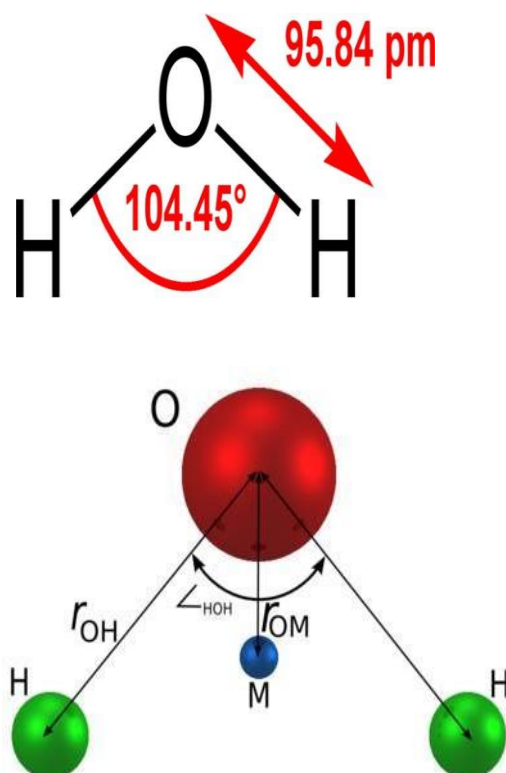


Figure 3.4. Schematic representation of TIP3P water model (Taken from [481]).

3.1.1.9. Molecular Dynamics Steps:

In order to propagate a molecular system using the above equations there are three typical stages:

- (a) Energy Minimization
- (b) Equilibration
- (c) Production Dynamics

(a) Energy Minimization:

To initiate dynamics, it is necessary to locate a stable point or minimum on the potential energy surface using the force field given to the system's atoms. On the surface of minimal potential energy, the net force on each atom vanishes. Constraints can be imposed during both minimization and dynamics. These constraints may be derived from

data, such as NOEs from an NMR experiment, or they may be imposed by a template to compel a ligand to locate the structure that is structurally closest to a target molecule. To minimize, a function (supplied by the force field) and an initial guess or set of coordinates are required. The magnitude of the first derivative can be utilized to identify the direction and magnitude of a step (i.e. change in coordinates) necessary to approach a minimum configuration. In addition to the magnitude of the first derivative, convergence can also be rigorously characterized by its magnitude.

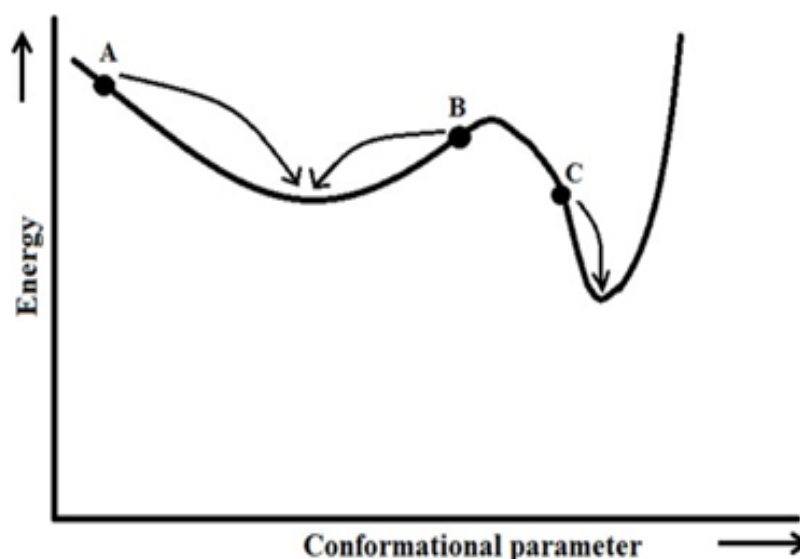


Figure 3.5. A schematic one-dimensional energy surface. Minimization methods move downhill to the nearest minimum (Taken from [481]).

The majority of minimization algorithms can travel downhill on the energy surface, allowing them to identify the minimum closest to the origin. Consequently, **Figure 3.5** depicts a schematic energy surface and the minima that would be reached by beginning from the three sites A, B, and C. To locate multiple minima or the global energy minimum, it is necessary to generate many initial locations, each of which is then minimized [481].

The convergence of minimal energy occurs when the derivatives are close to zero. Prior to initiating an MD simulation, it is crucial to perform energy minimization on the structure in order to eliminate poor connections that could otherwise result in structural distortion. There are three major minimization protocols: (i) Steepest descent, (ii) Conjugate gradient, and (iii) Newton-Raphson.

