CHAPTER 2

Computational methods

Computational methods

Tools and Techniques

We employed various computational methods and approaches to explore XPA's prominent structural properties and dynamics. We adapted the molecular dynamics (MD) simulation approach to achieve our goal. The following diagram depicts the basic idea behind the molecular dynamics (MD) simulation method:

2.1 Molecular dynamics (MD) simulations

2.1.1 Brief historical background

In late 1957, Alder and Wainwright were the first to present the Molecular Dynamics (MD) method [226], where they put forth this idea to study the interactions of hard spheres. They were able to study the behavior of simple liquids. Rahman (1964) carried out the first MD simulation using liquid argon in a realistic system [227], followed by Stillinger and Rahman (1974), who used MD simulation to study the realistic system of liquid water [228]. In 1977, McCammon and his team conducted the first MD simulations of bovine pancreatic trypsin inhibitor (BPTI) [229]. After many breakthroughs in computer technology and algorithmic improvements, today we can easily study all types of biological systems, be it their energetics, protein folding mechanisms, etc. Nowadays, MD simulation is parallelly studied along with their experimental counterparts viz. nuclear magnetic resonance (NMR) and X-ray crystallography [230-233].

2.1.2 Theory of molecular dynamics (MD) simulation

MD simulations are a bridge between theoretical and experimental studies. This particular study emphasizes the elucidation of the minute microscopic details which are otherwise not seen to the naked eyes, which include bonded and non-bonded interactions, the location of individual atoms, and their velocities, etc. these minute details can be understood and converted into understandable information viz. their energetics, heat, and pressure tolerances, etc., that can, in turn, help study the biological systems. MD simulations help mimic the physical atomic movements present in a protein, DNA, or any other biological molecule in an actual environment. Here, the atoms interact with each other for a particular period, and these movements of atoms as a function of time are then studied in detail [234]. The role of solvent here is pivotal to regulating the protein's internal motion concerning its structural conformations at various temperatures and, specifically, below the transition of glass temperature since it is unfeasible to capture these dynamics experimentally [235-238]. In the past few decades, there has been growth in modified MD simulation packages such as AMBER [239-242], GROMACS [243, 244], and NAMD [245, 246], which have distinctly improved their performance, with the workload of up to ~10-100 ns/day/workstation/cluster.

MD simulation provides an alternate route to understanding the protein dynamics to calculate the residual dipolar coupling of proteins at NMR relaxation time scales s [230-233]. This residual dipolar coupling provides us with information regarding the relative orientation of the structures present in the protein in pico and nanoseconds. The major distinction between MD simulations and NMR spectroscopy helps us to understand the disparity between experimental and theoretical results [234], as well as enhances the force field's quality related to the simulation-cum-integration methods. In the recent past, numerous alternative methods of classical MD simulations have been developed like steered MD [247-250], Monte-Carlo sampling of conformational space [251-254], hybrid Quantum Mechanics/ Molecular Mechanics (QM/MM) [255-258], Brownian dynamics [259-261], coarse-grained (CG) dynamics [262-264], molecular docking simulations [265-267] and normal vibration modes analysis [268-270], all of which are equally important in studying the simulations of biomolecules.

Furthermore, using MD simulation techniques, changes in the energetics of different biomolecules and their mechanisms of conformational changes, thermodynamic properties, and time-dependent (kinetic) phenomena can be studied [271-275].

The major principle of MD simulation is based on Newton's second law of motion [276-279], where the mass 'm' of interacting particles/atoms/systems, and their initial positions and velocities with an accurate description of their potential energy are studied as a function of time. It states that the positions and velocities of individual particles in a particular system vary with time in phase space, otherwise known as trajectories.

The trajectory is attained by solving the differential equation of Newton's II law. The force 'F' acts on a particle with mass 'm' of the particles, and acceleration 'a' of the particle. And these are derived from the potential energy μ(r^{N}), where $r^{N} = (r^{1}, r^{2} \ldots r^{N})$, and is the entire set of 3N atomic coordinates.

$$\vec{F} = ma \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.1$$

$$F = -\frac{d}{dr}\mu \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.2$$

The main aim of this equation is to define the position 'r_i (t+Δt)' at a time 't+Δt' in terms of the already known positions at a time 't'. In MD simulations, the Verlet algorithm, which uses Taylor series expansions of the positions and dynamic properties, is generally preferred to calculate the trajectories of particles due to its time-reversibility, simplicity, and numerical stability. The only variation in the Verlet algorithm is the leap-frog algorithm [149], where velocities can be calculated from the positions explicitly.

The leap-frog algorithm uses velocities at half time step

$$\dot{r}_i\left(t + \frac{\Delta t}{2}\right) = \dot{r}_i\left(t - \frac{\Delta t}{2}\right) + \ddot{r}_i(t)\Delta t \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.3$$

The velocities can be computed from the following formula at time 't':

$$\dot{r}_\iota\,(t) = \frac{\dot{r}_\iota\left(t+\frac{\Delta t}{2}\right)+\dot{r}_\iota(t-\frac{\Delta t}{2})}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.4$$

This is useful when the kinetic energy is needed at a time 't', e.g. in the case, where velocity rescaling must be carried out. The atomic positions are then obtained from:

$$r_i\ (t+\Delta t) = r_i\ (t) + \dot{r}_i(t + \frac{\Delta t}{2})\ \Delta t \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.5$$

This particular algorithm is computationally economic, taking less time and requiring less storage, due to which it is used for large-scale calculations with conservation of energy, even at large time steps. However, the Predictor-Corrector algorithm comes in handy, when more accurate velocities and positions are needed.

