

CHAPTER 3

Materials and Methods

Materials and Methods

3.1. Tools and Techniques:

To study the structural dynamics as well as interaction profile of p53-MDM2 complex at different sites of interaction, as well as p53 and MDM2 molecules bound to the small molecule inhibitors, we used MD simulation. The principle and theory of MD simulation is discussed below.

3.1.1. Molecular dynamics (MD) simulation:

A method for creating dynamical trajectories for systems with N particles is called molecular dynamics (MD) simulation. This method involves integrating the Newton's equations of motion. For the MD simulation to proceed, a set of initial conditions is needed, including the positions and velocities of each particle, as well as a good model that can represent different forces acting between the particles and derived either from electronic structure calculations or through the use of empirical force fields. Finally, boundary conditions must be defined in order to be used. However, statistical mechanics computational techniques are characterized by the MD simulation method. For complicated systems that cannot be estimated theoretically, this method, as a complement to experimental methods, estimates the dynamic as well as equilibrium properties. But by creating a fascinating interface between theory and experiment, it occupies a prestigious position at the intersection of biology, mathematics, chemistry, physics, and computer science. However, the static perspective of the biomolecules that is given by x-ray crystallography is insufficient to comprehend a very broad spectrum of biological activity. Nevertheless, it merely provides a generalized, frozen view of the complicated system. The atoms that make up molecules are thought of as living things that can communicate with one another and with their surroundings. However, the dynamic mobility that exists inside the molecules may show the system's greater variety of thermally accessible states. As a result, it is able to link the sequence to the structure as well as the function. Understanding the dynamics of molecular systems in both space and time allows us to gather a lot of data on their structural and dynamic features.

Although Frauenfelder and Wolynes' first description of the relationship between motion and protein function was novel, it is now widely accepted as being evident.

3.1.1.1. History of Simulation:

The molecular dynamics simulation approach was initially developed in the late 1950s by Alder and Wainwright to study the interactions of hard spheres. [389, 390]. Their research has produced a number of interesting discoveries on the behaviour of basic liquids. When Rahman performed the first simulation in 1964 using a plausible potential for liquid argon [391], it was the next significant advancement for the MD simulation. Rahman and Stillinger carried out the first MD simulation for a realistic system in their 1974 simulation of liquid water. [392]. The first protein simulations and the simulation of the bovine pancreatic trypsin inhibitor (BPTI) [393] were both published in 1977. The literature of today contains MD simulations of solvated proteins, lipid systems, and protein-DNA complexes that deal with a number of difficulties, including as the thermodynamics of ligand binding and the folding of tiny proteins. With the development of combined quantum mechanical-classical simulations, which are crucial for understanding enzyme processes within the context of the entire protein, the scope of simulation approaches has substantially increased. The majority of experimental approaches, including as x-ray crystallography as well as NMR structure determination, utilize MD simulation techniques.

3.1.1.2. Theory of molecular dynamics simulation:

The Newton's second law of motion, also known as the equation of motion, states that $F=ma$, where F is the force acting on the particle, m is its mass, and a is its acceleration. This equation of motion is the foundation of the MD simulation approach. However, if the force acting on each atom is known, it is likely possible to calculate the acceleration for every atom in the system. A trajectory is produced by integrating the motion equations, which provides information on the locations, velocities, as well as accelerations of particles with respect to time. It is possible to deduce the characteristics of the average values from these trajectories. This approach is deterministic, meaning

that once the velocities and locations of every single atom have been determined, the system state may be predicted in the past, present, or future. However, MD simulation techniques could be time- and resource-intensive. However, computers are becoming more affordable and faster. The simulations for solvated proteins are computed down to the millisecond level in time. But other studies have also reported on simulations in the millisecond time frame.

Newton's *equation of motion* is given by:

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

$$\vec{F} = m a \dots \dots \dots (3.2)$$

$$F = -\frac{d}{dr} \mu \dots \dots \dots (3.3)$$

In this equation, F_i defines the force exerted on particle i , m_i as the mass of particle i and a_i as the acceleration of particle i , which have been derived from the potential energy $\mu(r^N)$, where $r^N = (r_1, r_2 \dots r_N)$ denotes the entire set of $3N$ atomic coordinates.

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

$$F_i = -\nabla_i V \dots \dots \dots (3.4)$$

By combining the two equations give,

$$-\frac{dV}{dr_i} = m_i \frac{d^2 r_i}{dt^2} \dots \dots \dots (3.5)$$

Here V is defined as the potential energy of the system. The potential energy derivative used to account for position changes with respect to time can be connected to the Newton's equation of motion.

Finding a formula that may specify position $r_i(t+\Delta t)$ at time $t+\Delta t$ in connection to the locations that are previously known at time t is the major goal of the numerical integration for Newton's equation of motion. However, in order to determine the new locations at time $t+\Delta t$, the *Velocity Verlet* method uses both the positions as well as accelerations at time t as well as positions from time $t-\Delta t$. The explicit velocities are not used by the *Velocity Verlet* algorithm. The following are some reasons why employing the Verlet algorithm is important: In addition to being simple, the storage required is small. But this algorithm's mediocre precision is by far its worst drawback.

The *leap-frog algorithm*, which modifies the *Velocity Verlet* algorithm by calculating the velocities from the positions or directly propagating them, is one such version. In the *leap-frog approach*, the velocities are computed at time $t+1/2\Delta t$ and then used to compute the locations, r , at time $t+\Delta t$. The velocities and locations thus pass over one another. The key benefit of employing the *leap-frog algorithm* is the explicit calculation of velocities here; on the other hand, the main drawback is that velocities are not calculated concurrently with locations.

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

$$\dot{r}_i \left(t + \frac{\Delta t}{2} \right) = \dot{r}_i \left(t - \frac{\Delta t}{2} \right) + \ddot{r}_i(t)\Delta t \dots\dots\dots (3.6)$$

At the time t , the velocities can be computed from:

$$\dot{r}_i(t) = \frac{\dot{r}_i(t+\frac{\Delta t}{2})+\dot{r}_i(t-\frac{\Delta t}{2})}{2} \dots\dots\dots (3.7)$$

However, this becomes important when kinetic energy is needed at time t , such as when velocity rescaling is necessary. The atomic positions needed are then attained from:

$$r_i(t+\Delta t) = r_i(t) + \dot{r}_i \left(t + \frac{\Delta t}{2} \right) \Delta t \dots\dots\dots (3.8)$$

Force field is used to show the evolution of time for bond lengths, bond angles, and torsion as well as for non-bonding van der Waals as well as electrostatic interactions between atoms. A set of related constants and equations make up this force field, which is intended to mimic the molecular geometry and some of the tested structures' specific characteristics.

3.1.1.3. Force field (FF):

In mathematics, a force field is an expression that expresses how a system's energy depends on the positions of its constituent particles. It consists of the analytical interatomic potential energy form $U(r1, r2... rN)$, and a number of parameters that go into it. These parameters are typically obtained through semi-empirical quantum mechanical computations, from ab initio calculations, or by fitting to experimental data,

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

Bond lengths, angles, and rotations, as well as non-bonded interactions like van der Waals and electrostatic interactions, make up the majority of the potential energy function. **Figure 3.1** provides a graphical representation of the various interaction types.

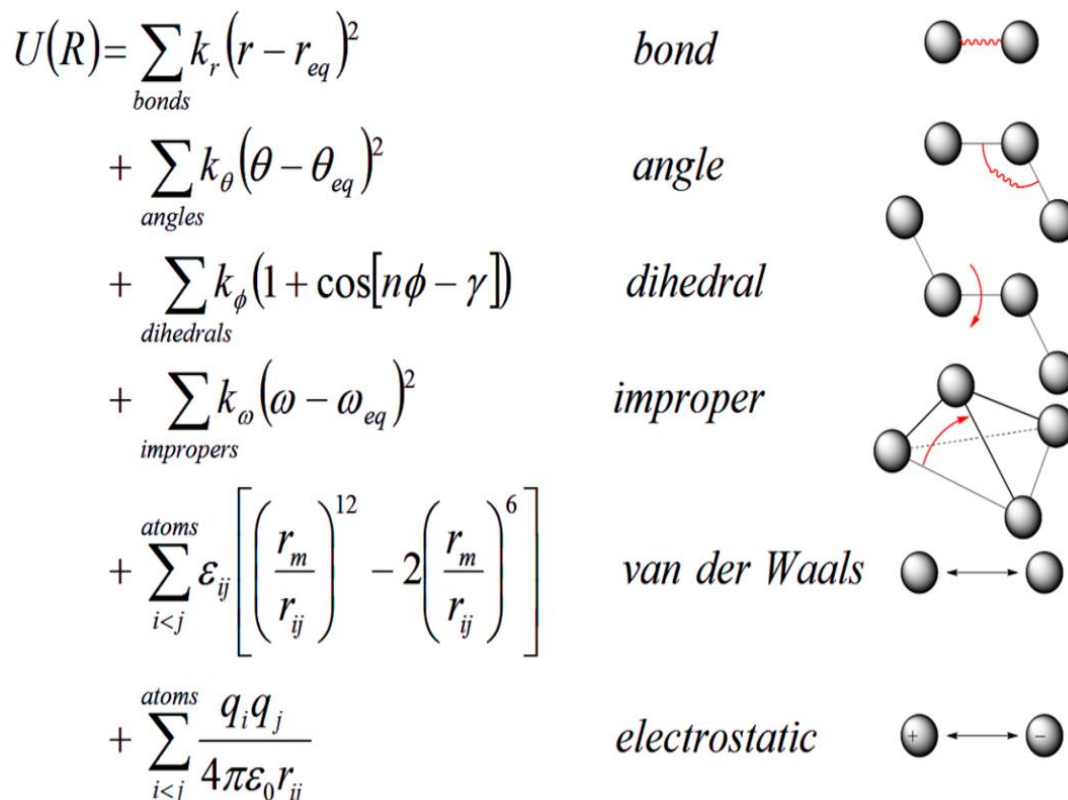


Figure 3.1. Schematic illustration of the main contribution to the potential energy function.

The first four terms in the equation signify the local or intramolecular contributions to the overall energy (*bond stretching, angle bending, and dihedral and improper torsions*). The repellent van der Waals interactions and the Coulombic contacts are described in the equation's final two terms, which in this case use a 12-6 Lennard-Jones potential.

3.1.1.4. Long range interactions: Ewald sum:

In computer simulations of condensed-matter systems, the electrostatic interactions are typically estimated using the Ewald Summation approach [394]. In the construction of the Ewald Sum, mistakes caused by truncating the infinite real- and Fourier-space lattice sums are analysed. An ideal selection with a screening parameter of 7 is retrieved for the Fourier-space cutoff. However, it is evident that a certain precision scales with $7/3$ regardless of the number of vectors contained in the Fourier space. However, the quality of the Ewald-sum implementations may be evaluated using the proposed method, which can also be used to compare different implementations, by controlling the effective computational parameters for Ewald sums. This technique is most likely the one that is used the most frequently to assess the long-range interactions in MD simulations. The fundamental idea underlying the Ewald sum is taking into account a charge distribution for opposing sign on each site. The interactions between the surrounding atoms are visible in the additional charge distribution. Although the interactions are short-range, the cut-off scheme can effectively handle them. The same charge distribution with the opposite sign and short-range interaction is added up in the reciprocal space in order to compensate for the excess charge distribution. Because the electrostatic potential caused by the screened charge is a quickly declining function of \mathbf{r} , it is simple to determine the input for the electrostatic potential at a certain point \mathbf{r}_i caused by a group of screened charges. The total potential energy for the long range Coulomb interaction is given by the expression:

$$\mu_c = \mu_q(\alpha) - \mu_{self}(\alpha) + \Delta\mu(\alpha) \dots \dots \dots (3.10)$$

Larger values of α in the equation result in sharper distributions, and for such large numbers, K summations are added for improved accuracy. However, a higher value for α narrows the filtered potential range, allowing us to employ a smaller cutoff radius. In order to provide a greater efficiency and accuracy, the value of α is therefore susceptible to optimization between the two aspects. In the aforementioned scales, only N^2 is used to represent the Ewald summation. Nevertheless, Finchman was successful in optimising the summing that scales as $N^{3/2}$ by selecting the appropriate values for α and K for the k-

space summation cut-off. In addition, this Ewald summation approach can be improved by assessing the reciprocal summation using the Fast Fourier Transform (FFT). However, the particle mesh-based approach focuses on using a fixed cutoff for direct space sum and an approximate reciprocal space sum based on FFT that grows as $N \log(N)$.

Under periodic boundary condition (PBC), this approach effectively calculates the endless range Coulomb interaction. Additionally, Particle Mesh Ewald (PME) is a variation that uses the three dimensional fast Fourier transform (3DFFT) to speed up the Ewald reciprocal sum to almost linear scaling. Due to the unlimited range of the coulombic interaction, under PBC particle i inside the unit cell interacts electrostatically with every other particle j inside the cell as well as with every periodic image of j , as well as with each of its own periodic images. Equation 3.11 provides the total Coulomb energy of a system made up of N particles inside a cubic box of size L and all of their infinite copies in PBC:

$$U = \frac{1}{2} \sum_n^N \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{r_{ij,n}} \dots \dots \dots (3.11)$$

A single slowly as well as conditionally convergent series, Eqn. (3.11), was rewritten by Ewald as the sum of two swiftly convergent series plus a constant term,

$$U_{\text{Ewald}} = U^r + U^m + U^o \dots \dots \dots (3.12)$$

As a result, the Ewald sum is denoted by the total of these three components: the real (direct) space sum (U_r), the reciprocal (imaginary, or Fourier) sum (U_m), and the constant term (U^o), also called the self-term.