(i) **The Steepest Descents Method:** The steepest descents method uses the first derivative to determine which direction leads to the minimum. It moves parallel to the net force's direction. For $3N$ Cartesian coordinates this direction is depicted by a $3N$ -dimensional unit vector, namely \mathbf{s}_k . Thus:

$$\mathbf{s}_k = -\mathbf{g}_k/|\mathbf{g}_k| \dots\dots\dots (3.16)$$

Having defined the direction along which to move it is then necessary to decide how far to move along the gradient. Consider the two-dimensional energy surface of **Figure 3.6**. From the starting point, the direction of the gradient is along the line. If we imagine cutting through the surface along the line, we can see that the function will go through a minimum and then go up. We have the option of locating the minimal location via a line search or by taking arbitrary steps in the direction of the force [482].

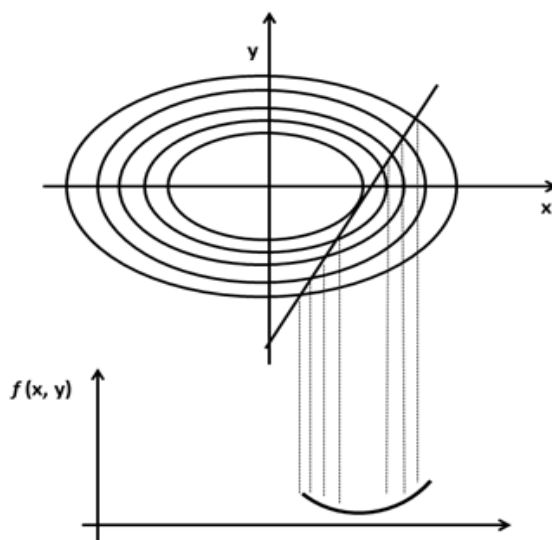


Figure 3.6. A line search is used to locate the minimum in the function in the direction of the gradient (Taken from [481]).

(ii) **Conjugate Gradients Minimization:** The conjugate technique generates a set of directions that does not exhibit the oscillating behaviour of the steepest descents method in confined valleys. In conjugate gradients, each point's gradient is orthogonal, but the directions are conjugate. A set of directions that are the same in both directions has the property that a quadratic function with M variables will reach its minimum in M steps. The conjugate gradients method moves in a direction \mathbf{v}_k from point \mathbf{x}_k where \mathbf{v}_k is computed from the gradient at the point and the previous direction vector \mathbf{v}_{k-1} [482].

(iii) Newton-Raphson Method: The Newton-Raphson approach makes use of both the first and second derivatives. In addition to employing gradient information, the curvature is used to anticipate where along the gradient of the function the direction will change. It is the strategy that necessitates the most processing resources in order to carry out energy minimization.

Before minimization, water molecules are added to the system if needed to make it more soluble. For solvation, a large box of water that has already been brought to the same temperature is used. The water box covers the whole system, and any water molecules that touch proteins are taken away. At this point, energy minimization should be done with the protein fixed in the position where it has the least amount of energy. This gives the water molecules a chance to adjust to the new shape of the protein molecule.

(b) Equilibration: The equations of motion for a group of atoms are solved by molecular dynamics. The solution to a molecule's equations of motion shows how its motions change over time. This is called the trajectory. Depending on the temperature at which a simulation is run, MD makes it possible to cross barriers and try out different arrangements. Before we can start MD, we must first assign velocities. The Maxwell-Boltzmann distribution is used to tell a random number generator how to do this. In the kinetic theory of gases, the average kinetic energy of the system is used to figure out the temperature. Equation 3.17 is the formula of the system's internal energy. Equation 3.18 is the formula for kinetic energy. By taking the average of the speeds of all the atoms in a system, you can figure out what the temperature is. Once an initial set of speeds has been made, it is assumed that the Maxwell-Boltzmann distribution will stay the same for the rest of the simulation.

$$U = (3/2) NkT \dots \dots \dots (3.17)$$

$$U = (1/2) Nmv^2 \dots \dots \dots (3.18)$$

After minimizing, we can think that the temperature is really zero Kelvin. To start dynamics, the system must be heated to the desired temperature. This is done by giving the particles their speeds at a low temperature and then using the equations of motion to figure out how they move. After a certain number of rounds of dynamics, the temperature goes up. Most of the time, scaling temperature is done by scaling speed. With a typical

time step of 1 fs, equilibration is run for at least 5 ps (5000 time steps) and often for 10 or 20 ps.

(c) **Production dynamics:** During the dynamics phase, thermodynamic averages are determined or new configurations are tested. The stage in which these applications are implemented is frequently referred to as production dynamics. During this phase, thermodynamic parameters can be computed. More than a hundred ps-ns can be used to create a production run.