The molecular trajectory theoretically mimics the motion of a real system, providing the approximate potential energy function between the particles. This describes both the dynamics as well as equilibrium properties of the system under consideration., where the functional form of the potential energy function used with the set of interaction parameters is called a *force field*.

2.1.3. Force field

The force fields provide in-depth information regarding the potential energy of a particular particle. Force field parameters are usually obtained from experimental quantum mechanical studies of small molecules, which may be transferred to usually desired larger molecules. These include bonded and non-bonded interaction, wherein the former interactions are harmonic oscillator energy with regards to their bond angles, bond lengths, and at times, the improper dihedrals and torsional dihedral, while the latter term consists of electrostatic and van der Waals interactions, described as the Lennard-Jones (LJ) [280, 281] potential function. LJ functions are dispersion or London interactions of transient dipoles, whereas Coulomb potentials describe the electrostatic interactions. So far, numerous research groups have developed various kinds of force fields to study the systems, such as AMBER03 [282], AMBER99SB [283], CHARMM22 [284], OPLS-AA [285], GROMOS87 [286], General Amber force field (GAFF) [287].

The typical functional form of a force field is:

$$V(r^N) = \sum_{bonds} \frac{k_i}{2}(l_i - l_{i,o})^2 + \sum_{angles} \frac{k_i}{2}(\theta_i - \theta_{i,o})^2 + \sum_{torsions} \frac{V_n}{2}(1 + \cos(n\emptyset - \emptyset_o)) + \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\epsilon_o\varepsilon_r r_{ij}}\right) \quad \ldots\ldots 2.6$$

Where,

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

k_i : force constant;

l ,l_0 : current and reference bond lengths;

θ, θ_0 : current and reference valence angle:

V_n : barrier height of rotation;

$\emptyset$: torsion angle;

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

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

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

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

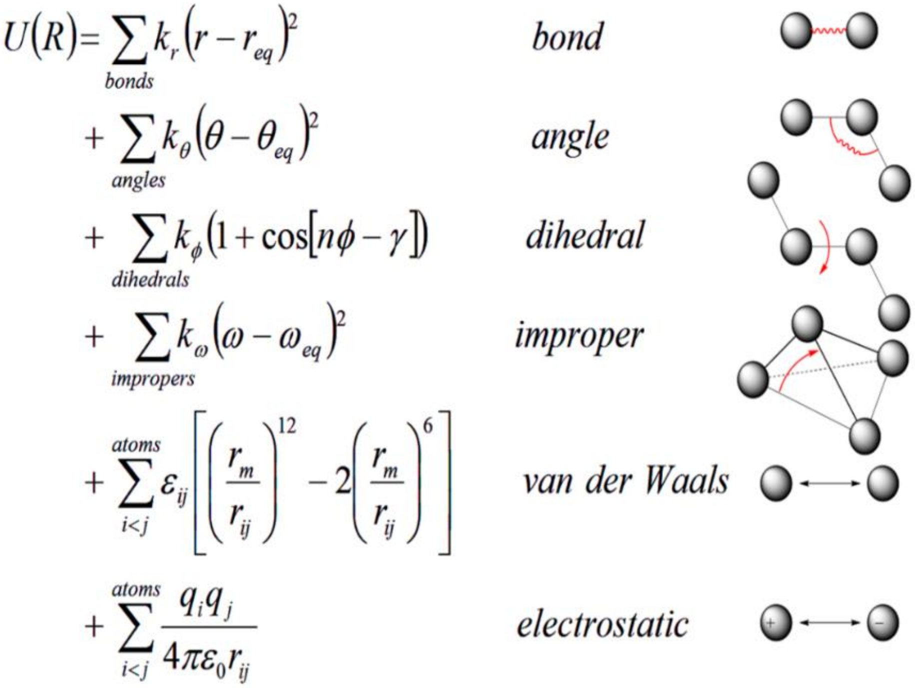
r_{ij} : current distance between the atoms i and j;

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

$\emptyset_0$: phase factor that determines where the torsion angle passes through its energy minima.

The different kinds of molecular interactions and their description representing potential energies as a function of time are given below in **Figure 2.1.**

i. Bond stretching between covalently bonded atoms.
ii. Angle bending due to vibrational motions requires less energy to distort an angle from its equilibrium value.
iii. Improper dihedral angles.
iv. Improper torsion angle shows changes in energies due to bond rotation.
v. Lennard-Jones potential for van der Waals interaction.
vi. Coulomb potential for electrostatic interactions.



$$U(R)=\sum_{bonds} k_r\left(r-r_{eq}\right)^2 + \sum_{angles} k_\theta\left(\theta-\theta_{eq}\right)^2 + \sum_{dihedrals} k_\phi\left(1+\cos[n\phi-\gamma]\right) + \sum_{impropers} k_\omega\left(\omega-\omega_{eq}\right)^2 + \sum_{i<j}^{atoms} \varepsilon_{ij}\left[\left(\frac{r_m}{r_{ij}}\right)^{12} - 2\left(\frac{r_m}{r_{ij}}\right)^{6}\right] + \sum_{i<j}^{atoms} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}$$

Figure 2.1 Molecular interactions represents potential energies as a function of time for MD simulation. Taken from [162]

2.1.4. Periodic boundary conditions

Many atoms experience a large boundary surface to a vacuum environment while simulating, which is irrelevant to studying phenomena taking place in bulk. Periodic boundary conditions (PBC) make it possible for small particles to experience forces if they are suspended in bulk solution [288-290]. The atoms are placed in a simulation box that is surrounded by translated copies of the coordinates of the atom as shown in **Figure 2.2.** A periodic 3D array surrounds the inner cell. If an atom crosses the boundary, it is replaced by a mirror image atom that enters from the opposite side with unchanged velocity. Thus, the number of particles within the central box remains constant. A non-bonded cutoff is used to deal with the non-bonded interactions such that each atom interacts with only one image of every other atom in the system.