3.1.1.5. Dealing with molecules: SHAKE algorithm:

The numerous time scales associated with vibrational degrees of freedom, such as bond vibration, angle stretching, or torsional mode within a molecular system, constrain the choice of time step. The quicker vibrational mode of the bonds involving hydrogen atoms limits the integration time step to 1 fs. However, one can confine these quick

degrees of freedom while solving the unconstrained degrees of freedom in order to require a longer time period. The SHAKE method, developed by *Ryckaert et al.*, can confine dynamics for bonds involving hydrogen since they have the highest frequency [395]. The SHAKE algorithm's first step is to relay the atomic system's unrestricted equations of motion. The primary principle of the SHAKE algorithm is to enforce bonding distances constant by using the Lagrange multiplier formalism. Assuming N_c , the constrained is given by:

$$\alpha_k = r_{k_1 k_2}^2 - R_{k_1 k_2}^2 = 0, \text{ Where } k = 1, 2, 3, \dots, N_c \dots \dots \dots (3.13)$$

The terms $R_{k_1 k_2}$ is considered as being constrained distant between k_1 and k_2 atoms. The modified constrained equation of motion is defined as:

$$m_i \frac{d^2 r_i(t)}{dt^2} = - \frac{\partial}{\partial r_i} [V(r_1 \dots r_N) + \sum_{k=1}^{N_c} \tau_k(t) \alpha_k(r_1 \dots r_N)] \dots (3.14)$$

Here m_i is defined as the mass of i^{th} particle, τ_k as the Lagrange multiplier (unknown) for k^{th} constraint. This modified constrained equation of motion can however be solved for unknown multiplier through solving N_c quadratic coupled equations. Finally, the following equation of motion has been obtained:

$$r_{k_1}(t + \Delta t) = r_{k_1}^{uc}(t + \Delta t) - 2(\Delta t)^2 m_{k_1}^{-1} \tau_k(t) r_{k_1 k_2}(t) \dots \dots \dots (3.15)$$

In the equation, r^{uc} is the position updates with unrestrained force only. However, unless the stated tolerance is supplied, this technique is repeated.

The SHAKE algorithm technique avoids the explicit matrix inversion, however, by repeatedly modifying particle coordinates until the system fulfils all criteria 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 buildup of numerical mistakes. However, because each time step's convergence must occur within a certain tolerance, iterative algorithms automatically provide correct constraint decay. Frequent corrections and checks are made to the confined distance deviations coming from the original values. The lack of a natural feedback mechanism for noticing changes in distance meant that non-iterative algorithms needed an intentional approach to combat constraint degradation.

3.1.1.6. Boundary conditions:

We will take into consideration a system containing N interacting particles at temperature T , and in a volume V in order to comprehend the periodic boundary conditions. To ensure that the system is bound by duplicates of itself, we must add periodic boundary constraints analogous to the *2D Ising system*. As a result, it can be seen that given a system of particles, when a particle exits the centre box on one side, it must reenter it on the other side. According to **Figure 3.2**, the molecules' atoms are arranged in a hypothetical box bordered by translated copies of their coordinates. According to Figure 3.2, several copies of particle 3 that are present in the centre box can theoretically interact with particle 1 inside the central box. Additionally, it is appropriate to take into account a specific interaction that exists between particles 1 and 3, and it is only logical to choose the interaction that results in the shortest interatomic distance. The nearest image convention is the term used to describe this approach. The inner cell is known to be surrounded by a periodic 3-dimensional array. When an atom crosses a barrier and enters from the other side with the same velocity, it is substituted by an image atom. From this point on, the volume of the core box's particles stays constant. To cope with non-bonded interactions, however, a non-bonded cutoff is generally employed such that each atom in the system interacts with just one image of every other atom present in the system.

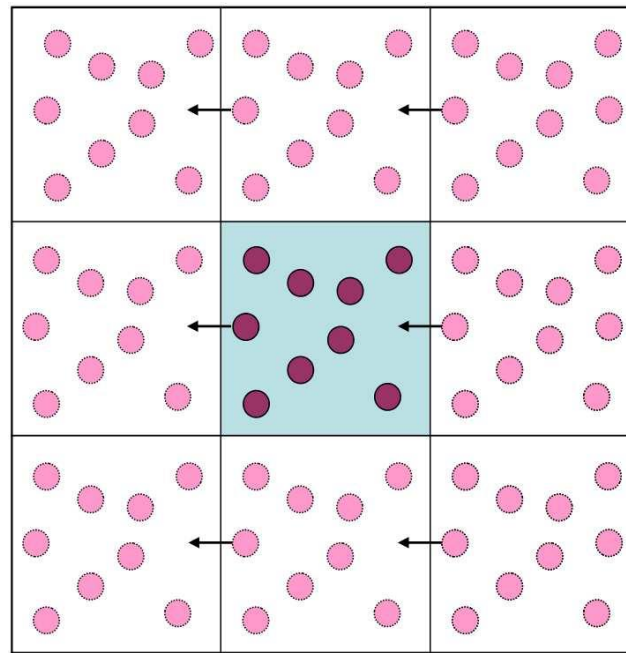


Figure 3.2. Periodic boundary conditions in two dimensions. The simulation cell (dark color) is surrounded by translated copies of itself (light color).

3.1.1.7. Temperature and Pressure computation and control:

In order to accomplish the isothermal MD simulations, a variety of approaches are currently being developed. Theoretically, using a thermostat is comparable to adding a made-up heat bath to the system to keep the average temperature **T** at a specific target temperature **T₀**. But for a specific particle **i** the heat bath is still relevant, either by changing the particle's speed or by changing Newton's equation of motion:

$$m_i \frac{d^2r_i}{dt^2} = F_i(r_i) \dots\dots\dots (3.16)$$

When there is no temperature control, the system typically develops according to the equation described above, which has micro-canonical (NVE) energy distribution. For a system that is characterised by a given force field, this micro-canonical ensemble provides the value of "real" dynamics, like classical Newtonian dynamics, at the accuracy level which is constrained by the integration technique and the force computations. The speed at which T deviations are "pulled back" to T₀ and the intensity with which a heat bath is delivered to the system are determined by the coupling

strength τT . In actuality, T is more closely regulated at $\tau T \rightarrow 0$ and also less tightly controlled at τT , but the physical significance of τT depends on the algorithm as the system approaches the NVE ensemble at $\tau T \rightarrow \infty$. When an algorithm successfully applies the canonical distribution, only the change rate of the simultaneous kinetic energy is affected. The distribution of the kinetic energies sampled is unaffected.

The system temperature $T(t)$ deviating from the bath temperature T_0 is corrected in a way such that:

$$\frac{dT(t)}{dt} = \frac{1}{\tau} \{T_0 - T(t)\} \dots \dots \dots (3.17)$$

In the above mentioned *equation* (3.13), the τ (time constant) gives a value of the strength of the coupling between the system and the bath. The system's temperature is modified by scaling the atom velocities at each step with a factor χ from the equation, given by:

$$\chi = [1 + \frac{\Delta t}{\tau T} (\frac{T_0}{T(t)} - 1)] \dots \dots \dots (3.18)$$

However, altering the time constant τ can alter the coupling strength.

There are techniques to test that randomise the particle velocities in order to manage the temperature among the six thermostat algorithms. A portion of the simulation's atoms are randomly picked at each Andersen thermostat time step, and fresh velocities are then determined from the Maxwell-Boltzmann distribution that correspond to the desired temperature [396]. Each particle experiences stochastic collisions that, on average, happen once every τT , and in the time between collisions, each particle moves according to Newton's equation of motion. The chance of selecting a certain particle for velocity reassignment at a particular time step is equal to $\Delta t / \tau T$ (where Δt is the time step) or equal to the size of the subset of the particles at that step that will be randomly selected. Stricter temperature control is gained when the probability of randomising a particle's velocity is high, which corresponds to low values of τT . The "*massive Andersen*" thermostat, which randomly assigns particle velocities at intervals of τT , is also put to the test. Both Andersen techniques, which do not directly modify the integration equations themselves, provide energy distributions that are compatible with the canonical ensemble [397].

Similar to this, a simulation of a constant pressure might be performed by utilising a "barostat" constructed with extra pressure-controlling factors. However, the variables and their dynamics can simulate the external environment that controls the temperature and pressure averages over time. The pressure may be kept constant, though, and the system can be connected to a barostat by periodically scaling the simulation cell size as well as atomic locations with a factor of μ :

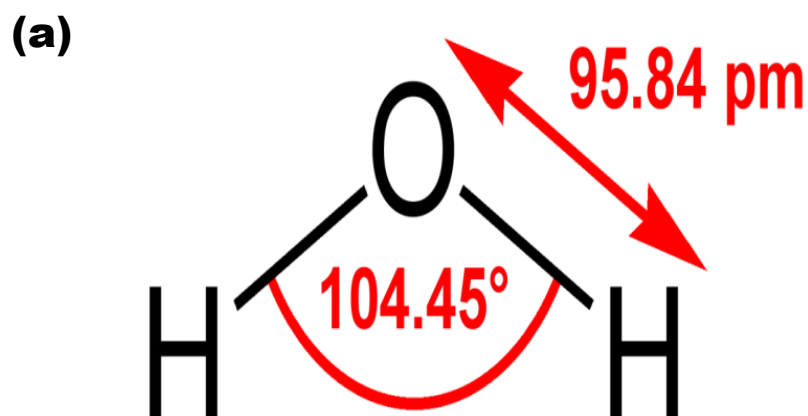
$$\mu = 1 - \omega \frac{\Delta t}{\tau_p} (P - P_0) \dots \dots \dots (3.19)$$

Here ω is defined as the isothermal compressibility, P_0 is the pressure of the barostat, P , the momentary pressure at time t and Δt is the time of step and τ_p represents the relaxation constant. For this work, the AMBER 14 standard simulation package is utilised [398]. The AMBER module's Pmemd one is utilised to perform the molecular dynamics.

3.1.1.8. Water molecule models:

Computer simulations of the biomolecular systems play a vital role in research by revealing details about the structure, dynamics, as well as energetics of biomolecules that are unavailable to experimental measurement methods. Different molecular models, however, are put forth that provide information for water in MD simulation. Site count, polarisation effects, and model structure are used to describe these models (rigid or flexible). The fact that the (known but hypothetical) model (i.e., computer water) can accurately predict the physical characteristics of liquid water demonstrates the significance of water models. This is because it reveals the (unidentified) structure of liquid water. There is a barter between the size that can be computed in a reasonable amount of time, the computational sophistication of the system, and all three of these factors. Even as computational power rises considerably year over year, the boundaries established by the system size, time constraints, and model complexity are being tested. Because of their simplicity, thermodynamic explanations, computational effectiveness, and logical structure, 3-site water models are the most often used ones in MD simulations. A water molecule's three atoms can interact with these models through

three different places. Each atom has a point charge that is unique to it. The only atom that has Lennard-Jones parameters enabling interaction is oxygen, though, out of all the atoms. One or more of the charged sites may or may not be covered by the models made up of Lennard-Jones sites as well as orienting electrostatic effects. However, Lennard-Jones interactions are important for determining molecule size. At very close ranges, this contact is regarded as repulsive, demonstrating that the electrostatic interactions prevent the structure from totally collapsing. At intermediate distances, it is very attractive but non-directional and competes with the directionally attractive electrostatic interactions. Transferable intermolecular potential three-point (TIP3P) model, simple point charge (SPC) model, extended simple point charge (SPC/E) model, etc. are a few prominent 3-site models [399]. However, each of these models uses a hard geometry that is consistent with the water molecule's known shape. The TIP3P water model is used in the simulation in this thesis. The O-H bond length (r_{OH}) as well as H-O-H bond angle (θ_{HOH}) of the TIP3P water model that we employed in this investigation are quantified to be equivalent to experimental gas-phase values at 0.9572 Å and 104.52°, respectively. **Figure 3.3** illustrates the schematic layout of a simple TIP3P water model.



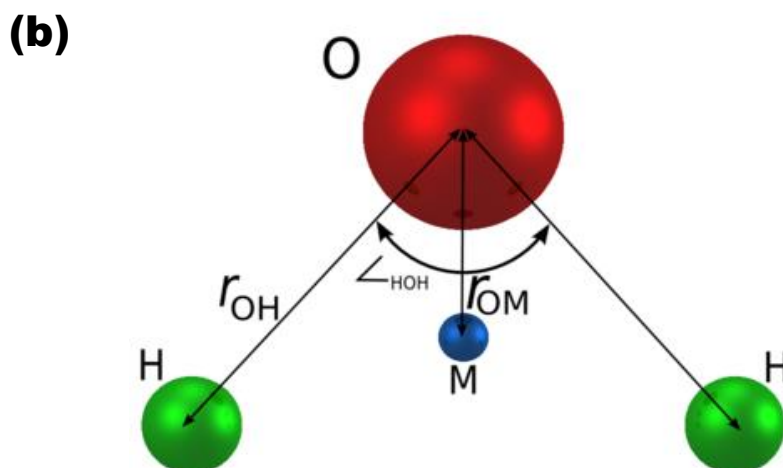


Figure 3.3. Schematic representation of TIP3P water model.

Four-site water models also developed after three-site water models. The first of the four-site models is the Bernal and Fowler model, which was developed in 1933 and is only sometimes used now but is nonetheless important for historical reasons [400]. On the bisector of the HOH angle, 0.15 Å from the oxygen atom, in that specific model, the negative charge is transferred from the oxygen and transported towards the hydrogens. Nevertheless, the oxygen atom is in the core of Lennard Jones' interaction site. An evaluation of the interaction function for a three-site model requires ten distances rather than nine. In contrast, a five-site approach, like the ST2 model, calls for 17 participants. Later, based entirely on the BF water model, Jorgensen and colleagues created the TIP5P as well as TIP4P potentials [401]. The scientists demonstrated that, in contrast to the experiment, TIP4P replicates more of the attributes of liquid water, including heat capacity (+7.3%), internal energy (+1.5%), enthalpy of vaporisation (+1.4%), and density (+0.2%).

In addition to these four- and five-site water models, the five-site TIP5P model was created by further extending the TIPnP class of potentials [402]. Using one LJ interaction point per molecule, centred on the oxygen atom, and charges on the hydrogen atoms and two lone pair sites, the van der Waals interaction that exists between the two water molecules, identical to the ST2 model, is calculated. In TIP5P, the clumsy cubic scaling function was dropped in order to reduce the short-range

electrostatic interactions present in the ST2 model. However, from -37.5 to 62.5 degrees Celsius, the TIP5P water model replicates energy and the density with an average inaccuracy of less than 1%. Furthermore, given the right temperature dependency, the dielectric constant started getting close to 80. However, the TIP5P parameters were obtained using a straightforward spherical cutoff approach that was available for long-range electrostatic interactions. It has been demonstrated that when the TIP5P water model [401] is applied along with techniques that take into account long-range electrostatic interactions, such as Ewald summation [403] as well as reaction field methods, [404], the calculated properties only offer a passable degree of agreement with the experimental data.