3.1.2. Potential of Mean Force:

The potential of mean force (PMF) [482] is fundamental concepts regarding the changes in free energy as a function of inter or intra-molecular coordinates of molecular systems. The reaction coordinate may be the distance between two atoms or the torsion angle of a bond, therefore, its distribution function is intrinsically linked to that coordinate. When the system is in a solvent, the PMF contains both solvent effects and the intrinsic interaction between the two particles. From the transition state of the process, the rate constant may be deduced. There are numerous ways to compute the PMF. The simplest sort of PMF is defined as the free energy change with reaction coordinate as the change in separation (r) between two particles [481]. It is denoted as:

$$A(r) = -k_{BT} \ln g(r) + \text{constant} \dots \dots \dots (3.19)$$

Over the relevant range of the parameter r , the PMF can change by several multiples of k_{BT} . Because of the radial distribution function and the PMF's logarithmic connection, a relatively modest change in the free energy may equate to a shift in $g(r)$ of an order of magnitude from its most likely value. Unfortunately, the radial distribution function in areas where this difference is significant is not sufficiently sampled by the MD simulation method, resulting in erroneous PMF estimates. *Umbrella sampling* is one of the most popular sample methods used to prevent this issue (US).

3.1.2.1. Umbrella Sampling (US):

US solve the sampling issue by confining a system to a particular portion of its conformational space, so altering the potential function so that the unfavorable states are sampled correctly. The expression for the modification of the potential function is :

$$P'(r^N) = P(r^N) + W(r^N) \dots\dots\dots(3.20)$$

Where $W(r^N)$ is a weighting function, which takes a quadratic form:

$$W(r^N) = k_W (r^N - r_0^N)^2 \dots\dots\dots (3.21)$$

For those configurations that are distant from equilibrium state r_0^N the weighting function shall be large, hence a simulation by using the modified energy function $P(r^N)$ will be biased away from the configuration r_0^N , along with some relevant ‘reaction coordinate’ (RC). The resulting distribution will, of course, be non-Boltzmann. Torrie and Valleau [483] introduced a method for extracting the corresponding Boltzmann averages from non-Boltzmann distributions. The result is:

$$\langle A \rangle = \frac{\langle A(r^N) \exp [+ W \frac{r^N}{k_B T}] \rangle_W}{\langle \exp [+ \frac{W(r^N)}{k_B T}] \rangle_W} \dots\dots\dots (3.22)$$

The subscript W denotes that the mean is based on the probability $PW(r^N)$, that in turn is calculated by the modified energy function $P(r^N)$. Most of the time, an umbrella sampling calculation is done in stages. Each stage has a certain value for the coordinate and a certain value for the forcing potential $W(r^N)$. But if the forcing potential is very large, then the denominator in Equation 3.22 is dominated by contributions from only a few configurations with especially high values of $\exp [W(r^N)]$ and the average takes too long to converge.

3.1.2.2. Running the umbrella sampling calculations:

With a relaxed starting structure one can run MD on the individual umbrella windows. The key point to remember when selecting the number of windows is that the end points must overlap, i.e. window 1 must sample some of window 2 etc. The force constant similarly has to be big enough to ensure that the subset of phase space are sampled but not too strong that the windows become too narrow and can’t overlap.



Figure 3.7. Working principle of Umbrella Sampling. Taken from [483].

"\" = lower bound linear response region

"/" = lower bound linear response region

"..." = parabola

"_" = flat region

Normally one can vary the size of the windows and the constraints as a function of position along the pathway. The amount of simulation we do in each window needs to be such we can converge our sampling. To specify the harmonic restraint a reference file is employed where R1, R2, R3, R4 define a flat-welled parabola which becomes linear beyond a specified distance. Essentially between r1 and r2 it will be harmonic with force constant rk2, between r2 and r3 it will be flat and between r3 and r4 it will be harmonic with force constant rk3.

3.1.2.3. The Weighted Histogram Analysis Method (WHAM) for free-energy calculations:

The WHAM method [484] is an extension of the regular US method, however it has some advantages over it. In addition to improving the linkages between simulations, the WHAM technique permits several overlaps of probability distributions in order to achieve more accurate estimations of free-energy differences. If three or more distributions are involved in the overlap zone, the previous method of producing a single distribution function by requiring that the probability distributions coincide at some point in the overlap region will fail to produce unique free energies. This approach incorporates an error estimation that gives scientists with objective estimations of the ideal location and duration of subsequent simulations to improve the precision of their results. The

WHAM technique takes into account all simulations that generate distributions that overlap. The WHAM approach optimally connects the many simulations through their overlapping histograms. Additionally, the WHAM equations can be utilised to create PMFs and free energies as a function of the coupling parameter(s) and/or temperature. This is advantageous because simulations can be conducted at a range of temperatures to optimise conformational sampling, and the results can be extrapolated (or interpolated) to the target temperature [487].

3.1.3. The molecular mechanics energies combined with the Poisson Boltzmann or generalized Born and surface area continuum solvation method (MM-PBSA and MM-GBSA):

By utilizing the Poisson-Boltzmann (PB) and Generalized Born (GB) models, *Kollman et al.*[485] developed the MM-PBSA and MM-GBSA methods [486-495] to calculate the absolute binding free energy for the association of two non-covalent molecules, A and B, in solution.