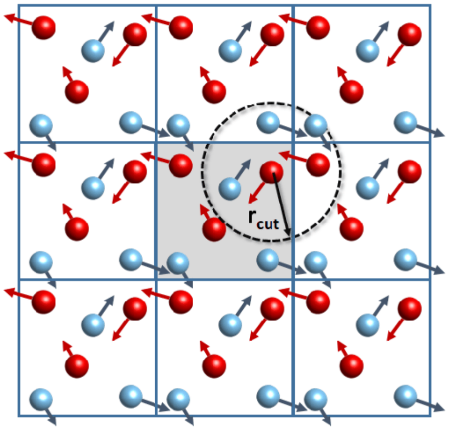


Figure 2.2. Periodic boundary conditions in two dimensions. Taken from [289].

2.1.5. Long-range range interactions Ewald sum

For treating a series of long-range interactions in a periodic system, one of the most used strategies is Ewald summation, which accurately deciphers all the effects of long-range interactions in a computer-modulated simulation. A particle in this simulation box interacts with other particles in the simulation box as well as all of their images in an endless array of periodic cells. The main idea behind the Ewald sum is to analyze a charge distribution with an opposite sign charge site, which screens interactions between neighboring atoms. [291, 292].

The infinite range Coulomb interaction under periodic boundary conditions is easily calculated using this method (PBC). The Particle Mesh Ewald (PME) modification uses the three-dimensional fast Fourier transform to accelerate the Ewald reciprocal sum to near-linear scaling (3DFFT). Because the coulombic interaction has an infinite range, particle I in the unit cell interacts electrostatically with all other particles j in the cell, as well as all of the j's periodic pictures, and it also interacts with all of its periodic images under PBC. The total Coulomb energy of a system of N particles in a cubic box of size L and their infinite duplicates in PBC is calculated as follows:

$$\mathrm{U} = \frac{1}{2}\sum_{n}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i\,q_j}{r_{ij,n}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.7$$

Ewald recast the potential energy of Eq. (2.7), a single slowly and conditionally convergent series, into the sum of two rapidly converging series plus a constant term,

$U_{Ewald} = U^{r} + U^{m} + U^{o}$ …………………………………………………………2.8

The real (direct) space sum (Ur), the reciprocal (imaginary, or Fourier) sum (Um), and the constant term (U°), also known as the self-term, make up the Ewald sum. Ewald sum is increasingly being applied to other systems where electrostatic effects are important, such as lipid bilayers, proteins, and DNA, in simulations that involve highly charged systems.

2.1.6. SHAKE algorithm

Due to the varied time scales associated with vibrational degrees of freedom including bond vibration, angle stretching, and torsional mode, the choice of time step in a molecular system is limited. The fastest vibrational mode is usually seen in bonds involving hydrogen atoms, which limits the time step of integration to 1 fs. These rapid degrees of freedom can be restrained while solving the unconstrained degrees of freedom, allowing for a bigger time step. Because hydrogen bonds have the highest frequency, they can be restricted during dynamics via Ryckaert and team's SHAKE algorithm. [293, 294].

SHAKE's main idea is to employ the Lagrange multiplier formalism to keep bond distances constant. Assume we have *N*c such constraints, as defined by

$\propto_k = r^2{}_{k_1\,k2} - R^2{}_{k1\,k2} = 0,$ where $k = 1, 2, 3$……*N*c……………..…… …. 2.9

R_{k1k2} being constrained distant between atoms k_1 and k_2 atoms. This leads to a modified constrained equation of motion

$m_i \frac{d^2 r_i\ (t)}{dt^2} = -\frac{\partial}{\partial_{ri}}\left[V\ (r_1\ \ldots\ldots\ r_N) + \sum_{k=1}^{N_c} \tau_k\ (t)\alpha_k(r_1\ \ldots .\ r_N)\right]$……………2.10

Where m_i is mass of i^{th} particle and τ_k is the Lagrange multiplier (unknown) for

k^{th} constraint. This equation can be solved for an unknown multiplier by solving N_c quadratic coupled equations. And we get the following equation of motion:

$r_{k1}(t + \Delta t) = r_{k1}^{uc}(t + \Delta t) - 2(\Delta t)^2 m_{k1}^{-1} \tau_k\ (t) r_{k1k2}(t)$…………….………… 2.11

Where r_{uc} is position updates with unconstrained force only. This procedure is repeated till defined tolerance is given.

2.1.7. Temperature and pressure computation

The initial temperature of the system can be measured by connecting it to a Berendsen thermal bath [295]. As a result, the bath serves as a thermal energy source, supplying and removing heat as needed from the system. The following is the outcome of rectifying the discrepancy between the system temperature T (t) and the bath temperature T0:

$$\frac{dT(t)}{dt} = \frac{1}{\tau}\{To - T(t)\} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; 2.12$$

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

$$\chi = [1 + \frac{\Delta t}{\tau_T}\,(\frac{To}{T(t)} - 1)] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.13$$

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

Pressure is controlled in the same way that temperature is controlled. The pressure can be maintained by scaling the simulation cell size and atomic positions by a factor regularly, and the system can be connected to a barostat.

$$\mu = 1 - \omega\frac{\Delta t}{\tau_p}\,(P\text{-}P_0) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.14$$

where ω represents the isothermal compressibility, τ_p represents the relaxation constant, P_0 is the pressure of the barostat ‘P’, the momentary pressure at the time ‘t’ and Δt is the time step. In this study, the standard simulation software AMBER12 was used [296]. The MD simulation is carried performed by Pmemd, one of the AMBER modules.