In a recent work, Nada and van der Eerden developed a six-site model for water [405], in which each hydrogen atom has a positive point charge, similar to the TIP5P water model, while each lone-pair site has a negative charge. Additionally, similar to the TIP4P water model, a negative charge also is applied to a site that is situated on the HOH angle's bisector. The LJ interaction affects both the hydrogen sites and the oxygen atom, in contrast to the other two water models. The six-site water model was designed in order to represent the ice and liquid water around the melting point. The structural and thermodynamic characteristics of ice and water that are near to the melting point were accurately replicated.

The employment of several models of this type is undoubtedly an approximation, and certain features cannot even be predicted. The vibrational spectrum is one illustration of this. Flexibility might be incorporated by adding bond stretching and angle bending elements to a rigid model's potential function. Ferguson, using the SPC model as a foundation, created a flexible water model by utilising harmonic angle bending, cubic plus harmonic bond stretching terms, as well as harmonic bond stretching terms [406]. The LJ parameters as well as charges were re-parametrized in that model, and the resulting values differed somewhat from the values of the rigid SPC model.

3.1.1.9. Molecular Dynamics steps:

There are typically four phases involved in propagating a molecular system utilizing the aforementioned equations:

- 1. Energy Minimization**
- 2. Heating**
- 3. Equilibration**
- 4. Production Dynamics**

1. Energy Minimization:

The information obtained from molecular dynamics or Monte Carlo is reproduced through energy minimization. Despite their importance in calculating thermodynamic averages and estimating entropy, Monte Carlo structures or ensembles of dynamics are too numerous to analyze in depth at the microscopic level. Although they represent the underlying configurations connected to the fluctuations that occur during dynamics, the minimized structures nonetheless serve as a useful and significant starting point for structural research [407-408]. A molecular structure's energy reduction happens in two stages. The first step is to define and then assess an equation that expresses the system's energy as a function of its coordinates for a certain conformation.

However, the issue of energy minimization might be formulated as follows: given a function f and one or more independent variables x_1, x_2, \dots, x_i , it must be shown that the minimal value of f finds the value of each of the independent variables. The function's lowest point's first derivative for each variable is zero, while its second derivatives are indeed positive:

$$\frac{\partial f}{\partial x_i} = 0; \frac{\partial^2 f}{\partial x_i^2} > 0 \dots\dots\dots (3.20)$$

Molecular mechanics involves a lot of minimization, which is often done in Cartesian coordinates with energy as a function of the $3N$ variables. This occurs most frequently when quantum mechanics is used and internal coordinates specified in the Z-matrix. The minimization algorithms may be divided into two categories: those that

employ the energy derivatives with respect to the coordinates and those that do not.

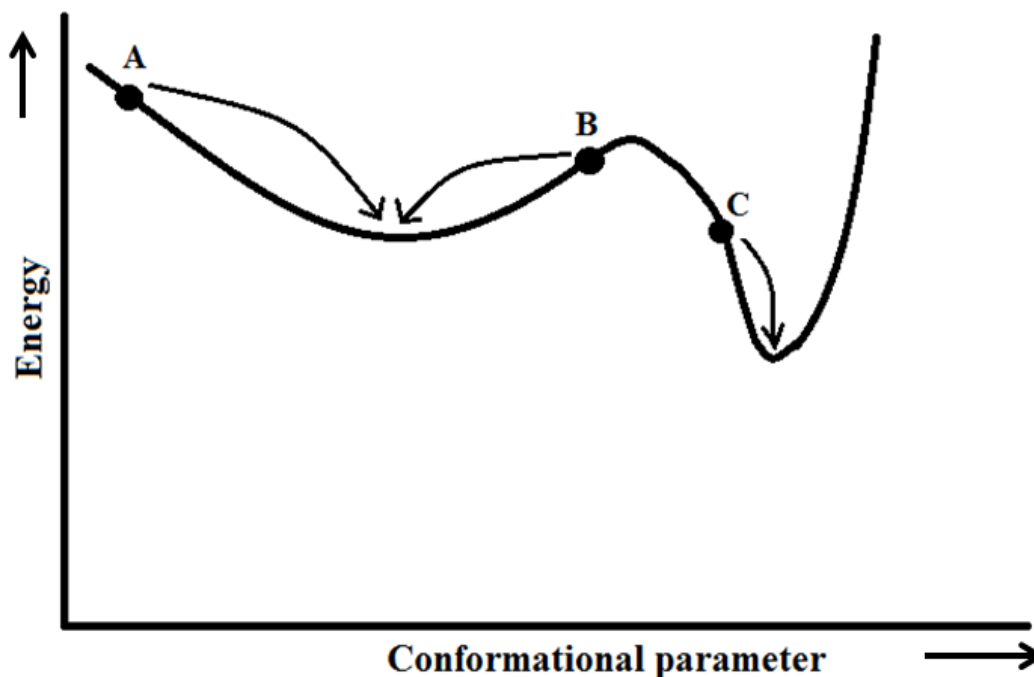


Figure 3.4. A schematic one-dimensional energy surface. Minimisation methods move downhill to the nearest minimum.

However, the technique used for the majority of energy reduction can identify the lowest that is located the closest to the beginning point by going downhill on the energy surface. So, from **Figure 3.4**, we deduced that an energy surface with minima might be produced by beginning from the three locations A, B, and C. Different starting points must be generated in order to locate the global energy minima, and many minima must be found [409]. The minimum has been reached when the derivatives are getting near to zero. To eliminate the poor connections prior to MD simulation and prevent structural deformation, it is crucial to conduct energy minimization on the structure.

To employ a derivative minimization technique, the derivatives of energy with respect to the variables (i.e., Internal or Cartesian coordinates, as applicable) must be calculated. However, it is possible to obtain the energy derivatives analytically or numerically. These derivatives are employed by a number of well-known minimization strategies and offer a wealth of knowledge that is beneficial to energy minimization strategies. While the gradient's magnitude indicates how steep the local slope is, the

direction for the first derivative's energy defines the location of the minimum. It is possible to reduce the energy of the system when the force equals minus the gradient by having each atom move in reaction to the force exerted on it. The second derivatives give information about the function's curvature as well as information that may be used to anticipate where the function will change course (by moving through a minimum or another stationary point). Steepest descents and conjugate gradient methods are the two methods most frequently employed in molecular modelling for the first-order minimization procedures.

(i) *The Steepest Descents Method:*

The steepest descent technique follows a path parallel to the net force, which, according to the geographical comparison, is the same as travelling downward straight. The $3N$ -dimensional unit vector, \mathbf{s}_k for the $3N$ Cartesian coordinates, is the most appropriate representation of this specific direction. Hence:

$$\mathbf{s}_k = -\mathbf{g}_k/|\mathbf{g}_k| \dots\dots\dots (3.21)$$

It is crucial to choose how far the gradient may go by deciding which direction to travel in. observing the **Figure 3.5** two-dimensional energy surface. If we visualize a cross-section of the surface all along line, the beginning point from where the gradient direction begins is along the line. The function would pass through a minimum, as seen in the picture, and then grow. We can find the minimum point by running a line search or by taking a step of any size along the force's direction [483].

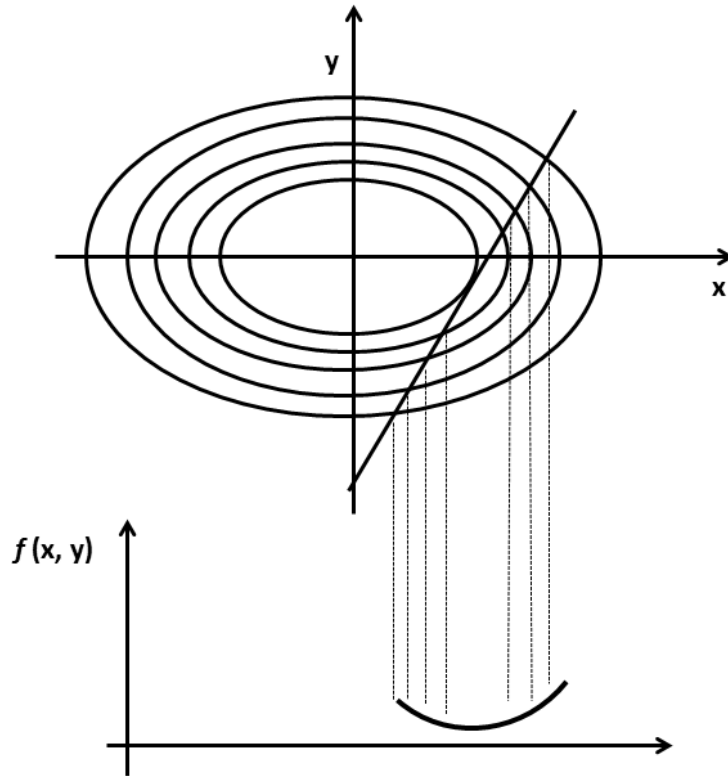


Figure 3.5. A line search is used to locate the minimum in the function in the direction of the gradient.

(ii) Conjugate Gradients Minimization:

The conjugate gradient approach, used in the energy minimization methodology, produces a collection of directions that do not exhibit the steepest descents oscillating behaviour in the narrow valleys. While the directions are conjugate, the gradients at each location in the conjugate gradient technique are orthogonal. From a set of conjugate directions with the condition that given a quadratic function of M variables, the minimum will be obtained in M steps. The conjugate gradient technique advances in the direction \mathbf{v}_k from point \mathbf{x}_k , where \mathbf{v}_k is derived using the gradient at the point as well as the prior direction vector \mathbf{v}_{k-1} [483].

$$\mathbf{v}_k = \mathbf{g}_k + \gamma_k \mathbf{v}_{k-1} \dots \dots \dots (3.22)$$

In the following equation, γ_k is a scalar constant that is given by:

$$\gamma_k = \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_{k-1} \cdot \mathbf{g}_{k-1}} \dots \dots \dots (3.23)$$

However, for the purpose of finding a minimum, the second-order approaches utilise

both the first and second derivatives. The function's curvature is disclosed in this information. Newton-Raphson is the most straightforward second-order approach for energy reduction.

(iii) Newton-Raphson:

This second-order Newton-Raphson approach is better appropriate for tiny molecules with around 100 atoms or less. The *Newton-Raphson* technique explores a minimum for a strictly quadratic function, such as the function $f(x, y) = x^2 + 2y^2$, in just one step from any surface point. In addition to this, it also makes use of the curvature to forecast where the gradient of the function would turn. The most computationally costly approach to energy reduction is thought to be this one. There are several versions of this technique that try to do away with the requirement to calculate the entire matrix of second derivatives. Except for a family of techniques called the quasi-Newton method, just the first derivative is needed, which may be used to gradually build the inverse Hessian matrix when the computation goes along. Using the same Hessian matrix throughout the Newton-Raphson algorithm's many subsequent phases, with just the gradients being computed after each iteration, is one straightforward technique to perhaps speed up the Newton-Raphson process. This method is mathematically expressed as:

$$r_{min} = r_o - A_o^{-1} \cdot \nabla V(r_o) \dots \dots \dots (3.24)$$

where r_{min} is the predicted minimum, r_o is an arbitrary starting point, A_o is the matrix of second partial derivatives of the energy with respect to the coordinates at r_o (also known as the Hessian matrix), and $\nabla V(r_o)$ is the gradient of the potential energy at r_o .

If solvation is needed before using energy minimization techniques, water molecules are supplied to the system to solvate it. An appropriately sized, already-equilibrated huge box of water is used for the solvation process. The water box completely encloses the system, eliminating the water molecules that were covering the proteins. Proteins must be fixed in their energy-minimized positions for energy minimization to take place.

The water molecules can then readjust to the protein molecule in this way.

Energy minimization is frequently utilized for molecular modelling and is a crucial component of methods like conformational search algorithms. The method of energy minimization is also used to set up the systems needed for various kinds of computations. Prior to doing either a molecular dynamics simulation or a Monte Carlo simulation, energy minimization may be performed to eliminate any negative interactions in the system's initial configuration. This is mostly used to simulate complicated systems like macromolecules or big molecular assemblies.

2. Heating:

The Newton's equations of motion, which depict the system's temporal development, are numerically integrated during the heating phase, during which starting velocities (at 0 K) are assigned to each atom in the system. New velocities are assigned at brief, predetermined intervals that correlate to slightly higher temperatures, and the simulation is then allowed to run until the target temperature is reached (that is 300 K). As structural stresses are loosened by heating, force constraints on various simulation system subdomains are gradually lifted. The normal operating condition for heating dynamics is a constant volume (NVT).

3. Equilibration:

As a system evolves from its initial configuration, equilibrium is reached during the equilibration phase. Equilibration should go on indefinitely, or at least until a set of monitored attributes' values stabilise. The key measured properties include thermodynamic variables including energy, temperature, and pressure as well as structural characteristics. The simulations of the liquid state, however, have a beginning structure that resembles a solid lattice. In reality, it's essential to set things up such that the lattice has "melted" before the manufacturing phase begins. The order parameters may be utilised to determine when the liquid state will reach its reaching point. The measurement of a system's degree of order is what is meant by this order parameter.

While simulating a crystal lattice, the atoms could, nevertheless, stay in about the same place throughout, maintaining a high level of order. The species that are present in a liquid are likely to move about quite a bit, which might cause translational disorder.

Molecular dynamics is the process of resolving the atomic system's motion equations. The answer to a molecule's motion equations represents the trajectory and temporal evolutions of its molecular movements. Depending upon the temperature at which the simulation is performed, MD enables the bridging of barriers and the investigation of many possibilities. To start the MD, velocities must be assigned at first. This is accomplished by employing the Maxwell-Boltzmann distribution restriction for the random number generator. According to the kinetic theory of gases, the system's average kinetic energy determines the temperature. $U = \frac{3}{2} NkT$ is the unit of measurement for the system's internal energy. The system's kinetic energy is given by $U = \frac{1}{2} Nmv^2$. But by averaging the velocities of all the atoms present in the system, it is possible to calculate the temperature. Once the initial set of velocities is established, the Maxwell-Boltzmann distribution could be preserved throughout the simulation.