In this equation, $[A]_{aq}$ stands for molecule A's dynamical structure when it is in solution, $[B]_{aq}$ for molecule B's dynamical structure when it is in solution, and $[A^* B^*]_{aq^*}$ for the complex formed from molecule A and molecule B. The binding free energy for the noncovalent interaction between two molecules can be expressed in terms of various thermodynamic quantities, including:

$$\Delta G = \Delta H - T\Delta S \dots \dots \dots (3.24)$$

The parameters ΔH represent enthalpy, ΔS represent entropy, and T represent the temperature of the system at a temperature of 300 K. For the receptor-ligand complex, the binding free energy is computed as:

$$\Delta G_{\text{bind}} = G_{\text{com}} - [G_{\text{rec}} + G_{\text{lig}}] \dots \dots \dots (3.25)$$

The parameter G_{com} denotes the absolute free energy of the complex, G_{rec} , the absolute free energy of the receptor, and G_{lig} , the absolute free energy of the ligand. The enthalpy term shown in *equation 3.27* can be divided into sub-energy terms such as:

$$H_{\text{tot}} = H_{\text{gas}} + G_{\text{solv}} \dots \dots \dots (3.26)$$

$$H_{\text{gas}} = E_{\text{el}} + E_{\text{vdw}} + E_{\text{int}} \dots \dots \dots (3.27)$$

The parameters, such as H_{gas} describe the potential energy of the solute, that is denoted as the sum of van der Waals (E_{vdw}), electrostatic (E_{el}) and internal energies (E_{int}) in the gas phase. G_{solv} is the sum of the electrostatic (G_{el}) and non-electrostatic (hydrophobic) contributions (G_{nonel}) to the solvation-free energy needed to move a solute from a vacuum into a solvent. This is shown in *equation 3.26*:

$$G_{\text{solv}} = G_{\text{el}} + G_{\text{nonel}} \dots \dots \dots (3.28)$$

The total entropy, S_{tot} arose from changes in the degree of freedom as shown in *equation 3.27*:

$$S_{\text{tot}} = S_{\text{trans}} + S_{\text{rot}} + S_{\text{vib}} \dots \dots \dots (3.29)$$

The parameters in *equation 3.28*, specify the translational (S_{trans}), rotational (S_{rot}), and vibrational (S_{vib}) entropies of each species. The representation of the energy terms are shown in **Figure 3.8**. The binding free energy (ΔG) has the following form for all the absolute energy terms:

$$\Delta G_{\text{binding}} = [\Delta H_{\text{gas}} + \Delta G_{\text{solv}}] - T\Delta S_{\text{tot}} \dots \dots \dots (3.30)$$