2.1.8. Water molecule models

TIP3P [297], TIP4P [298], TIP5P [299], and simple point charge (SPC/E) [300,301] are just a few of the molecular water models proposed for defining water in MD simulations. These models can be classified according to the number of sites, structure (rigid or flexible), and polar effects. 3-site models are the most often used in MD simulations because of their ease of use, appropriate structural and thermodynamic descriptions, and computational efficiency. In these models, the three contact sites represent the three atoms in a water molecule. Each atom is given a specific point charge. LJ interaction properties are only seen in the oxygen atom. Some of the most well-known 3-site models include the TIP3P (transferable intermolecular potential three-point), SPC (simple point charge), SPC/E (extended simple point charge), and others [170]. The majority of these models have a stiff geometry similar to that of a water molecule. Simulations for this thesis were performed using the TIP3P water model. The TIP3P water model's O-H bond length (rOH) and H-O-izationH bond angle (HOH) are 0.9572 and 104.52°, respectively, in the gas phase. A basic TIP3P water model is shown in **Figure 2.3.**

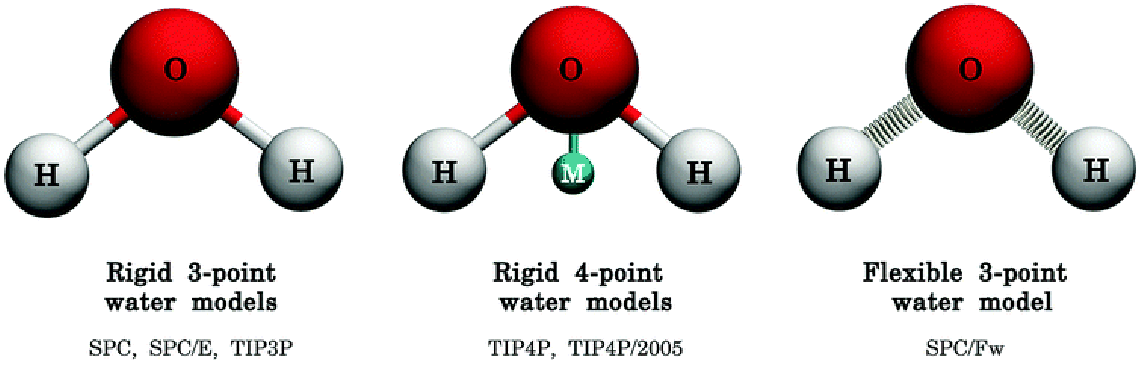

O
H
H
Rigid 3-point water models
SPC, SPC/E, TIP3P
O
H
M
H
Rigid 4-point water models
TIP4P, TIP4P/2005
O
H
H
Flexible 3-point water model
SPC/Fw

Figure 2.3. Water molecules models. Taken from [301].

2.2. Simulation methodology in AMBER

Figure 2.4 depicts the many procedures required to set up and execute MD simulation. The potential energy of the system about its position coordinates is the starting point for MD simulation. The first derivative of the potential function to position coordinates aids in the calculation of the force operating on individual atoms in the system. The following are the key steps involved in protein MD simulations.

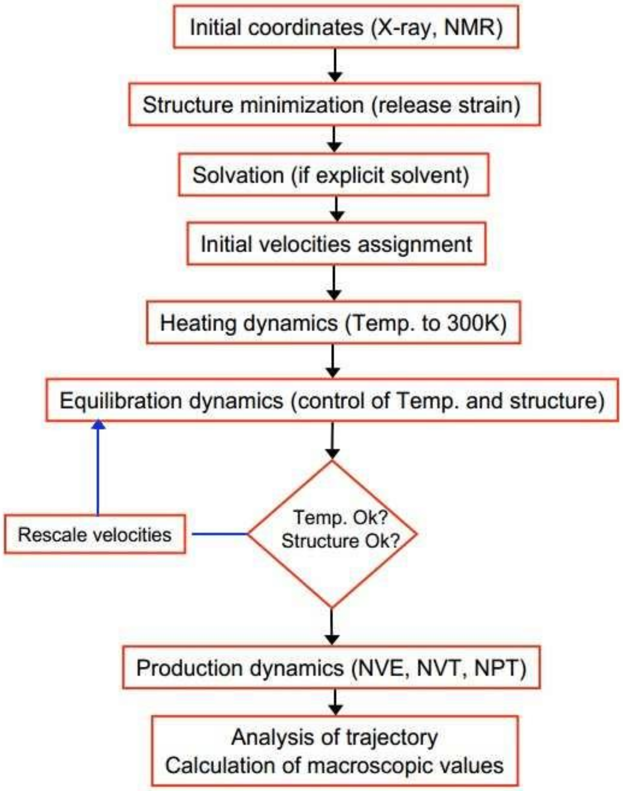

Initial coordinates (X-ray, NMR)
Structure minimization (release strain)
Solvation (if explicit solvent)
Initial velocities assignment
Heating dynamics (Temp. to 300K)
Equilibration dynamics (control of Temp. and structure)
Rescale velocities
Temp. Ok?
Structure Ok?
Production dynamics (NVE, NVT, NPT)
Analysis of trajectory
Calculation of macroscopic values

Figure 2.4. Flowchart showing the steps involved in MD Simulation.