Following energy reduction, the temperature may be thought of as being strictly zero Kelvin. The dynamics must first be initialised before the system can be heated to the desired temperature. At a low temperature, the velocities are assigned, and dynamics is carried out in accordance with the equations of motion, in order to do this. However, after several dynamics iterations, the temperature is scaled higher. Under atomic restrictions, a 20 ps timeline of progressive heating dynamics is carried out from 0 to 300 K. The most popular method of temperature scaling among them all is velocity scaling. For equilibration time steps of 1 fs, a run of at least 5 ps (5000 time steps) and often 10 or 20 ps is required. Dynamic equilibration is carried out for 100 ps after heating.

4. Production Dynamics:

It is mostly used to calculate thermodynamic averages or sample new configurations at this period of interest, dynamics. Calculations are being made about additional data as well as the thermodynamic characteristics at this stage. The parameters, such as the

kinetic, potential, and total energy, the velocities, the temperature, and the pressure, mostly depend on the simulated system to decide whether or not the equilibrium has been reached. The kinetic and potential energies, however, may vary in a simulation of the micro canonical ensemble while the total energy is kept constant. The three directions of x, y, and z should each have an equal amount of kinetic energy, and the components of the velocities should fit into a Maxwell-Boltzmann distribution. During the manufacturing stage, the system's variable is the temperature. At the start of the production phase, all counts are reset to zero and then the system is left to develop. Temperature becomes a computed attribute of the system since no velocity scaling is done during the creation phase of a micro canonical ensemble. Throughout the production phase, the attributes are accurately computed and saved for further processing and analysis. It may be necessary to restart the simulation if issues arise if the simulation is being watched closely based on its behaviour. It's also usual practise to save configurations' energies, locations, and velocities across time so that the other attributes may be ascertained when the simulation is complete. Though it is possible to compute the thermodynamic parameters while running the MD simulation. A few hundred ps to ns or even more are used to create the production run. The different processes necessary to set up and conduct an MD simulation are covered in more depth below and are represented in **Figure 3.6** as a flowchart.

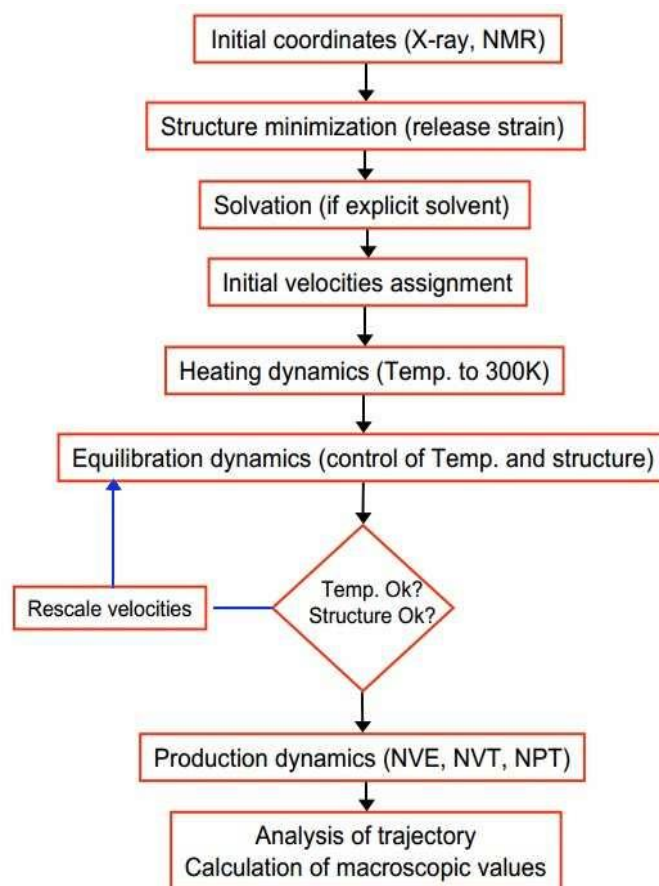


Figure 3.6. Schematic flowchart of steps involved in MD Simulation.

3.1.2. Potential of mean force (PMF):

The chosen reaction coordinate's (eg. distance of separation between two molecules) free energy surface is referred to as the PMF. The distribution function of that coordinate may be thought of as being essentially connected to the reaction coordinate, which can be thought of as the separation between two atoms or the torsion angle of a bond. When the system is contained in a solvent, PMF takes into account both the effects of the solvent and the intrinsic interaction that exists between the two particles. The greatest energy point on the free energy profile corresponds to the transition state of the process, and from this point, the rate constant may be calculated. The PMF may be calculated using a variety of techniques. The simplest sort of PMF involves a change in free energy together with a change in the reaction coordinate as a result of a change in the spacing

(r) between the two particles. This free energy is defined as [410].

$$A(r) = -k_B T \ln g(r) + \text{constant} \dots \dots \dots (3.25)$$

The choice of this constant in the equation is made so that the most likely distribution corresponds to a zero free energy. However, over the important range of the parameter r , PMF diverge from distinct multiples of $k_B T$. The radial distribution function and PMF have a logarithmic connection, and this suggests that a very minor change in the radial distribution function might result in a change in $g(r)$ of an order of magnitude. The MD simulation approach does not effectively sample places where the radial function deviates significantly from the value that is considered to be the most likely value, resulting in erroneous PMF readings. The PMF used to calculate the free energy can also be utilised as a coordinate in geometry or as a more basic energetic (solvent) coordinate. In contrast to mutations in estimates of free energy perturbations, which frequently follow non-physical routes, it may also be utilised to calculate physically realisable processes. It is helpful for clarifying the reaction mechanism of the condensed phase reactions, such as enzymatic processes, and estimating chemical solution rates. Umbrella Sampling (US) is one sampling method that might help with this issue.

3.1.2.1. Umbrella Sampling (US):

The sampling method known as US restricts a system to a restricted area of its conformational space as well as modifies the potential function to ensure that the unfavourable states are sampled correctly. Both Monte Carlo simulations and molecular dynamics simulations employ this technique. A bias is given to the system to ensure efficient sampling of the whole reaction coordinate. The distribution may be overlapped by targeting at a single simulation or a number of simulations (windows). The term US came about as a result of how the bias potential connected the energetically isolated areas in phase space. According to W, a bias potential is the extra energy component that mostly depends on the reaction coordinate:

$$\vartheta' (r^N) = \vartheta (r^N) + W (r^N) \dots \dots \dots (3.26)$$

$W (r^N)$ is a weighting function that takes a quadratic form such that:

$$W (r^N) = k_W (r^N - r_0^N)^2 \dots \dots \dots (3.27)$$

The weighting function will be higher for configurations positioned away from the equilibrium state r_0^N and thus using simulation the modified energy function $\vartheta' (r^N)$ will be biased along some relevant ‘**reaction coordinate**’ away from the configuration r_0^N . The distribution that results will unquestionably be regarded as non-Boltzmann. The Torrie and Valleau [484] approach may be used to obtain the relevant Boltzmann averages from the non-Boltzmann distribution. The result is:

$$\langle A \rangle = \frac{\langle A(r^N) \exp\left[+\frac{W(r^N)}{k_B T}\right] \rangle_W}{\langle \exp\left[+\frac{W(r^N)}{k_B T}\right] \rangle_W} \dots \dots \dots (3.28)$$

The modified energy function $\vartheta' (r^N)$, which is derived from the subscript W , is used to identify the average based on the probability $P_W (r^N)$. Usually, it is necessary to calculate the US in a sequence of steps that are distinguished by a certain coordinate value and a suitable forcing potential value $W (r^N)$. If the forcing potential is too great and also the average takes too long to converge, just a few configurations with particularly large values of $\exp [W (r^N)]$ contribute the majority of the denominator in equation 3.23.

3.1.2.2. Bias potentials:

It is preferable to choose the bias potential in a way that ensures uniform sampling over the whole range of reaction coordinate ξ . Thus, $w_{opt} = -A (\zeta)$ is chosen as the ideal bias potential. As a result, it might occur due to $P_i^b (\zeta)$ really uniform distribution. Although it is uncertain, we plan to use US to derive the parameter $A (\zeta)$. In actuality, two important families of bias potentials have been identified: harmonic biases within a series of windows along ξ , plus an adaptive bias, that is modified to match $-A (\zeta)$ in a single window encompassing the whole range of ξ .

3.1.2.3. Harmonic bias potentials:

To assure sampling in all areas of interest, the range of interest of ξ is divided into a number of windows. The system is kept near to the window's reference point ξ_i^{ref} by applying a specific bias function for each window i . It's common to utilise a straightforward harmonic bias of strength K :

$$\omega_i(\xi) = K/2 (\xi - \xi_i^{\text{ref}})^2 \dots\dots\dots (3.29)$$

Following the simulation, the US Simulations methodologies and the free-energy curves are integrated (typically weighted histogram analysis method (WHAM) or umbrella integration). The bias form presented in equation is intriguing since it just has a few parameters, including K (which, in theory, might be window dependent), the number of pictures, and ξ_i^{ref} . Most often, a bias with a uniform distribution along ξ is adopted. The statistical inaccuracy that is proportional to CPU time decreases as the number of photos increases. Contrarily, equilibration and increasing the quantity of pictures demand CPU time. The majority of MD simulation pictures are independent and parallelizable. The decision about K and the bias's intensity is the only crucial one. However, it must be made before the simulations are executed. A second series of windows might be added, though, if the first series causes significant gaps between the distributions. K must be sufficiently large to push the system beyond the barrier. There will be extremely narrow distributions $P_i^b(\xi)$ with excessively big K . The distributions must sufficiently overlap for WHAM to function. Umbrella integration has benefits even if it is not necessary. As the constant time step K is increased, there are growing inaccuracies in the numerical integration of the motion equation. When the time step is set too big, the configurations would have an excessive amount of higher energies (or K is too large). For umbrella integration analysis, which may frequently be calculated before sampling, analytical formulations for statistical error can be constructed, enabling an estimation of an ideal K based on the data. Additionally, it has been claimed that the position and widths of the window to be well sampled next (ξ_{i+1}^{ref}) can be set so that they match the predicted half maxima of the preceding window. An alternate strategy is to use the experimental results to establish the bias settings that are most beneficial.

3.1.2.4. Adaptive bias US:

With adaptive bias US, the reaction coordinate's (ξ) complete interest range is covered in a single simulation. Theoretically, this is possible by selecting a bias $w(\xi) = -A(\xi)$. This results in a flattening of the energy surface and consistent sampling along ξ . It is necessary to start with an estimate of $w(\xi)$ and improve iteratively to get a uniform distribution because $A(\xi)$ is not regarded as a priority.

3.1.2.5. Specialized umbrella potentials:

A history-dependent (and hence time-dependent) bias is added to the potential energy by the local elevation approach, much like meta-dynamics does. It has recently been integrated with the US technique by first creating a local elevation bias in a quick simulation, then sampling the distribution within the bias to reassemble the free energy. Aside from that, several umbrella potentials were employed.

3.1.2.6. Running the US calculations:

Running MD with a slack beginning structure is possible for the particular umbrella windows. The end points' overlap, which requires window 1 to sample some of window 2, etc., is the primary determinant of the number of windows. The force constant must be large enough to guarantee sampling of the phase space subsets. The force is not, however, so great that the windows cannot overlap because they are too small.

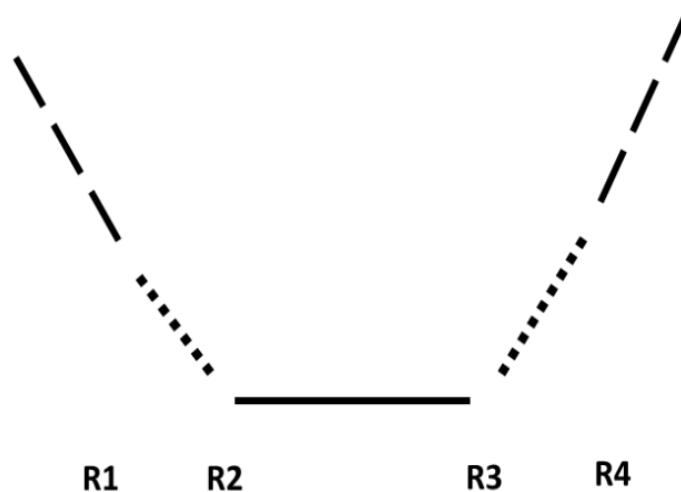


Figure 3.7. Working principle of US. Taken from [410].

Where,

\ = lower bound linear response region

/ = lower bound linear response region

... = parabola

_ = flat region

The size of the windows and the restrictions, however, may be altered depending on where one is standing along the walkway. The quantity of simulation we run for each window must be sufficient for our sampling to converge. The harmonic constraint is specified using a reference file where R1, R2, R3, R4 define a flat-welled parabola which is linear outside of a specific range. It will be harmonic between r1 and r2 with the force constant of rk2, it will be flattened between r2 and r3 and it will be harmonic between r3 and r4 with the force constant of rk3.

3.1.2.7. The weighted histogram analysis method (WHAM) for free-energy calculations:

An expansion of the single- and multiple-histogram approaches, the WHAM was

developed by Ferrenberg and Swendsen. To calculate expectations from canonical ensemble at any relevant temperature, it enables the reweighting of the configurations acquired from separate canonical simulations at various temperatures. It provides a number of benefits over the traditional US method while being an extension of that technique. In order to combine the simulation batches with different biasing potentials so that the impartial PMF can be seen, the WHAM approach is important. Overlapping the windows is necessary for the study of WHAM and is preferred for umbrella integration. The sample quality can be improved using Hamiltonian replica exchange. To calculate free energy and the PMF, the US simulation technique extension is employed. Over the approaches now in use, this algorithm has a number of advantages: **(1)** By including a built-in estimation of sample errors, one may get estimates of the best location and number of further simulations to run in order to reach a given degree of precision; **(2)** Obtaining the "best" free energy value by taking into consideration all simulations to reduce statistical mistakes; **(3)** Along with improving the connections between simulations, it permits numerous probability distribution overlaps for improved estimations of the free-energy differences. In addition to improving the connections across simulations, WHAM permits several probability distribution overlaps to get better estimates of free-energy differences. This approach provides the researchers with objective estimations of the ideal location and duration of subsequent simulations, enhancing the correctness of their conclusions. The majority of the simulations that result in the overlapping distributions are also explained by this. The overlapping histograms serve as an ideal connection between the numerous simulations. It is easily utilized to produce PMFs and free energies in relation to the coupling parameter(s) h_i and/or temperature. This technique is important because it allows for the improvement of conformational sampling simulations by allowing for the extrapolation (or interpolation) of data to the target temperature at a range of temperatures [411]. The main objective of the WHAM equation is to get the best estimates of the probability density $P_{\{\lambda\}, \beta}(\{V\}, \xi)$ at given $\{\lambda\}$ and β . These equations also result in the R free energies: A_1, A_2, \dots, A_R -of the system linked to the R simulations.