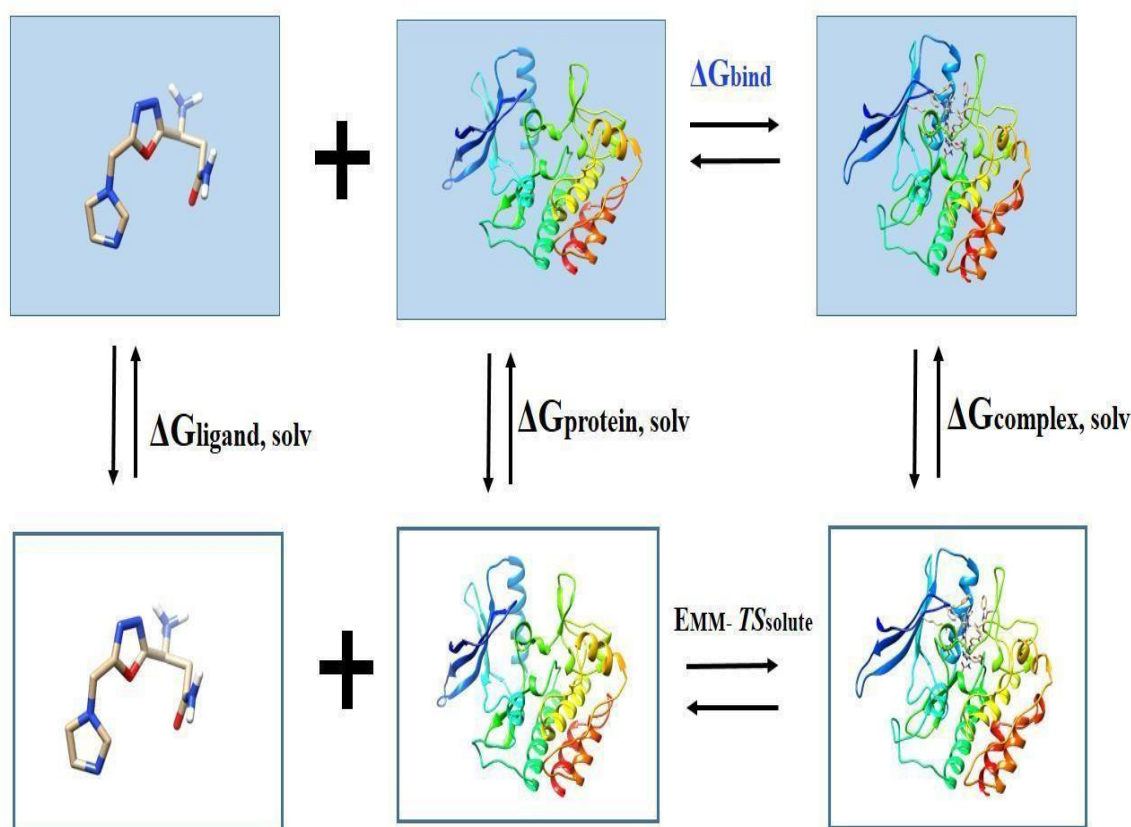


Figure 3.8. Computational schemes of the binding free energies based on MM-PBSA/GBSA. The free energies colored in black are directly calculated; while the free energy of interest colored in blue is indirectly did using the thermodynamic cycle of other free energies (Modified from [496]).

In the theoretical framework of MM-GBSA and MM-PBSA, the free energy can be divided into various components, including the gas-phase potential energy, the polar and non-polar solvation free energies, and the entropy upon ligand-receptor interactions. These components can be independently calculated based on the conformations extracted from the generated MD trajectories. Molecular mechanics can calculate the potential energy based on the various force fields (MM). The polar solvation free energy can be estimated using either the Poisson-Boltzmann equation or the Generalized-Born (GB) model. The entropy is typically calculated by normal mode analysis (NMA). Due to their acceptable accuracy, relatively low computational cost, and broad applicability, such as for small ligand-protein systems, protein-protein systems, and protein-RNA/DNA systems that represent nearly the entire interaction-omics of life science, MM-GBSA and MM-PBSA may be the most popular methods for calculating the large-scale binding free energy. The change in entropy of protein-ligand interaction is, however, often ignored by applications that use MM-GBSA and MM-PBSA due to the NMA's relatively poor

prediction accuracy and prohibitively high computational cost. Instead of using a huge number of MD snapshots, the computation for MM-PBSA might be based on a single optimized structure. This will save a significant amount of computational work and disregard dynamical effects, making the outcomes very dependent on the initial structure and obliterating all information regarding the statistical accuracy of the procedure. Even though several researches have emphasized the value of MD sampling, reduced structures yield better results from MD simulations. This method was tried using several MD snapshots and started with minimization. It produced results that were comparable to MD's, albeit certain irrational structures required to be removed. However, minimization in a GB continuum solvent can save even more time. By aggregating interactions between a certain residue and every other residue in the system, the PRED [497] determines the energy contribution of that residue. The computations for the MM-PBSA/GBSA took all trajectories into account.

Using the PB and GB models, the electrostatic solvation energy was calculated. The inside (solute) and exterior (water) have dielectric constants of 1 and 80, respectively. The solvent's probe radius was adjusted to 1.4. With the solute dielectric constant set to 1, the exterior dielectric constant was altered to 80. The polar component of the solvation-free energy (ΔG_{GB}) was calculated using the modified Generalized Born (GB) model [498], and the solvent accessible surface-area (SASA) was measured using the LCPO technique [499]. Multiple automated variants of the MM-PBSA binding free energy approach, which was initially created for the AMBER software, have been introduced. For the PB method, γ is 0.00542 kcal (mol⁻¹ Å⁻²) and $b = 0.92$ kcal mol⁻¹, while for the GB method, $\gamma = 0.0072$ kcal (mol⁻¹ Å⁻²) and $b = 0$ kcal mol⁻¹.

3.1.4. PDBsum:

The PDBsum is a web-based database [500, 501] established in 1995 that provides information regarding all empirically proven structural models as published by the Protein Data Bank [502, 503]. PDBsum server also provides a schematic depiction of the inter-molecular interactions as a LIGPLOT diagram [504]. The LIGPLOT program automatically generates schematic 2-D representations of protein-ligand complexes from standard Protein Data Bank file input. The primary objective of this server is to portray the structural information for each 3-D model as graphically as feasible. Consequently, it gives graphic diagrams of the molecules constituting each PDB entry, i.e.

protein/DNA/RNA chains, ligands, and metals, as well as of their interactions. Numerous new and evolving features have been added over the course of time. In addition to the literature references and other links to databases, it is possible to laboriously gather such information for a single individual, although it would be best to present it immediately. The functional annotations of this server are however present. For the relevant UniProt sequences [505], data from the Gene Ontology annotations is provided, which is accessible as a functional annotation in the UniProt Knowledgebase [505]. Given is a reaction diagram in which any products or reactants for enzymes that are identical to any ligand bound in the given structure are underlined. However, while the information is useful, it may not convey the structure's greater significance. The scientific literature is a more comprehensive source of knowledge, as it contains the original paper authored by the structure's authors. The authors explain how the 3-D structure connects to or explains the biological activity of the molecule(s) under consideration. This information can be retrieved by following a link to the article, however not everyone has access to the articles.