2.2.1. Simulation environment

Protein simulation is used to simulate experimental circumstances, hence several factors for various physical situations are evaluated (such as pressure, and temperature). The protein simulation is usually done in a canonical ensemble (NVT) [302], especially up to the initial equilibration steps; after equilibration, the production dynamics are usually done in an isothermal-isobaric (NPT) [303] ensemble. The canonical ensemble (NVT) is a collection of all systems with a set number of atoms (N), a fixed volume (V), and a fixed temperature (T). The isobaric-isothermal ensemble (NPT) is a set of atoms with a fixed number of them (N), a fixed pressure (P), and a fixed temperature (T).

The protein molecules must be retained in the unit cell and solvated with a specified solvent to execute MD simulations. Our simulation until the TIP3P water model. Water models are required to accurately represent the precise nature and complexity of molecular hydration, including solvent dipole orientation and effective electrostatic shielding, minor hydrogen bond network rearrangements, and entropy changes. Because it is difficult for simulation environments to explicitly treat hydrogen bonds due to limitations in MD simulation time resolution and the intricate quantum structure of hydrogen bonds, the SHAKE method is employed for solvent hydrogen repositioning. When we utilize implicit solvent models, on the other hand, we try to approximate the solute potential of the mean force, which regulates the statistical weight of solute conformations and is calculated by averaging over the solvent degrees of freedom [304-306]. To avoid polarization of the simulation ensemble during MD simulation, keep the total charge of the system at zero by injecting positive or negative ions as needed. Constrained spherical boundary models for solute and solvent, or the widely used cubic or rectangular PBC method, can also be used to eliminate interaction difficulties at the system boundary. To compute the electrostatic interaction in biomolecular systems, the Ewald summation has been utilized directly in typical solvated periodic boundary simulations [288-292, 307].

2.2.2 Energy minimization

Finding the global minimum energy concern the position of side-chain atoms, which indicates the geometry of the particular groupings of atoms in which the net attractive force on each atom reaches a maximum, which is what energy minimization entails. It's a numerical approach for finding a minimum on the potential energy surface, starting with a higher-energy initial structure, such as "1" in Figure 2.5. The shape of the molecule is modified progress during energy minimization ion, from steps 2 to 3 to 4 as shown in **Figure 2.5**, such that the energy of the molecule is minimum. After a series of steps, the potential energy surface reaches a local or global minimum [308].

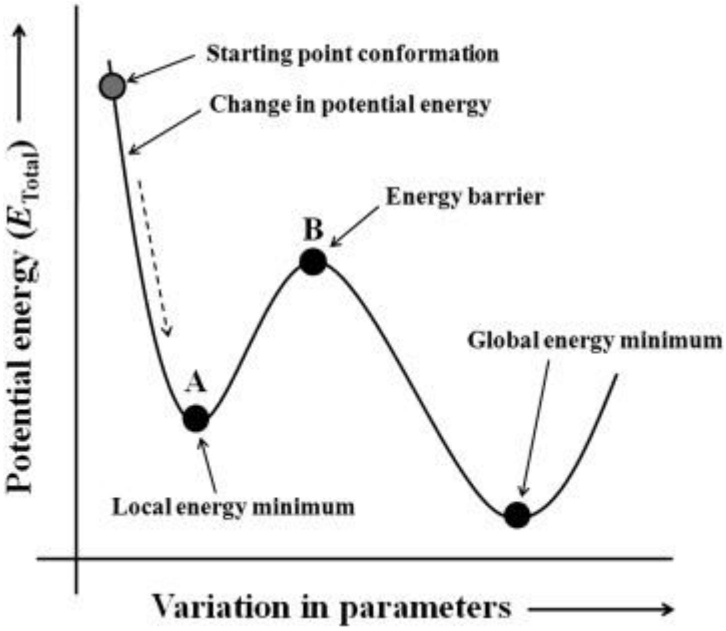

Starting point conformation
Change in potential energy
Energy barrier
B
Global energy minimum
A
Local energy minimum
Potential energy (E_{Total})
Variation in parameters

Figure 2.5. The process of energy minimization. Taken from [308].

It's vital to do energy minimization on the structure in order tozed eliminate bad contacts that could cause structural distortion. The steepest descent and conjugate gradient methods are the most often used methods for computing the least energy.

i.*The Steepest Descents Method*

The steepest descent method [309-311], which uses the gradient of the potential energy surface, is one of the numerous first-order iterative descent methods. It is directly related to the forces in the molecular mechanical description of molecular systems, and it is used to lead a search path to the nearest energy minimum. It moves in a direction that is parallel to the net force. For 3*N* Cartesian coordinates this direction is most conveniently represented by a 3*N* dimensional unit vector, $\boldsymbol{S_k}$ Thus:

$$\boldsymbol{s}_k = -\boldsymbol{g}_k/|\boldsymbol{g}_k| \quad \text{.........} \quad 2.15$$

After deciding on a movement direction, the next step is to choose how far along the gradient you want to go. When we picture a cross-section through the surface along the line, the gradient direction from the beginning point is along the line indicated; the function will pass through a minimum and then grow. We can use a line search to find the minimal position, or we can take a little step along the force's direction.