3.1.2.8. Umbrella Integration:

An variant to the WHAM known as umbrella integration is used to combine the windows in the simulations of US with harmonic biases. Because of this, the difficulty with computing F_i is overcome by averaging the mean force instead of the distribution P . The significant differences between WHAM and umbrella integration are: **(1)** WHAM averages the unbiased image distributions whereas umbrella integration averages the mean force. **(2)** In umbrella integration but not in WHAM, the biased distributions are modelled by normal distributions. **(3)** The (non-normalized) weights used to combine the windows differ. However, the equation for the umbrella integration allows for statistical error estimation in A using MD simulation data. As a consequence, by limiting the need for CPU time, this may be used to set the simulation's parameters, such as the intensity of the bias K and also the number of windows, in order to reduce statistical error. However, this technique may also be used to multidimensional reaction coordinates. The required integration phase becomes more challenging for more dimensions; however, the alternate WHAM analysis may be expanded to more dimensions more easily.

The independence of the grid points (bins) number as well as the accessibility of the error estimate are what distinguish the umbrella integration above the WHAM. It may be inferred from the fact that only ξ and σ^2 undergo the umbrella integration study. Additionally, this may be used to check the MD runs for the equilibration of these two parameters. For WHAM, where the entire distribution is considered in the analysis, it is really impossible to do so directly. When equilibration is carried out, the entire distribution is presumed to be equilibrated. The equilibration of ξ and σ^2 can be checked. The analysis is expedited by the umbrella integration's non-iterative nature. The amount of CPU time needed for the analysis seems to be quite small in contrast to the time needed for sampling MD data. There is a significant reduction in noise when going from A_i (ξ) to the second order found in ξ . When there are extremely few windows, this can potentially lead to mistakes.

3.1.3. Molecular Docking:

One of the most difficult tasks in structural biology is the computerised prediction of protein-protein interactions and protein-small molecule interactions. Many biological research, both in the academic and industrial worlds, may profit from accurate, dependable interaction prediction. To accurately associate two interacting molecules is the challenge in protein-protein docking. On the interactions between residues engaged in the target interaction, the precise prediction is built. There have been several docking methods created [412-416]. However, there are now just a few number of algorithms that may be used for free online. Most of the differences between the algorithms are found in the search strategy used and in how the resolved complexes are assessed in the six-dimensional transformation space.

In order to understand the behaviour of small molecules at the target protein's binding site or to obtain the interacting interface residues participating in protein-protein interactions, molecular docking is used to model the interaction between a protein and a small molecule or interaction between two proteins at the atomistic level [417]. There are two phases in the docking procedure. The position of the ligand at the binding site is determined in the first step. The conformers of ligands are given rank in the second stage using a scoring formula that is based on binding affinity. The scoring function must be able to score the experimental binding mode as being the greatest among all the produced conformations once the sampling algorithms first replicate it. In this thesis we used PatchDock server [418] protein-small molecule docking, and ClusPro server [419] for protein-protein docking.

3.1.3.1. PatchDock:

The stiff docking of molecules, like protein-protein or protein-drug interactions, is carried out online via PatchDock, a docking service, with surface variability taken into account during intermolecular penetration [420]. Geometry molecular docking method serves as its foundation. Additionally, it looks for docking modifications that provide

strong complementarity between molecule shapes. When these docking changes are applied, it produces both minor quantities of steric conflicts and large interface regions. Wide interface region is verified to have many matching local characteristics of the attached molecules with complimentary properties. This approach divides the molecules' representation of the Connolly dot surface into concave, convex, and flat patches. The next step is matching complimentary patches to create the candidate transformations. Every potential transformation is further assessed using a score formula that takes into account both geometric fit and atomic desolvation energy. RMSD (root mean square deviation) clustering is then applied to the candidate solutions in order to eliminate the duplicate ones. The PatchDock's great efficiency is largely attributable to its quick transformational search, which is sped up by local feature matching rather than by brute-force scanning the six-dimensional transformation space. The computational processing time may be further sped up by making use of sophisticated data structures as well as spatial pattern recognition methods, including geometric hashing and posture clustering, established in the area of computer vision. On one 1.0 GHz PC processor running the Linux operating system, the PatchDock runtime for two protein inputs of typical size (approx 300 amino acids) is <10 minutes or less. The basic characteristic of this technique is based on the Kuntz algorithm for local form feature matching (421). The proper conformation is maintained via the docking approach, which recognises the higher probability molecule surface areas that are present in the binding site. This method successfully docks big proteins with tiny drug molecules by handling receptors as well as ligands of various sizes. Additionally, the algorithm passed the Critical Assessment of Prediction of Interaction (CAPRI) test [422-431]. There are three key steps via which this algorithm operates to carry out its tasks:

- 1. Molecular Shape Representation:** This stage involves computing the molecule's molecular surface, and as a result, we use a segmentation technique to identify geometric patches (concave, convex, and flat surface pieces). As a result, only geometric patches made up of "hot spot" residues are maintained after being filtered.

2. Surface Patch Matching: The patches discovered in the previous phase are matched in this stage using a mix of the Geometric Hashing and Pose-Clustering matching algorithms. Patches that are concave and convex go together, and any additional patches go with them. There are two distinct approaches for matching the patches, which are:

- i. Single Patch Matching:* Small ligands like medicines or peptides are docked in this matching process, matching the receptor patch to the ligand patch.
- ii. Patch-Pair Matching:* Protein-protein docking is used in this matching process to match two patches on the receptor and two patches on the ligand.

3. Filtering and Scoring: The potential complexes from the preceding stage have undergone extensive research. Complexes with inappropriate interactions between the receptor and ligand atoms are eliminated. As a result, the remaining alternatives are ranked according to their complementarity with geometric shapes.

- I. Steric Clashes Test:* In this specific stage, the distance transform grid is frequently employed. The ligand surface locations are where change is mostly applied. The receptor's distance transform grid and each surface point's coordinates are matched after that. If the transformation's distance falls below each surface point's penetration threshold, it is kept for the following phase; otherwise, it is rejected.
- II. Geometric Scoring:* This step is significant because it separates the receptor into shells dependent on how far they are from the molecular surface. A range of distances in the distance transform grid defines each shell. Geometric score is defined as the weighted average of each of the shells, where the candidate complexes with many points in the shell and fewest points in the "penetrating" shells are preferred.

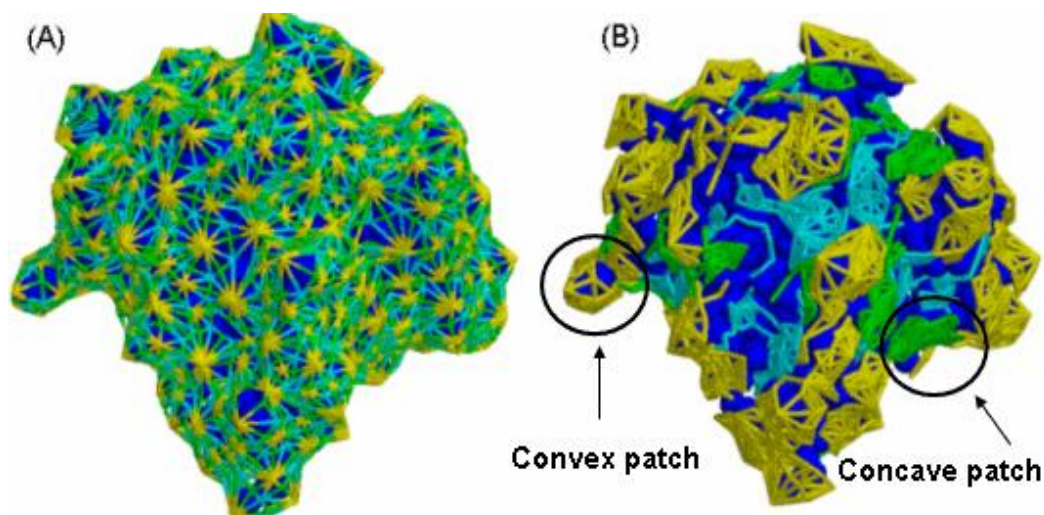


Figure 3.8. (A) Diagrams of surface topology for a protein's three-dimensional structure. The caps, belts and pits are connected with edges. (B) Geometric patches: the patches are depicted in light colors and the protein is depicted in dark colors. Modified from [418].

Input

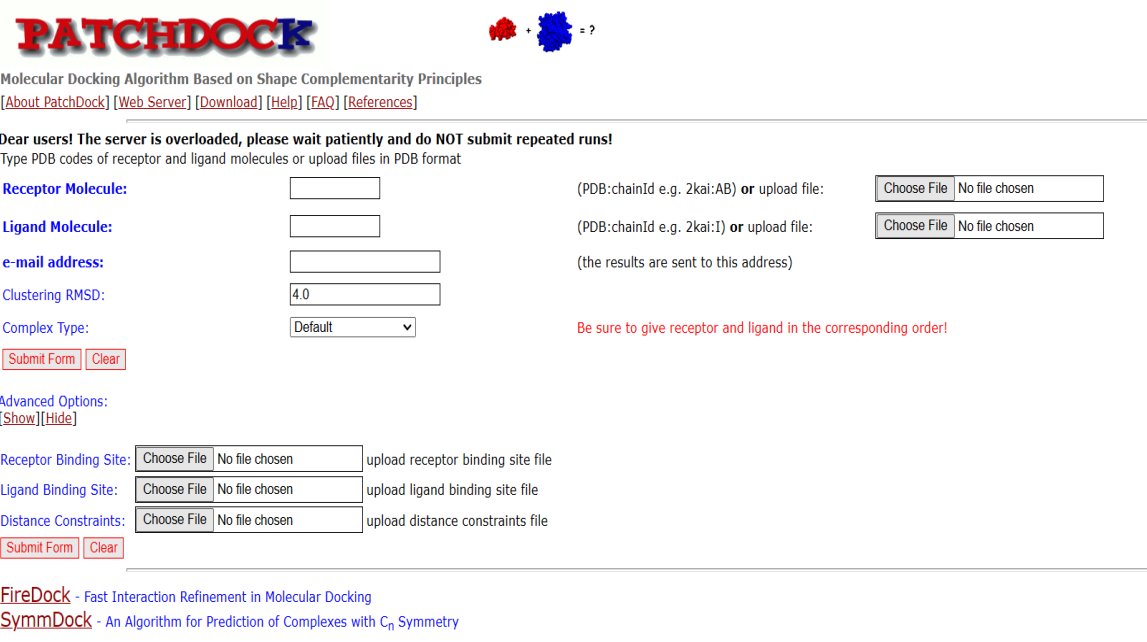
The two molecules (to be docked) in PDB format serve as the algorithm's input. Either the Protein Data Bank is accessed to obtain the molecules for the algorithm, or the molecules are submitted to the server. In the second incident, the user need just input the PDB code. In order to dock a particular chain or chains, the user must give the appropriate chain ID or chains. User email is the other mandatory box on the docking request form for outcome notification. In addition, the docking request form has four more optional fields.

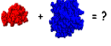
(i) **Clustering RMSD:** The angstrom-based RMSD clustering radius is represented by this positive value. During the algorithm's last clustering step, this field's value is mostly utilised. It demonstrates that for any given distance between any two solutions (output), the clustering RMSD value will be the lowest. RMSD's clustering parameter has a default value of 4 seconds.

(ii) **Complex Type:** PatchDock has a variety of parameter settings to optimise the various complexities. If the field however is not identified, the software will use the

default configuration. The enzyme-inhibitor complex type's search space is constrained by the algorithm to the cavities of enzyme molecules. However, the algorithm limits the search to the complementarity-determining regions (CDRs) of antibody in the antibody-antigen complex type. The parameter set used by the method for protein-small ligand docking is tailored for small-size molecules.

(iii) **And (iv)** Potential binding sites for a ligand and a receptor, respectively. Whenever the biological data identify particular residues as being a part of a probable binding site, the user might take into account the information provided in the algorithm. The specifics regarding the lists of residues for prospective binding site or sites are supplied in the file that has been uploaded. The format for the submitted file must be the same as that found in the PDB file for the receptor or ligand: a space between each line containing the chain ID and residue index.



PATCHDOCK 

Molecular Docking Algorithm Based on Shape Complementarity Principles
[\[About PatchDock\]](#) [\[Web Server\]](#) [\[Download\]](#) [\[Help\]](#) [\[FAQ\]](#) [\[References\]](#)

Dear users! The server is overloaded, please wait patiently and do NOT submit repeated runs!
 Type PDB codes of receptor and ligand molecules or upload files in PDB format

Receptor Molecule: (PDB:chainId e.g. 2kai:AB) or upload file:

Ligand Molecule: (PDB:chainId e.g. 2kai:1) or upload file:

e-mail address: (the results are sent to this address)

Clustering RMSD:

Complex Type: Be sure to give receptor and ligand in the corresponding order!

Advanced Options:
[\[Show\]](#)[\[Hide\]](#)

Receptor Binding Site: upload receptor binding site file

Ligand Binding Site: upload ligand binding site file

Distance Constraints: upload distance constraints file


[FireDock](#) - Fast Interaction Refinement in Molecular Docking
[SymmDock](#) - An Algorithm for Prediction of Complexes with C_n Symmetry

Figure 3.9. The PatchDock user interface: The receptor molecule and the ligand molecule are given either by the PDB code of the molecule (chain IDs are optional) or by uploading a file in PDB format. Taken from [418].