This server was the first webserver to take advantage of the new World Wide Web technology by producing a PDB structural information catalogue. Its primary objective was to provide a broad visual compendium of the PDB's proteins and their complexes. It was originally created at University College London (UCL) in 1995. These pictures are comprised of numerous structural studies that are not provided or readily available elsewhere.

Before being transferred to the European Bioinformatics Institute (EBI) in 2001, the PDBsum server was built at UCL. Since then, enhancements and additions have been implemented concurrently with other servers. However, two of the WWPDB12 members—the consortium that now operates the PDB archives are the most powerful. The Research Collaboratory for Structural Bioinformatics (RCSB) with its server located at <http://www.rcsb.org>, and PDB Europe (PDBe) with its server located at <https://www.ebi.ac.uk/pdbe> [502]. Both sites offer exhaustive and extensive coverage of all PDB entries in addition to potent structural analysis facilities.

3.1.5. Molecular Docking:

Computerized prediction of protein-protein and protein-small molecule interactions is one of the most difficult tasks in structural biology. Numerous biological studies, both in academia and industry, can benefit from accurate and dependable interaction prediction. In protein-protein docking, the issue is to precisely connect two interacting molecules. The prediction is based on the interactions between residues involved in the target interaction. Several docking methods have been developed [506-510]. However, there are now just a handful of free algorithms available online. The search method and evaluation of resolved complexes in the six-dimensional transformation space account for the majority of variations between the algorithms.

Molecular docking is used to model the interaction between a protein and a small molecule or between two proteins at the atomic level [510] in order to comprehend the behaviour of small molecules at the target protein's binding site or to obtain the interacting interface residues participating in protein-protein interactions. Two phases comprise the docking technique. In the first stage, the position of the ligand at the binding site is determined. In the second stage, conformers of ligands are ranked using a score formula based on binding affinity. The scoring function must be able to rank the experimental binding mode as the best among all created conformations once it has been replicated by sampling methods. PatchDock server [511] was utilized for protein-small molecule docking, whereas ClusPro server [512] was utilized for protein-protein docking.

(i) PatchDock: Using the online docking tool PatchDock, stiff docking of molecules, such as protein-protein or protein-drug interactions, is carried out while taking into account surface alterations during intermolecular penetration [511]. Its foundation is the geometry molecular docking method. Additionally, it looks for docking modifications that significantly compliment each other's molecular morphologies. These docking alterations lead to both minor steric conflicts and substantial interface regions when they are put into practice. It is established that a wide interface region contains a number of local traits that coincide with those of the molecules that are linked and have complementary features. With this technique, concave, convex, and flat patches are distinguished in the Connolly dot surface representation of molecules. The next step is to match complementary patches to produce candidate modifications. An additional grading method that takes into account both geometric compatibility and atomic desolvation

energy is used to assess each proposed change. Then, duplicate candidate solutions are omitted using the RMSD (root mean square deviation) clustering algorithm.

The high efficiency of the PatchDock is largely owing to its rapid transformational search, which is accelerated by local feature matching as opposed to brute-force scanning of the six-dimensional transformation space. Utilizing complex data structures and spatial pattern recognition techniques, such as geometric hashing and posture clustering, established in the field of computer vision can further accelerate the computational processing time. On a 1.0 GHz PC processor running Linux, the PatchDock runtime for two protein inputs of normal size (about 300 amino acids) is less than or equal to 10 minutes. This technique's fundamental characteristic is based on the Kuntz algorithm for local form feature matching [496]. The correct conformation is maintained via the docking method, which identifies the higher likelihood molecule surface regions present at the binding site. This technique docks large proteins with small drug molecules by manipulating receptors and ligands of varying sizes.

(ii) ClusPro: The web-based server ClusPro [512] was first made available in 2004. It has undergone considerable enlargement and modification since then. Direct docking of two interacting proteins is possible with ClusPro. To perform docking, the server needs two protein files in PDB format. When docking, the server does the following three computations:

- The sampling of billions of conformations using rigid body docking.
- 1000 structures with the lowest energy were clustered based on root-mean-square deviation to determine the largest clusters that reflect the most credible models of the complex (RMSD).
- Improving the selected buildings by minimizing energy use. PIPER [513], a docking tool, makes advantage of the Fast Fourier Transform (FFT) correlation approach during the rigid body docking stage.