The minimum in the function along the gradient's direction is located using a line search.

ii. *Conjugate Gradients Minimization*

In narrow valleys, the conjugate technique [309, 312] produces a set of directions that does not exhibit the oscillating behavior of the steepest descents method. The gradients at each location are orthogonal in conjugate gradients, but the directions are conjugate. For a quadratic function with M variables, a set of conjugate directions has the condition that the minimum will be attained in M steps. Using point x k, the conjugate gradients technique proceeds in the direction v k, where v k is computed from the gradient at the point and the prior direction vector v (k-1) [177].$v_k = -g_k + \gamma_k v_{k-1}$

iii. *Newton-Raphson Method*

Both the second and first derivatives are used in the Newton-Raphson approach [309, 313, 314]. It uses the curvature information in addition to the gradient information to estimate where the function's gradient will change direction. It is the approach of energy minimization that requires the largest processing resources. We may multiply the inverse of the second-derivative matrix by the gradient to get a vector that will take us straight to the nearest minimum because the whole second-derivative matrix determines the curvature in each gradient direction. In mathematics, this is stated as:

$r_{min} = r_o - {A_o}^{-1.\nabla V(r_o)}$ ……………………………………………..… 2.16

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.

Water molecules are added to the system before minimization if it is needed to solvate it. The solvation process is carried out in a big box of water that has already been equilibrated. Water molecules that overlap the proteins are removed from the system because it is completely covered by the water box. At this step, the protein should be fixed in its energy-minimized position and energy minimization should be performed. This permits the water molecules to re-adjust their positions about the protein molecule [309, 313, 314].

2.2.3. Heating the system

During the heating phase, each atom in the system is assigned an initial velocity (at 0 K), and Newton's equations of motion are numerically integrated to reflect the system's time evolution. New velocities corresponding to a slightly higher temperature are assigned at brief specified intervals, and the simulation is allowed to run until the target temperature is reached (that is 300 K). As structural stresses decrease due to heating, force constraints on various subdomains of the simulation system are gradually lifted. The majority of heating dynamics are done at a constant volume (NVT) [302].

2.2.4 Equilibration

During the heating phase, each atom in the system is given an initial velocity (at 0 K), and Newton's equations of motion are numerically integrated to reflect the time evolution of the system. At short predetermined intervals, new velocities corresponding to a slightly higher temperature are assigned, and the simulation is left to run until the goal temperature is reached (that is 300 K). Force constraints on various subdomains of the simulation system gradually loosen when structural stresses diminish as a result of heating. The vast majority of heating dynamics are carried out at a fixed volume (NVT)

2.2.5 Production

MD simulation concludes with the production phase. It is used to produce a protein molecule's trajectory following specific equilibrium conditions over a given time frame (NPT) [303]. Thermodynamic parameters can be determined throughout the simulation's creation phase. The timescale might range from a few hundred picoseconds to thousands of microseconds or more.

2.2.6. Analysis

For subsequent analysis, the system's stored coordinates and velocities are employed in this stage. For analysis, you'll need MD trajectory files. When paired with visualization software (e.g., UCSF Chimera [315], VMD [316]), which may display the structural parameters of interest in a time-dependent manner, MD simulations can assist view and comprehending conformational changes at an atomic level. Quantities like time average structure, Root Mean Square Deviation (RMSD) difference between two structures, Root

Mean Square Fluctuation (RMSF), Radius of Gyration (Rg), Secondary Structure Analysis: Using the cpptraj or ptraj modules [317] of AMBER12, quantities like time average structure, Root Mean Square Deviation (RMSD), Root Mean Square Fluctuation (RMSF), Radius of Gyration (Rg).

i. Time average structure

ii. RMSD: The deviation of a structure concerning a particular conformation is measured by RMSD. It is defined as:

$$\mathrm{RMSD} = \left(\frac{\sum_N (R_i - R_i{}^0)^2}{N}\right)^{1/2} \quad \text{………………………………………..…………… 2.17}$$

where N is the total number of atoms/residues considered in the calculation, and Ri is for the vector position of particle i (target atom) in the snapshot, R_i^0 is the coordinate vector for reference atom i. RMSD was computed based on backbone atoms and taking the first frame of the simulation as the reference.

iii. RMSF: It is helpful for describing regional variations along the protein chain. It is calculated as:

$$\mathrm{RMSF} = \left(\frac{1}{T}\sum_{t=1}^{T}(r_i(t) - r_i{}^{ref})^2\right)^{1/2} \quad \text{……………………………………………… 2.18}$$

T is the trajectory time over which the average is taken, r_i (t) is the position of the atoms in residue i and r_i^{ref} is the reference position of particle i.

iv. Rg: It computes how an object's parts are distributed along its axis. It gives the compactness of a protein. It is calculated as:

$$\mathrm{Rg} = \left(\frac{1}{N}\sum_i (r_i - r_{cm})^2\right)^{1/2} \quad \text{…………………………………………..……………2.19}$$

where r_i - r_{cm} is the distance between atom i and the center of mass of the molecule.