Output

Top 20 solutions are produced automatically from the PatchDock website. The user

receives an email containing the link to this page (<https://bioinfo3d.cs.tau.ac.il/PatchDock/>). Usually, the best 20 solutions are shown in a table or row. The values of parameters such the geometrical shape complementarity score, the size of the interface region, the desolvation energy, and the actual rigid transformation are displayed along with the solutions. A URL is provided alongside the PDB file for each docking solution. The user can see or download it. There is a button labelled "next 20 solutions" in the table's lower right corner that may be used to examine the lower-scoring solutions as well. On the solution page, there is an opportunity to download the top-scoring answers. The best solutions are available for download as a compressed ZIP file. The top-scoring solutions' PDB files are contained in the compressed file that was downloaded. The user can choose their own solution numbers, however there is a limit of 100.



Molecular Docking Algorithm Based on Shape Complementarity Principles
[\[About PatchDock\]](#) [\[Web Server\]](#) [\[Download\]](#) [\[Help\]](#) [\[FAQ\]](#) [\[References\]](#)

Receptor	Ligand	Complex Type	Clustering RMSD	User e-mail	Receptor Site	Ligand Site	Distance Constraints
MDM2NTD.pdb	Idasanutlin.pdb	drug	1.5	p.das.mbbt@gmail.com	-	-	-

Solution No	Score	Area	ACE	Transformation	PDB file of the complex
1	5302	633.90	-270.14	0.16 1.24 -1.46 23.47 -21.01 -13.28	result_1.pdb
2	5276	581.70	-234.49	0.71 -0.00 3.06 30.96 -21.01 -6.52	result_2.pdb
3	5224	537.70	-250.05	0.17 -1.20 -1.63 22.38 -20.92 -0.31	result_3.pdb
4	5120	646.10	-294.10	0.76 1.04 -1.87 25.61 -20.21 -14.50	result_4.pdb
5	5002	605.10	-221.08	1.28 1.11 -2.60 26.94 -21.77 -14.91	result_5.pdb
6	4868	597.60	-266.54	1.49 0.16 2.17 31.35 -27.68 -11.39	result_6.pdb
7	4812	580.10	-293.09	2.99 -0.08 -3.02 34.77 -25.39 -8.75	result_7.pdb
8	4810	579.30	-277.39	1.93 -0.68 1.40 23.50 -30.10 -4.12	result_8.pdb
9	4676	609.40	-256.24	1.13 -0.70 0.93 23.25 -29.34 -3.87	result_9.pdb
10	4668	567.90	-222.52	3.12 -0.28 -1.81 29.95 -19.69 -4.44	result_10.pdb
11	4656	543.40	-167.37	0.73 0.16 2.97 30.14 -22.68 -7.17	result_11.pdb
12	4640	531.70	-260.84	2.74 -0.38 -2.77 33.18 -22.85 -6.87	result_12.pdb
13	4626	564.90	-234.75	1.06 0.87 -2.17 25.95 -19.92 -13.88	result_13.pdb
14	4566	552.20	-202.79	0.78 -0.96 0.69 21.86 -28.56 -0.57	result_14.pdb
15	4550	682.70	-335.98	-1.78 -1.17 0.66 22.00 -22.21 4.83	result_15.pdb
16	4536	609.20	-329.01	1.51 -0.40 1.56 25.97 -29.63 -5.94	result_16.pdb
17	4536	580.20	-189.03	0.65 -0.00 2.77 29.88 -23.06 -5.30	result_17.pdb
18	4522	633.30	-316.14	-2.99 -0.11 -1.93 30.40 -18.78 -4.64	result_18.pdb
19	4496	558.60	-219.12	0.86 -0.14 2.79 31.39 -22.37 -6.85	result_19.pdb
20	4482	636.80	-327.10	2.28 -1.05 1.16 24.69 -27.53 -4.03	result_20.pdb

[show next 20 >>](#)

[NEW: Jmol view](#)

DOWNLOAD best solutions as a ZIP file: (solutions number, from 2 to 100) (this takes few seconds, please wait patiently)

DOWNLOAD [solutions table](#) [transformations file](#)

REFINE best solutions with FireDock: (solutions number, from 1 to 1000)

Figure 3.10. The solutions page presents the geometric score, interface area size and desolvation energy of the 20 top scoring solutions. Taken from [418].

FireDock is a web server for rapid molecular docking interaction refinement, and it can be used to further refine the structures acquired via PatchDock.

3.1.3.2. ClusPro web server:

In 2004, ClusPro, a web-based server, was first released [432, 433]. Since then, it has undergone significant modification and expansion [434-436]. Using ClusPro, two interacting proteins may be docked directly [437]. The server requires two PDB-format protein files in order to perform docking. The server executes three computational steps when docking:

- (i) Utilizing rigid body docking to sample billions of conformations.
- (ii) In order to identify the biggest clusters that reflect the most plausible models of the complex, 1000 lowest energy structures were clustered based on root-mean-square deviation (RMSD).
- (iii) Using energy minimization to improve the chosen structures (**Figure 3.11**). The Fast Fourier Transform (FFT) correlation technique is used by PIPER [438], a docking tool, during the rigid body docking stage.

Now, the ClusPro web server has been updated to the version ClusPro 2.0.

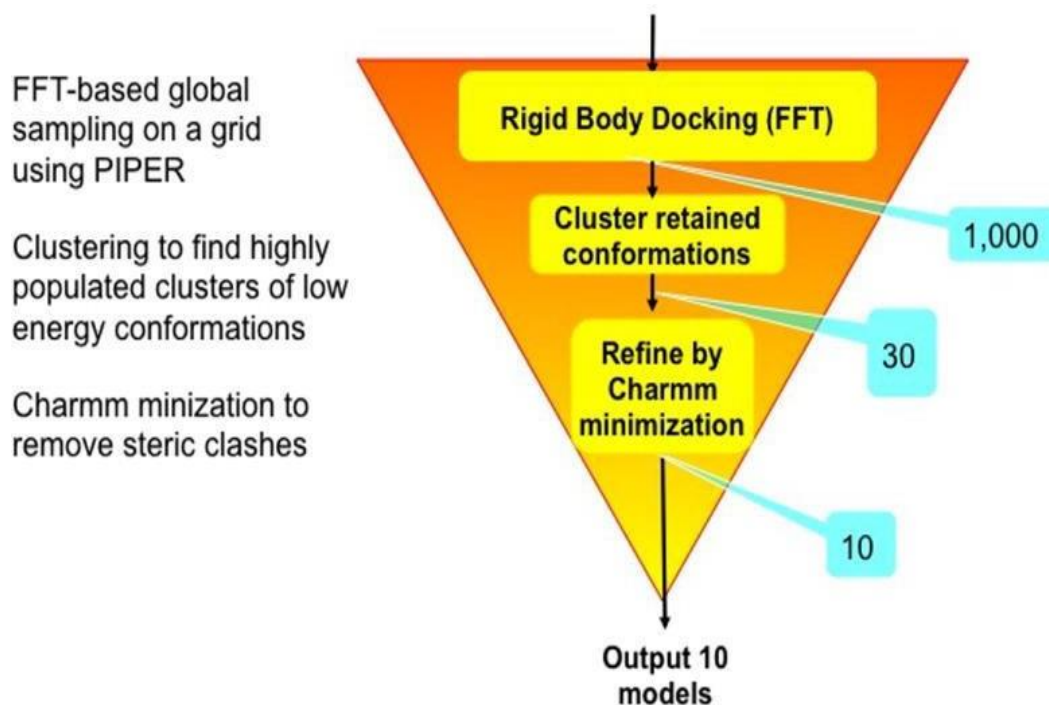


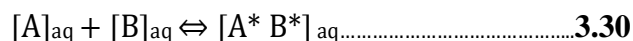
Figure 3.11. Representation of the ClusPro algorithm, the number of structures retained after each step is shown in a blue box. Taken from [437].

3.1.4. Binding free energy calculation using Molecular Mechanics energies combined with the Poisson-Boltzmann or Generalized Born and Surface Area continuum solvation method (MM- PBSA/GBSA):

3.1.4.1. Free energy calculation using Perl Script (mm_pbsa.pl):

To calculate the BFE of tiny ligands (small molecules) to receptor proteins or protein-protein complexes, researchers frequently use the MM-PBSA and MM-GBSA techniques [439–441]. They are frequently reproducible with excellent precision and are generally based upon MD simulations of the protein-ligand complex.

In simple terms, to calculate the absolute BFE for a complex made of two molecules: A and B bonded non-covalently:



where $[A]_{aq}$ refers to the molecule A dynamical structure free in the solution, $[B]_{aq}$ refers to the molecule B dynamical structure free in the solution, and $[A^* B^*]_{aq}$ refers to the complex formed by A and B molecules; the BFE is calculated using the second law of thermodynamics:

$$\Delta G = \Delta H - T\Delta S \dots \dots \dots 3.31$$

wherein, ΔH is the enthalpy, ΔS represents entropy and T is the temperature of the system at 300 Kelvin.

In MM-PBSA or MM-GBSA, the BFE ($\Delta G_{bind}/\Delta G_{binding}$) between a receptor and a ligand to form a protein-ligand complex is calculated as:

$$\Delta G_{bind}/\Delta G_{binding} = \Delta G_{complex,solv} - (\Delta G_{protein,solv} + \Delta G_{ligand,solv}) \dots \dots \dots 3.32$$

where, $\Delta G_{complex,solv}$, $\Delta G_{protein,solv}$, and $\Delta G_{ligand,solv}$ represent the differences in free energy for the complex, the protein, and the ligand, respectively, with or without solvent.

$$\Delta G_{bind}/\Delta G_{binding} = [E_{MM} + \Delta G_{solvation}] - T\Delta S_{total} \dots \dots \dots 3.33$$

$$E_{MM} = E_{intra} + E_{elec} + E_{vdW} \dots\dots\dots 3.34$$

$$E_{internal} = E_{bond} + E_{angle} - E_{torsion} \dots\dots\dots 3.35$$

$$\Delta G_{solvation} = \Delta G_{PB/GB\ solvation-elec} + \Delta G_{SASA,nonpolar} \dots\dots\dots 3.36$$

E_{MM} is the molecular mechanics (MM) energy from the force field without the solvent. $E_{internal}$ (intra) consists of three intramolecular contributions, i.e. E_{bond} , E_{angle} , and $E_{torsion}$. E_{elec} and E_{vdW} are the intermolecular electrostatic and van der Waals interaction energies, respectively. $\Delta G_{solvation}$ is the solvation free energy, and $\Delta G_{solvation-elec}$ is estimated from the Poisson–Boltzmann method. $\Delta G_{nonpolar}$ is estimated from the solvent-accessible surface area (SASA). T and S_{solute} are the temperature and the entropy of a solute. We show the relationship for each energy in **Figure 3.12**.

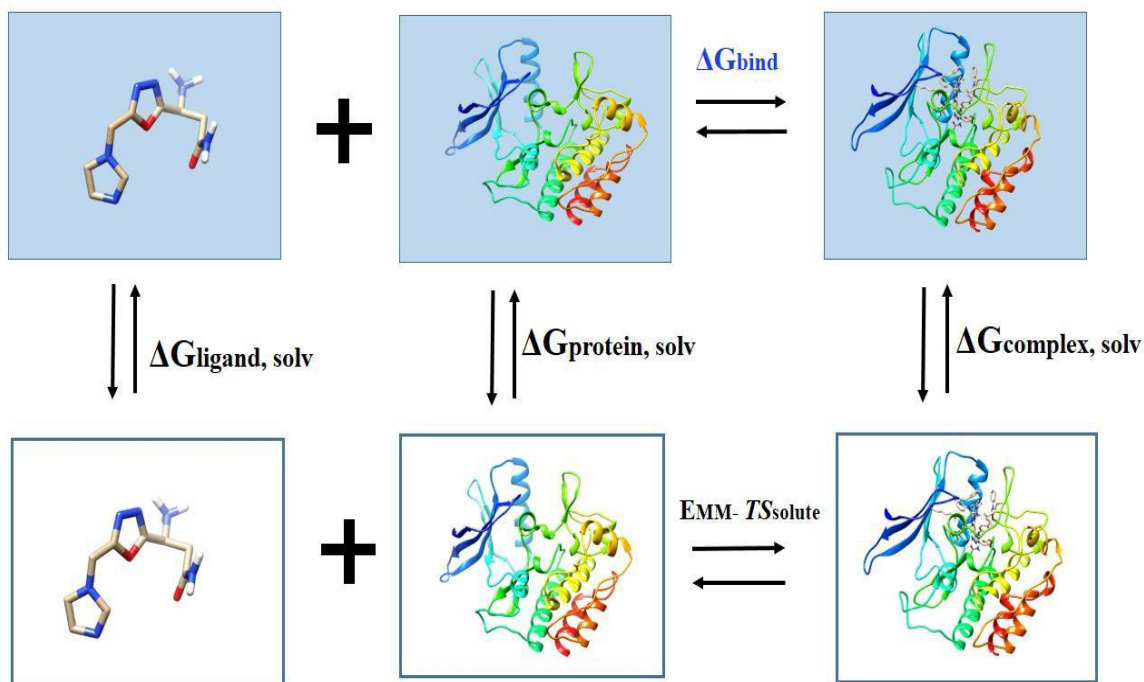


Figure 3.12. Computational schemes of the binding free energies based on MM-PBSA/GBSA. The free energies colored in black are directly calculated, while the free energy of interest colored in blue is indirectly did using the thermodynamic cycle of other free energies. Modified from [442].

It is possible to calculate the electrostatic solvation energy using the PB and GB approach. For the inner (solute) and outside (water), the dielectric constants were

adjusted to 1 and 80, respectively. Atomic charges and radii are the same as those employed in MD simulations. From solvent accessible surface-area, the non-polar contribution (ΔG_{SASA}) to the solvation free energy was calculated by means of eqn. 3.37.

$$\Delta G_{SASA} = \gamma \times SASA + b \dots \dots \dots 3.37$$

Here, SASA is the solvent-accessible surface-area and γ is surface tension parameter. ' γ ' is set as 0.005 kcal ($\text{mol}^{-1}\text{\AA}^{-2}$) for PB and 0.0072 kcal ($\text{mol}^{-1}\text{\AA}^{-2}$) for GB. ' b ' is a parameterized value set as 0.92 kcal mol^{-1} for PB and 0 kcal mol^{-1} for the GB method. The probe radius of the solvent is set to 1.4 \AA .