The ClusPro web server has since been upgraded to ClusPro 2.0.

3.1.6. *In silico* prediction of protein-protein interaction:

Protein-protein interactions (PPIs) are crucial to many cellular physiological activities and many diseases [514-516]. Since protein-protein interactions differ, the protein interface must be carefully examined. Protein-protein interactions depend on stability and specificity depends on protein contact size. The interface between two proteins typically has a 1500–3000 Å² surface area submerged in each protein [517-519]. Proteins with significant shape complementarity [520-522] and hydrophobic effects [523] from van der Waals interactions between nonpolar protein residues form protein-protein interaction sites. Electrostatic complementarity between the two proteins' interacting protein surfaces promotes the growth and stability of the complex. At some interfaces, hydrogen bonds and electrostatic interactions play a significant role in determining how one protein docks with the binding site of another. Protein-protein interaction prediction is crucial for developing new therapeutics. Protein interaction is essential for many biological processes, both good and harmful, although it can be impeded by external chemicals. The modern drug discovery process consists of two main steps: selecting a possible pharmacological target, learning more about it, and developing an appropriate ligand [524]. The development of modulators that selectively target protein complexes can therefore benefit from an understanding of protein-protein interactions.

When a protein-ligand complex in the PDB format is submitted, the LigPlot⁺ program [525] generates a 2-D graphic depiction of the hydrogen bonds as well as non-bonded interactions between the protein residues with which the ligand interacts (**Figure 3.9**). To create the protein-ligand interaction profile, the LigPlot⁺ program also offers a standalone version called LigPlot⁺ that can be downloaded and installed. The outcome is a PostScript (PS) file in colour or black and white that displays intermolecular interactions as well as their intensities, including hydrophobic interactions, hydrogen bonds, and atom accessibility. For every ligand, the software is completely universal. For the prediction of residue interactions in nucleic acids, there are additional dedicated servers [526-528].

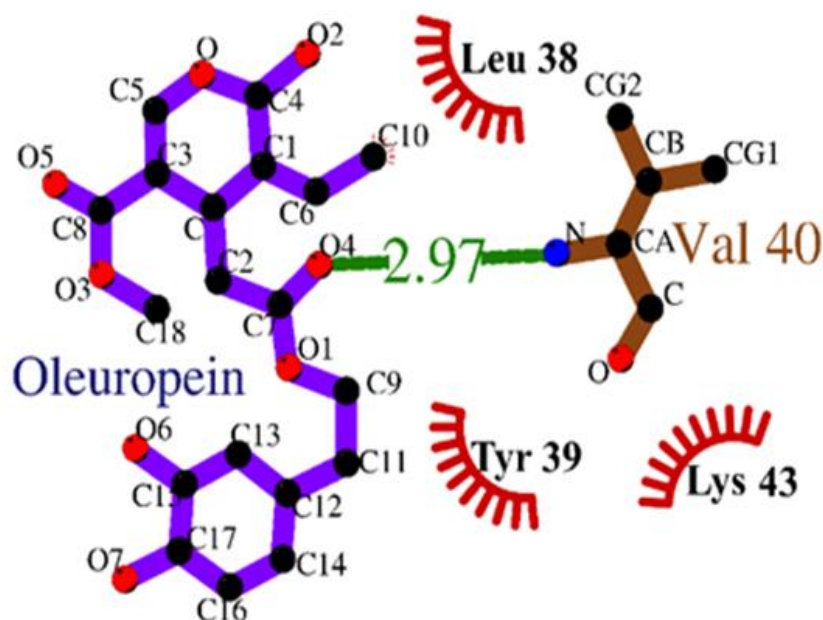


Figure 3.9. Ligplot⁺ analysis showing the interaction of hydrophobic residues of α -Synuclein with OleA [Taken from Chapter 3].

3.1.7. 3-D structure visualization tools:

(i) **Visual molecular dynamics (VMD):** VMD is an application for modelling and visualizing molecules [529]. VMD's primary function is to visualize and assess the results of MD simulations. In addition, it supports the manipulation of arbitrary graphical objects, sequence data, and volumetric data.

(ii) **UCSF Chimera:** UCSF Chimera is a highly adaptable tool for interactively viewing and analyzing molecular structures and associated data, such as conformational ensembles, density maps, sequence alignments, supra-molecular assemblies, and docking results [530]. Chimera was created by the Resource for Biocomputing, Visualization, and Informatics (RBVI) with assistance from the National Institutes of Health (NIH).

(iii) **ArgusLab:** ArgusLab is a Windows-based molecular modelling, graphics, and drug design application. Mark Thompson, a research scientist with the Department of Energy at Pacific Northwest National Laboratory, develops ArgusLab [531].