2.3. 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)

2.3.1 Free energy decomposition using MM-PBSA/GBSA.pl script and python script MMPBSA.py

The free energy of small ligand binding to receptor proteins or protein-protein complexes is calculated using the methods MM-PBSA and MM-GBSA [318-320]. They're usually correct because they're based on molecular dynamics simulations of the protein–liganded combination. The binding free energy (G_{bind}) of a ligand and a receptor to create a protein-ligand complex is computed in MM-PBSA or MM-GBSA as:

$$\Delta G_{bind} = \Delta G_{complex,solv} - (\Delta G_{protein,solv} + \Delta G_{ligand,solv}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.20$$

where $\Delta G_{complex,solv}$, $\Delta G_{protein,solv}$, and $\Delta G_{ligand,solv}$ are, respectively, the variations in free energy for the complex, the protein, and the ligand with or without solvent. Herein, a subscript "solv" in eqn. 2.20 represents the aqueous solution. The solvation free energies are calculated as follows;

$$\Delta G_{comp,\ solv} = E_{MM} + \Delta G_{solvation} - TS_{solute} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.21$$

$$E_{MM} = E_{int} + E_{elec} + E_{vdW} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.22$$

$$E_{int} = E_{bond} + E_{angle} - E_{torsion} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.23$$

$$\Delta G_{solvation} = \Delta G_{PB/GB\ solvation-elec} + \Delta G_{SASA,nonpolar} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.24$$

In Eqn. 2.21, 'comp' denotes the complex (protein + ligand). The energy of molecular mechanics (MM) from the force field without the solvent is called EMM. E_{int} is made up of E_{bond}, E_{angle}, and $E_{torsion}$, which are all intramolecular contributions. The intermolecular electrostatic and van der Waals interaction energies are E_{elec} and E_{vdW}, respectively. The solvation free energy is $G_{solvation}$, and $G_{solvation-elec}$ is determined using the Poisson–Boltzmann technique. The solvent-accessible surface area is used to calculate $G_{nonpolar}$ (SASA). The temperature and entropy of a solute are T and S_{solute}, respectively. **Figure 2.6** depicts the relationship for each energy.

Electrostatic solvation energy can be calculated using the PB and GB methods. The internal (solute) and outside (water) dielectric constants were set to 1 and 80, respectively, for the interior (solute) and exterior (water). The atomic radii and charges employed in MD simulations are the same. The non-polar contribution (G_{SASA}) to the solvation free energy was determined using Eqn. 2.24 from the solvent accessible surface area.

$$\Delta G_{SASA} = \gamma \times SASA + b \qquad \text{..} \; 2.25$$

Here, SASA is the solvent-accessible surface area and γ is the surface tension parameter. 'γ' is set as 0.005 kcal ($mol^{-1}Å^{-2}$) for PB and 0.0072 kcal ($mol^{-1}Å^{-2}$) 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. Solvent probe radius is configured at 1. 4. The entropy calculation was neglected in the above calculation as we are interested in calculating only relative binding energy contribution to the formation of the protein-ligand complex.

Amber12, after the work of Gohlke et al [191, 190], presents numerous strategies to break down estimated free energy into particular residue contributions using either the GB or PB models [190]. Per-residue energy decomposition decomposes interactions for each residue by only including those interactions in which one of the residue's atoms is involved. Pairwise decomposition, on the other hand, decomposes interactions by specific residue pairs by including only those interactions in which one atom from each of the studied residues participates. In free energy calculations, these decomposition strategies can reveal crucial interactions [191, 190]. The dielectric barrier generated between the protein and the bulk solvent is intrinsically nonlocal and depends on the configuration of all atoms in space, hence solvation free energies utilizing GB and PB are not strictly pairwise decomposable. As a result, when evaluating free energy decomposition data, caution is required.

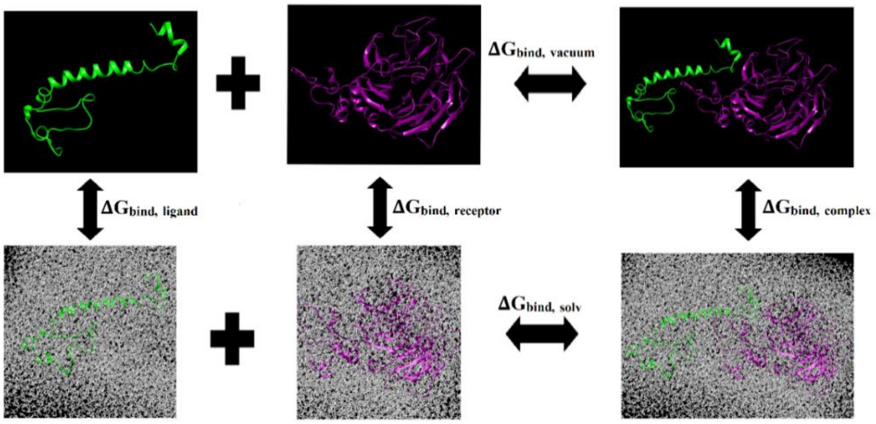

$\Delta G_{bind,\ vacuum}$
$\Delta G_{bind,\ ligand}$
$\Delta G_{bind,\ receptor}$
$\Delta G_{bind,\ complex}$
$\Delta G_{bind,\ solv}$

Figure 2.6. Computational plans based on MM/PBSA for the binding free energies. Taken from [chapter 5B].

Using the Per-residue decomposition approach in Python script MMPBSA.py [321-323], we can determine the partial binding free energy contribution to the amino acid residue Y (G^{Y} bind). The contribution of each residue to the overall binding free energy may be calculated using per-residue basis decomposition [321-323]. To derive G^{Y} bound, first split terms in Eqn. (2.25) into their atomic contributions. The contribution of each atom a to the overall electrostatic interaction energy is calculated as follows:

$$E^{a}_{elec} = \frac{1}{2}\sum_{b\neq a}\frac{q_a q_b}{r_{ab}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.26$$

q_a and q_b are the atomic partial charges of atoms *a* and *b*, respectively, while r_{ab} is the distance between them. Similarly, to avoid duplicate counting, one-half of the pairwise energy for van der Waals interaction energy between protein and ligand, E^{a}_{vdW}. A non-polar fraction of the solvent impacts on binding free energy is expressed as using the SASA of each atom *a*.