The total entropy (S), has been formulate from variations in the degree of freedom as shown in Equation 3.36:

$$S = S_{\text{trans}} + S_{\text{rot}} + S_{\text{vib}} \dots \dots \dots 3.38$$

where, S_{trans} is the translational, S_{rot} the rotational, and S_{vib} the vibrational entropy of each component.

3.1.4.2. Free energy decomposition using Python Script MMPBSA.py:

In accordance with the work of Gohlke *et al* [443, 444], AMBER14 includes a number of strategies to deconstruct estimated free energy into particular residue contributions using either the GB or PB models. It is possible to deconstruct interactions for each residue by only taking into account interactions in which at least one of the residue's atoms is active. This process is known as per-residue energy decomposition. The opposite method, known as pairwise decomposition, allows interactions to be broken down by particular pairs of studied residues by only include interactions wherein one atom from each of those pairings is involved. With regard to crucial interactions during free energy calculations, these decomposition approaches can offer helpful insights [443, 444]. The dielectric border between the protein and the bulk solvent is essentially nonlocal and depends on how all the atoms are arranged in space, hence it is crucial to notice that solvation free energies utilizing GB as well as PB are not strictly pairwise

decomposable. As a result, results from free energy decomposition must be interpreted with caution.

Using the Per-Residue Decomposition Method via Python Script MMPBSA.py [445], we can determine the partial BFE contribution towards the amino acid residue Y (ΔG^Y bind). The contribution of each residue to the overall BFE may be calculated using a per-residue based decomposition method [446-449]. We divide the terms in Eqn. (3.37) first to get the ΔG^Y bind. into its atomic contribution, or (3.37). Equation 3.39 is used to calculate each atom's (*a*) contribution to the overall electrostatic interaction energy:

$$E_{\text{elec}}^a = \frac{1}{2} \sum_{b \neq a} \frac{q_a q_b}{r_{ab}} \dots \dots \dots \mathbf{3.39}$$

where q_a and q_b are atomic partial charge on the atoms *a* and *b*, and r_{ab} is the distance between them. To prevent duplicate counting, just use half of the pairwise energy for the van der Waals contact between the protein and the ligand, E_{vdw}^a . Equation 3.40 illustrates how the non-polar effects of solvents on BFE are expressed using each atom's (*a*) SASA.

$$\Delta G_{\text{nonpolar,solv}}^a = \gamma \{ (\text{SASA}^{a,\text{complex}} - (\text{SASA}^{a,\text{protein}} + \text{SASA}^{a,\text{ligand}})) \} \dots \dots \dots \mathbf{3.40}$$

Where, $\text{SASA}^{a, \text{protein}}$ and $\text{SASA}^{a,\text{ligand}}$ is equal to zero depending on which component the atom belong to. γ is set to $0.0072 \text{ kcal mol}^{-1} \text{ \AA}^{-2}$ in AMBER 14. The Generalized Born/Poisson–Boltzmann (GB/PB) approach is used to calculate the contribution of atom “*a*”, to the electrostatic part of solvent effects. The contribution of atom “*a*” is given by:

$$\Delta G_{\text{elec,solv}}^a = -\frac{1}{2} \sum_a \left(1 - \frac{e^{-k \int_{ab}^{GB}}}{\epsilon_\omega} \right) \frac{q_a q_b}{\int_{ab}^{GB}(r_{ab})} + \frac{1}{2} \sum_{b \neq a} \frac{q_a q_b}{r_{ab}} \dots \dots \dots \mathbf{3.41}$$

$$\int_{ab}^{GB} = \left[r_{ab}^2 + \alpha_a \alpha_b \exp\left(\frac{-r_{ab}^2}{4\alpha_a \alpha_b}\right) \right]^{1/2} \dots \dots \dots \mathbf{3.42}$$

where κ is the Debye-Huckel screening parameter. ϵ_{ω} is a dielectric constant for the solvent set as 80. α_a and α_b are the effective Born radii of atoms a and b , respectively. Evaluating the partial BFE contribution to amino acid residue Y using these contributions to each atom results in equation 3.34, which is shown below:

$$\Delta G_{\text{bind}}^Y = \sum_{a \in Y} (E_{\text{elec}}^a + E_{\text{vdw}}^a + \Delta G_{\text{nonpolar,solv}}^a + \Delta G_{\text{elec,solv}}^a) \dots \dots \dots \mathbf{3.43}$$

In this study, the entropic as well as intra-molecular contributions found in equations (3.25) and (3.26) are disregarded.

The introduction of particular mutations into the protein sequence, along with examination of the effects on BFE or stabilities are two more methods of deconstructing free energies [450]. One method for highlighting the significance of the electrostatic plus the steric properties of the original side chain is known as alanine scanning mutagenesis, which involves changing one amino acid in the system to alanine [451]. We may immediately incorporate the mutation into each component of the original ensemble under the presumption that it won't significantly affect protein structure. By doing this, generating an ensemble for the mutant may be accomplished without running a separate MD simulation.

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

Protein-protein interactions (PPIs) are an important mechanism that drives a variety of cellular physiological functions and are also implicated in the pathophysiology of many illnesses [452-454]. It is important to carefully examine the characteristics of the protein interface since protein-protein interactions can vary greatly. A crucial factor in protein-protein interactions is the one that determines their stability and specificity. If the combination is temporary or necessary, it depends on how large the protein contact is. There is normally a 750–1500 Å² surface area submerged in each protein at the interface between two proteins, which has a typical area of 1500–3000 Å² [455-457]. Protein-protein interaction sites are created by proteins that have strong shape complementarity

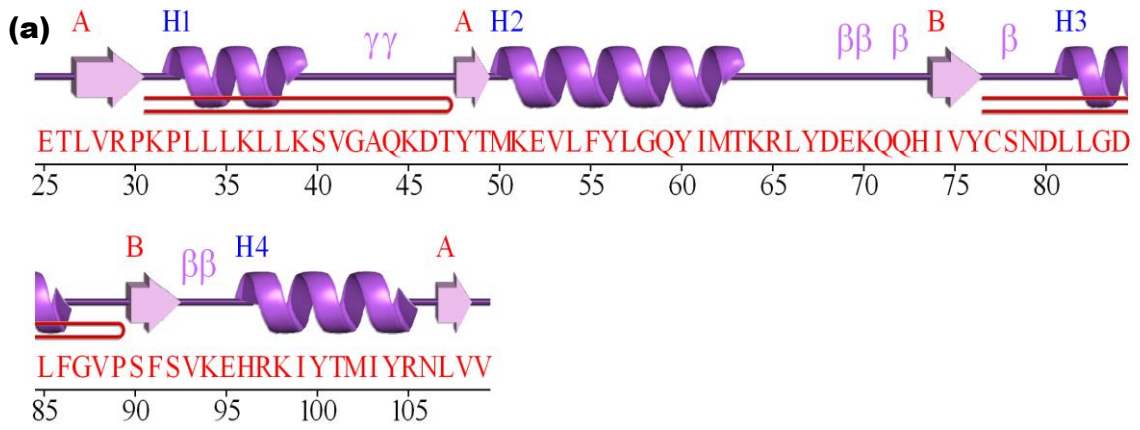
[458-460] and are propelled by hydrophobic effects [461], which happen through van der Waals interactions between the nonpolar regions of protein residues. The two proteins' interacting protein surfaces' electrostatic complementarity encourages the development and longevity of the complex. Hydrogen bonds and electrostatic interactions are important factors at some interfaces that influence how one protein docks with another's binding site. In order to find new drugs, protein-protein interaction prediction is essential. Many biological functions, both healthy and unhealthy, depend on the interaction between proteins, which can be hampered by outside substances. Two primary processes make up the contemporary drug discovery process: choosing a potential drug target, learning more about it, and creating a matching ligand [462]. As a result, understanding protein-protein interactions can help when creating modulators that specifically target protein complexes.

3.1.5.1. PDBsum web server:

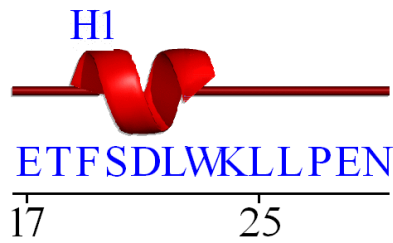
PDBsum is a web-based database that gives a visual overview of the crucial details about each macromolecular structure submitted to the Protein Data Bank (PDB) (<http://www.ebi.ac.uk/pdbsum>) [463]. Comprehensive structural analyses, annotated plots of the secondary structure of each protein chain, detailed structural photographs, a summary of the PROCHECK results, as well as schematic diagrams of protein-protein, protein-small molecule, and protein-DNA interactions are all included. RasMol scripts draw attention to crucial structural components such the protein's domains, PROSITE patterns, and protein-protein/ligand interactions. PDBsum, which is publicly available at <http://www.biochem.ucl.ac.uk/bsm/pdbsum>, is updated anytime new structures are made available by the PDB.

3.1.5.1.1. Wiring diagram:

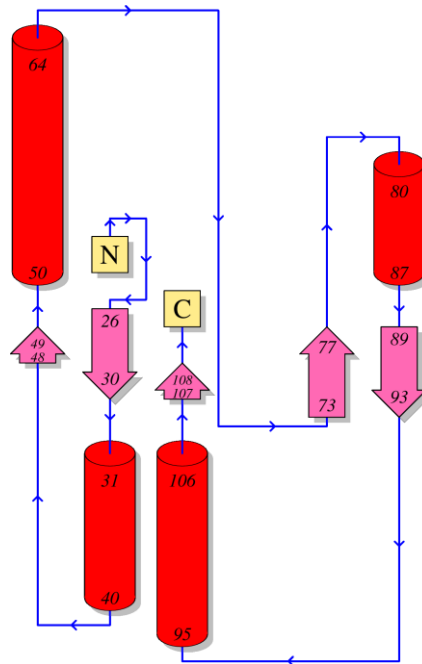
For each unique protein chain of a structural model, PDBsum provides a 'protein page' that includes a schematic diagram of the protein's secondary structure that is 'wiring diagrams' (Figure 3.13a and 3.13b).



(b)



(c)



(d)

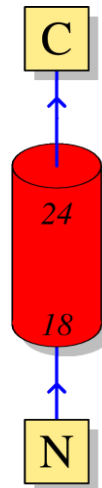


Figure 3.13. Schematic diagrams from the PDBsum for entry *1ycr*: The ‘wiring diagram’ shows the protein's secondary structure elements (α -helices and β -sheets) together with β - and γ -turns, and β -hairpins in (a) MDM2(NTD), and (b) p53(TAD1). Topology diagram illustrates the β -strands by the large arrows joined side-by-side, forming central β -sheet. The α -helices represented by the red cylinders. The small arrows indicate the directionality of the protein chain, from the N- to the C-terminus in (c) MDM2(NTD), and (d) p53(TAD1). Taken from [464, 465].

3.1.5.1.2. Topology diagram:

The protein page also features a topology diagram showing how the protein's helices as well as strands are connected and arranged (**Figure 3.13c** and **3.13d**). When a protein chain has many domains, each domain's diagram is created separately and color-coded in accordance with the wiring diagram's domain shading. With Gail Hutchinson's HERA software, topology diagrams are produced from hydrogen bonding plots [466].

3.1.5.1.3. Protein-protein interfaces:

Another novel feature in PDBsum shows how interactions occur across protein-protein interfaces. When a protein-protein complex has multiple protein chains (such as in **Figure 3.14a**), the interfaces between the chains are being shown using three different types of plots: the first plot summarizes an overview of which chains interact with which (**Figure 3.14b**), the second plot summarizes the interactions across any chosen interface (**Figure 3.14c**), and the third plot illustrates which residues are actually interacting

across that interface in depth information (Figure 3.14d).

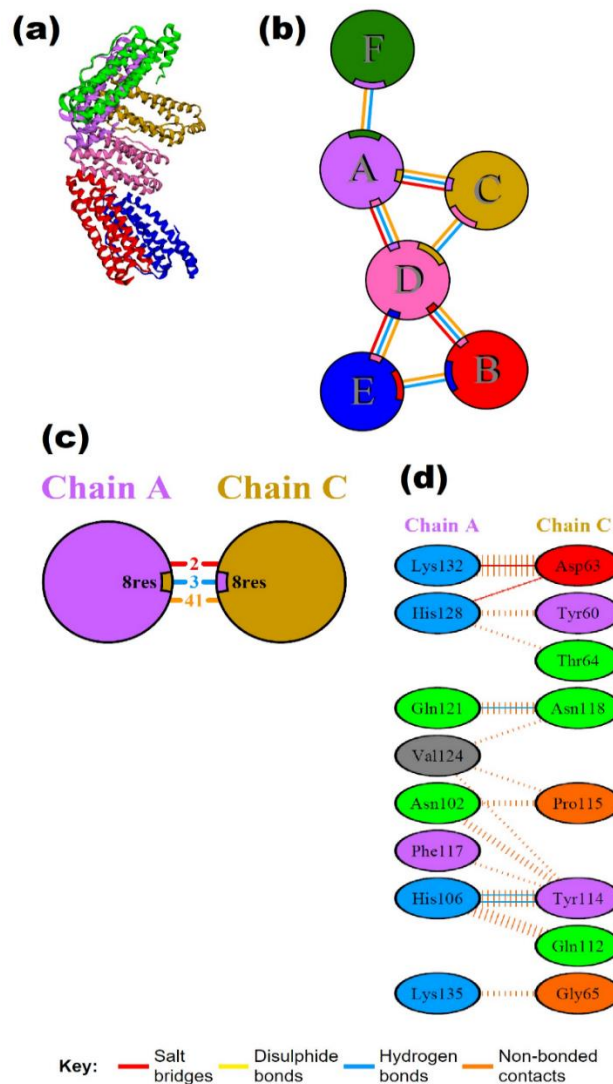


Figure 3.14 Protein-protein interaction diagrams in PDBsum for PDB entry *1eum*: (A) Thumbnail image of the 3D structural model which contains six protein chains (B) Schematic diagram showing the interactions between the chains. The area of each circle is proportional to the surface area of the corresponding protein chain. The joining lines are coloured light blue for hydrogen bonds and orange for non-bonded contacts. (C) A schematic diagram showing the numbers of interactions across one of the interfaces, namely the A-C protein interface, and the numbers of residues involved. (D) Detail of the individual residue-residue interactions across this interface. Hydrogen bonds (blue lines), non-bonded contacts (orange tick-marks), and salt bridges (red lines) between residues on either side of the protein-protein interface. Taken from [463].