3.1.8. Analysis of MD trajectories:

(i) Root Mean Square Deviation (RMSD):

The deviation of a structure with respect to a particular conformation is measured by RMSD. It is defined as:

$$\text{RMSD} = \left(\frac{\sum_N (R_i - R_i^0)^2}{N} \right)^{1/2} \dots \dots \dots (3.31)$$

where N is the total number of atoms/residues considered in the calculation, and R_i is for the vector position of particle i (target atom) in the snapshot, R_i^0 is the coordinate vector for reference atom i . The RMSD was estimated utilizing backbone atoms and the initial frame of the simulation as the reference. The value of N in the equation 3.31 represents the total number of variables required to calculate the RMSD, which is the product of the number of locations (i) the number of strands and (j) the number of angular parameters. The estimated RMSD is a radial vector of length r in the structure space specified by the RMSD absolute magnitude. The radial problem is based on the fact that the wider the radius, the more configurational space volume exists between a given r and $r+dr$. At a greater r , the same RMSD value could capture both comparable and dissimilar structures. Additionally, crucial information connected to the comparison may be compromised. Methods that rely on comparatively reduced RMSD values provide a more accurate measurement of difference. Due to the inherent flexibility of the molecule, the presence of two or more structural substates presents a second significant issue with the application of RMSD. To reliably describe the dynamical characteristics, however, without losing information, a method is necessary.

(ii) Root Mean Square Fluctuation (RMSF):

Root Mean Square Fluctuation (RMSF) defines the measure of deviation between the particle position i and some reference position:

$$\text{RMSF} = \left(\frac{1}{T} \sum_{t=1}^T (r_i(t) - r_i^{ref})^2 \right)^{1/2} \dots \dots \dots (3.32)$$

In the equation 3.32, T is defined as the time over which one wants to average and $\mathbf{r}_i^{\text{ref}}$ as the reference position of particle \mathbf{i} . The reference position will be the time-averaged position of the same particle i , i.e. $r_i^{\text{ref}} = r_i$.

In molecular dynamics (MD) simulations, RMSD (root mean square deviation) and RMSF (root mean square fluctuation) are used to measure the spatial variations of biomolecules. RMSD is the difference between two structures for a particular set of atoms, while RMSF is the fluctuation around an average, per atom/residue, across a set of structures (e.g. from a trajectory). It is absolutely feasible to have RMSD=0 with a non-zero RMSF for each atom, or a big RMSD with a very small RMSF, if there has been a substantial conformational shift followed by modest fluctuations in atomic locations.

(iii) Radius of Gyration (R_g):

In order to measure the compactness of the structure, the radius of gyration is calculated:

$$R_g = \left(\frac{\sum_i |r_i|^2 m_i}{\sum_i m_i} \right)^{1/2} \dots \dots \dots (3.33)$$

In equation 3.33, m_i is the mass of atom \mathbf{i} and r_i the position of atom \mathbf{i} with respect to the center of mass of the molecule.

(iv) Secondary Structure Analysis:

Kabsch and Sander (1983) [532] created a method called Dictionary of Secondary Structure of Proteins (DSSP) to compute the solvent accessibility of residues and set up a database of the Accessible Surface Area (ASA) for the majority of the proteins stored in the PDB. This is a typical program used to generate ASA values for prediction algorithms. It can be found at <http://www.cmbi.kun.nl/gv/dssp>. It functions by primarily categorizing protein secondary structures based on H-bonds to the backbone. In addition, it provides the $C\alpha$ -pseudo dihedral angles and $C\alpha$ -pseudo bond angles. It is distinguished by the detection of hydrogen bonds according to an electrostatic criterion. Consequently, elements of the secondary structure are assigned based on the distinctive hydrogen-bond patterns. As a measurement for the assignment of secondary structures, this method is widely recognized. DSSP is utilized by a number of software applications when assigning secondary structures. Rasmol, for instance, is a popular visualization tool that

assigns repeated structures using a DSSP-like method. Repeating H-bond patterns of the same type lead to the categorization of helix or strand, whereas non-repetitive H-bonds lead to the classification of β -bridges. The relative orientations of the backbone oxygen and nitrogen atoms are reflected in the respective (ϕ , ψ) backbone torsion angles, which cluster residues belonging to the same secondary structure type reasonably well in a Ramachandran plot. The DSSP provides information on two levels regarding the secondary structure of proteins. The C-pseudo bond angle is the angle between the vectors $C_{\alpha}(i) - C_{\alpha}(i-2)$ and $C_{\alpha}(i) - C_{\alpha}(i+2)$ for every residue. If this angle is less than 110° , corresponding to a substantially bent geometry without typical backbone H-bonds for residues not assigned to a helix, strand, or turn, then the summary class S of bends is utilized. If none of the above conditions are met, DSSP creates a space, which we label with the letter "C" for improved discrimination. However, these residues are located in a rather straight section of the protein backbone and do not form secondary structure-relevant Hydrogen bonds.