The LIGPLOT tool [467] creates a 2-D graphic representation of the hydrogen bonds as well as non-bonded interactions between the protein residues with which the ligand interacts from the input of a protein-ligand complex in the PDB format (**Figure 3.15**). The LigPlot program has also a standalone version called LigPlot+, which can be installed and then can be used to generate the protein-ligand interaction profile. The result is a color or black-and-white PostScript (PS) file that presents the intermolecular interactions but also their intensities, including hydrophobic interactions, hydrogen bonds, and atom accessibilities. The software is entirely universal for any ligand. It may also be used to demonstrate different interactions between nucleic acids and proteins using the NUCPLOT tool present in the PDBsum web server.

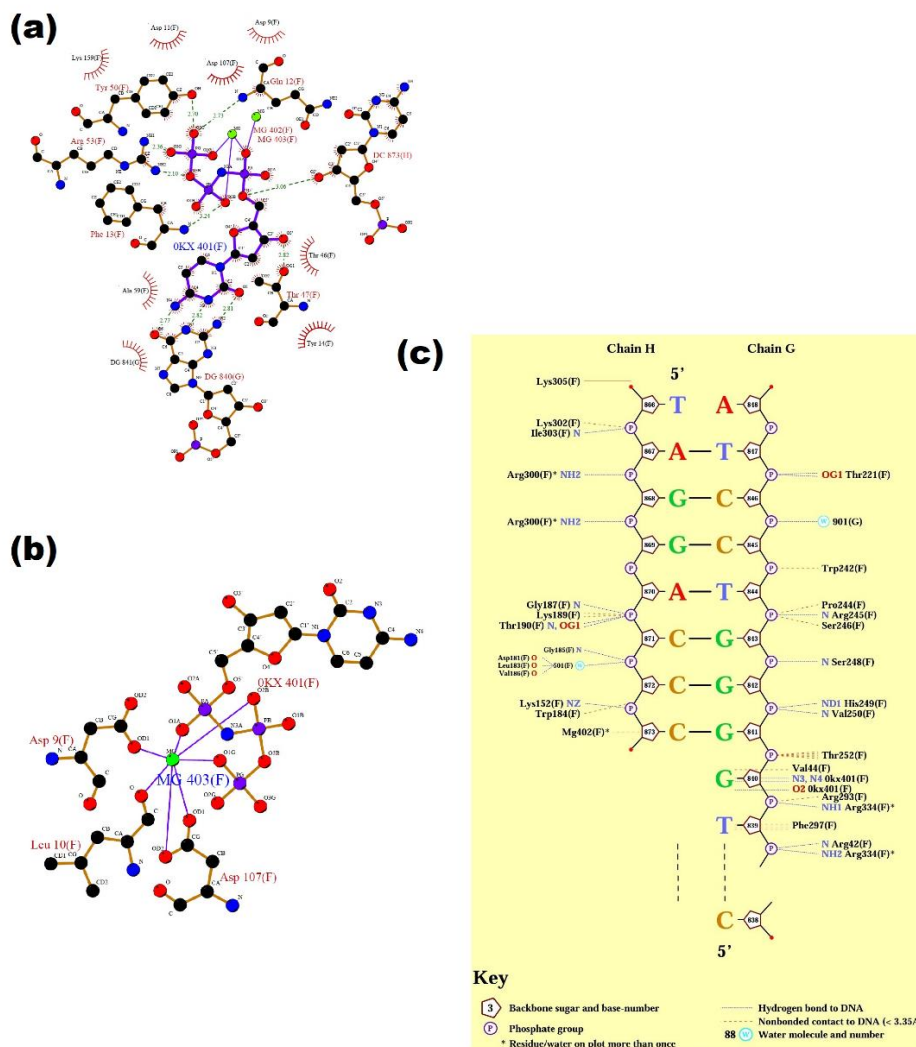


Figure 3.15. PDBsum interaction plot for PDB entry **6jun** in LIGPLOT and NUCPLOT: (a) LIGPLOT diagram showing the protein residues interacting with the ligand: 0KX (Formula: $C_9H_{17}N_4O_{12}P_3$; Chemical Name: 2'-Deoxy-5'-O-[(R)-Hydroxy{[(r)-Hydroxy(phosphonooxy)phosphoryl]amino}phosphoryl]cytidine), with hydrogen bonds shown by the green dashed lines and non-bonded contacts by the brown rays, and; (b) as in (a), but for residues interacting with the bound magnesium ion; (c) NUCPLOT diagram showing protein-DNA interactions, with H-bonds as blue dashed lines and non-bonded contacts as brown dashed lines. Taken from [463].

3.1.5.2. Hot spot residue prediction:

Protein-protein interactions are not entirely influenced by all the residues present on the protein-protein interface (PPIs). The significant proportion of the BFE is contributed by a small selection of residues known as hot spots [468]. A residue is considered to be a hot spot when the replacement of an alanine results in a significant rise in the BFE with a minimum value of $2.0 \text{ kcal mol}^{-1}$ [469]. On the other hand, null-spots (NS) insulate hot spots from solvent exposure and correspond to residues with (shift in BFE) $\Delta\Delta G_{\text{binding}}$ that are lower than $2.0 \text{ kcal mol}^{-1}$ when they are changed to alanine. [470]. Compared to all other interface residues present in the protein-protein complex, hot spot residues are well conserved, clustered, and more buried [471-474]. Leu, Thr, Ser, and Val amino acids are less prone to function as a hot spot, but Tyr, Arg, and Trp have a stronger chance to be a hot spot [475]. Similarly, Asp and Asn amino acids have been identified to be hotspots more often than amino acids Glu and Gln [475, 476].

We may be able to regulate protein-protein binding and better understand protein-protein interactions by identifying these hot spots within the protein-protein interfaces [477]. Various computational techniques, like KFC (Knowledge-based FADE and Contacts), DrugScorePPI, Robetta, and PredHS online servers, that are freely accessible, can be used to identify hot regions at protein-protein interfaces.

In order to identify hot spot residues within the protein-protein interfaces, the KFC server [478] uses *in silico* alanine scanning mutagenesis while taking into account hydrogen bonds, residue sizes, and atomic contacts.

DrugScorePPI [479] and Robetta [480, 481] online servers use *in silico* alanine scanning mutagenesis to determine the hot spot residues.

Using structural neighborhood-based techniques, the PredHS server predicts hot spot residues and then uses random forest as well as sequential backward elimination algorithms to choose the best features [482].

3.1.6. Analysis of trajectories:

The system's recorded coordinates and velocities are employed in this stage for additional analysis. The analysis requires MD trajectory files. When used with visualisation software (such as VMD), which may show the structural parameters of concern in a time-dependent manner, MD simulations can aid in the viewing and understanding of conformational changes at the atomic level. By means of the ptraj and cpptraj modules of AMBER14, parameters such as time average structure, Root Mean Square Deviation (RMSD), Root Mean Square Fluctuation (RMSF), and Radius of Gyration (Rg) can be analyzed. The secondary structure analysis can be carried out using the Dictionary of Secondary Structures in Proteins (DSSP) algorithm.

(i) Time average structure: This specific structure is generated by taking into account coordinate frames that have been averaged over slipping time windows of a specific size.

(ii) Root Mean Square Deviation (RMSD): RMSD is a measurement of a structure's deviation from a certain conformation. It is defined as:

$$\text{RMSD} = \left(\frac{\sum_N (R_i - R_i^0)^2}{N} \right)^{1/2} \dots\dots\dots 3.44$$

where N is the total number of atoms/residues involved in the calculation, and R_i is for the vector position of particle i (the target atom) in the snapshot, R_i^0 is the coordinate vector for the reference atom i . The initial frame of the (MD) simulation was used as the reference point for computing the RMSD using backbone atoms. In equation 3.30, the letter N stands for the total number of the variables needed to calculate, RMSD, which is the sum of number of the positions (i), number of the strands (j), and number of the angular parameters. The computed RMSD is indeed a radial vector in the structure

space that the absolute RMSD magnitude provides, and its length is r . According to the radial issue, there is more configurational space between a given radius and $r+dr$ the bigger the radius. At a greater r , the same RMSD value might capture both comparable and dissimilar structures. Additionally, crucial data for the comparison may be corrupted. Methods that rely on comparatively lower RMSD values provide more accurate measures of difference. Due to the molecule's inherent flexibility, another significant issue with the usage of RMSD arises whenever two or more structural substrates exist. Due to the molecule's inherent flexibility, another significant issue with the usage of RMSD arises whenever two or more structural substrates exist. However, one needs a technique that is accurate without compromising the information to represent the dynamical characteristics.

(iii) Root Mean Square Fluctuation (RMSF): The measurement of deviation between particle position i and a reference position is known as the root mean square fluctuation (RMSF):

$$\mathbf{RMSF} = \left(\frac{1}{T} \sum_{t=1}^T (r_i(t) - r_i^{ref})^2 \right)^{1/2} \dots\dots\dots 3.45$$

In *equation 3.36*, The time period over which one intends to average over is known as \mathbf{T} and \mathbf{r}^{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. $r_i^{ref} = r_i$.

Difference between RMSD and RMSF: In a molecular dynamics (MD) simulation, the spatial variations of the biomolecules are often measured using the **RMSD** (root mean square deviation) as well as **RMSF** (root mean square fluctuation) metrics. The RMSF is the fluctuation(s) around an average, per atom/residue/over a collection of structures (*such as*, from a trajectory), whereas the RMSD is the difference between two structures for a given set of atoms. It is quite feasible to have RMSD=0 along with a non-zero RMSF value for every atom or a substantial RMSD with a relatively small RMSF when there has been a significant conformational shift followed by very minor

fluctuations in atomic locations.

(iv) Radius of Gyration (R_g): The radius of gyration is computed to assess the structure's compactness.:

$$\mathbf{R}_g = \left(\frac{\sum_i |r_i|^2 m_i}{\sum_i m_i} \right)^{1/2} \dots\dots\dots 3.46$$

In *equation 3.37*, m_i is the mass of the atom i and r_i is the position of atom i with respect to the center of mass (CoM) of the molecule.

(v) Secondary Structure Analysis: Kabsch and Sander (1983) used a technique known as DSSP to determine the solvent accessibility of the residues and create a database of the ASA for the majority of the proteins stored in the PDB [483]. The ASA values for their use in prediction algorithms are often generated using this software [484-489]. One needs to visit <http://www.cmbi.kun.nl/gv/dssp/> to access it. It functions by categorizing protein secondary structures mostly according to backbone H-bonds. Additionally, it provides details on C α -pseudo dihedral and bond angles, of which just the latter is needed by DSSP to identify a residue's LSS. It is distinguished by the electrostatic hydrogen bond detection criteria. As a result, the elements of secondary structure are being assigned based on the distinctive hydrogen-bond patterns. This approach is extensively used as a gauge for secondary structure assignment. Many software applications employ DSSP to allocate secondary structures as necessary. A popular visualization tool like Rasmol, for instance, assigns repeating structures using a quick technique that is similar to the DSSP. Classification of helix or strands results from repeating patterns of the identical type of H-bonds, whereas classification of β -bridges results from non-repeating H-bonds [490]. Due to the fact that the relative orientations of nitrogen and oxygen atoms in the backbone are mirrored in the corresponding (ϕ , ψ) backbone torsion angles, residues with the identical secondary structure pattern are rather strongly grouped in a Ramachandran plot. On two levels, the DSSP provides details on the protein's secondary structure. The secondary structure from DSSP analysis is summarized by the one-character secondary structure information (1CSSI) code,

which at the upper level describes the LSS of a residue based mostly on the H-bond arrangement of the protein backbone into eight classes. The C-pseudo bond angle, which is the angle present between the vectors $C_{\alpha}(i) - C_{\alpha}(i-2)$ and $C_{\alpha}(i) - C_{\alpha}(i+2)$ for every residue, is instead determined using DSSP. If this angle is less than 110° and corresponds to a substantially bent geometry without distinguishing backbone H-bonds for the residues not allocated to a helix, or a strand, or a turn, the summary group S of bends is utilised. If none of the aforementioned requirements are met, DSSP creates a gap for improved discrimination, which we denote by the letter "C." However, these residues do not include backbone H-bonds necessary for the creation of secondary structures and instead belong to a region of the protein backbone structure that is generally straight.

3.1.7. 3-D structure visualization tools:

(i) **Visual molecular dynamics (VMD):** VMD is a computer application for molecular modelling and visualization [491]. The basic purpose of VMD is to see and evaluate the outcomes of MD simulations. Additionally, it has capabilities for working with arbitrary graphical objects, sequence data, and volumetric data.

(ii) **UCSF Chimera:** A very flexible tool called UCSF Chimera is used to interactively see and analyse molecular structures and associated information, such as conformational ensembles, density maps, sequence alignments, supramolecular assemblies, and docking outcomes [492]. The Resource for Biocomputing, Visualization, and Informatics (RBVI), with assistance from the National Institutes of Health (NIH), have developed Chimera.

(iii) **ArgusLab:** ArgusLab is a molecular modeling, graphics, and drug design program for Windows operating systems. ArgusLab is developed by Mark Thompson, a Research Scientist for the Department of Energy at Pacific Northwest National Laboratory [493].

3.1.8. 3-D structure modelling tools:

3.1.8.1. I-TASSER web server:

According to the Critical Assessment of protein Structure Prediction (CASP) ranking, I-TASSER has been reported to be the best automated 3-D structure of protein prediction server. Five models are obtained as results. The best model can be identified based on the C-score calculated from the relative clustering structural density and consensus significance. Template modelling (TM) score as well as root mean square deviation (RMSD) are used to evaluate the accuracy of the best model [494].