$$\Delta G^{a}_{nonpolar,solv} = \gamma\{(SASA^{a,complex} - (SASA^{a,protein} + SASA^{a,ligand})\} \ldots\ldots\ldots\ldots 2.27$$

Where, $SASA^{a,\ protein}$ and $SASA^{a,\ the\ ligand}$ is equal to 0 depending on which component the atom belongs to. γ is set to 0.0072 kcal mol^{-1} $Å^{-2}$ in AMBER 12. To calculate the contribution of an atom *a,* to the electrostatic part of solvent effects, the generalized Born (GB/PB) approach is used. The contribution of an atom *a* is given by;

$$\Delta \mathrm{G}^{\mathrm{a}}_{\mathrm{elec,sol}} = -\frac{1}{2}\sum_{a}\left(1-\frac{e^{-k\int_{ab}^{GB}}}{\varepsilon_{\omega}}\right)\frac{q_a\ q_b}{\int_{ab}^{GB}(r_{ab})}+\frac{1}{2}\sum_{b\neq a}\frac{q_a\,q_b}{r_{ab}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 2.28$$

$$\int_{ab}^{GB} = [\,r_{ab}^{2} + \alpha_a\alpha_b \exp\left(\frac{-r_{ab}^{2}}{4\ \alpha_a\ \alpha_b}\right)]^{1/2} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.29$$

Where the Debye-Huckel screening parameter [324, 325] is written as *κ*. ε_{ω} is a dielectric constant for the solvent set as 80. effective Born radii of atoms *a* and *b* are α_a and α_b, respectively. The partial binding free energy contribution to amino acid residue Y is calculated using these contributions to each atom.

$$\Delta G_{bind}^{Y} = \sum_{a\in Y}\left(E_{elec}^{a} + \ E_{vdw}^{a} + \Delta G_{nonpolar,solv}^{a} + \Delta G_{elec,solv}^{a}\right) \quad \ldots\ldots\ldots 2.30$$

Here the entropic (Eqn 2.21) and intra-molecular contributions are neglected in this analysis.

Another method for deconstructing free energy is to introduce particular mutations into the protein sequence and see how binding free energies or stabilities are changed. Alanine scanning mutagenesis is a method in which an amino acid in the system is transformed into alanine, which might emphasize the relevance of the electrostatic and steric character of the original side chain [226-229]. Assuming that the mutation does not a protein structure, we may include it directly into each member of the original ensemble. This eliminates the requirement for an addition-al MD simulation to produce an ensemble for the mutant.

2.4. Molecular docking

The molecular docking method is used to model the interaction between a protein and a small molecule or between protein-protein interactions at the atomic level, allowing us to characterize the behavior of small molecules in the binding sites of target proteins or to obtain the interacting interface residues in protein-protein interactions, which may reveal fundamental biochemical processes [330-333]. The docking procedure consists of two main parts. First, anticipate the ligand conformation as well as its location and orientation within the active site (referred to as pose), or just choose the ligand conformations in the active site. The second step is to rank the conformations using a scoring mechanism based on binding affinity. Algorithms should be able to recreate the experimental binding mode during the first sampling, and the scoring function should be able to rank it highest among all created conformations.

2.4.1 Docking methodologies

Protein-protein interactions (PPIs) or DNA-protein interactions (DPI) are important driving mechanisms in many physiological processes in the cell and are also implicated in the pathophysiology of many disorders. Because of the variety of PPIs and DPIs, comprehensive analysis of the nature of the biomolecular interactions is required. The determinant of the specificity and stability of PPIs and DPIs is critical. The size of their interface determines whether the complex is temporary or permanent [334-337], which is driven by hydrophobic effects that occur between the nonpolar areas of biomolecular residues through van der Waals interactions. The electrostatic complementarity of the interacting protein surfaces between two molecules increases the development and longevity of the protein-protein complex (PPC) or DNA-protein complex (DPC). Hydrogen bonding and electrostatic interaction are important in guiding the docking characteristics between two molecules at particular surfaces. Prediction of PPIs and DPIs is critical in drug discovery. Many physiological and pathological cellular processes rely on PPIs and DPIs, which can/may be disrupted by external influences. The contemporary drug discovery method consists of two major steps: (i) identifying a potential therapeutic target, evaluating its characteristics, and (ii) creating a matching ligand [334-337].

One of the most difficult challenges in structural biology is the computerized prediction of PPIs and DPIs. Many biological experiments, both academic and industrial, may benefit from accurate high-accuracy interaction prediction. The goal of protein-protein docking and DNA-protein docking is to discover the precise connection of two interacting molecules. The correct forecast is based on the residue contacts engaged in the target interaction. Many docking methods have been developed. However, only a handful algorithms are now provided as free online services [338-341. The methods largely differ in the way they use to search the six-dimensional transformation space and in how they evaluate the resolved complexes. For protein-protein docking, and DNA-protein docking, we have utilized Hex software [342, 343], SymmDock server [344, 345] and the ClusPro [346, 347] server in this work.

2.4.1.1 HexDock software

Hex 8.0.0 software is a docking tool that uses the rigid-body fast Fourier transform (FFT). The docked conformers are generated using spherical polar Fourier (SPF) correlations. The docking parameters usually employed are as follows:

HEX Dock Parameters:
Correlation type – Shape + Electro
FFT Mode – 3D
Sampling methods- Range angles
Grid Dimension – 0.6
Receptor range – 180
Ligand Range – 180
Twist range – 360
Distance Range – 40
Steric scan – 16
Final search- 30

Hex software employs five docking steps, which are as follows:

SPF transform → FFT Steric scan → FFT final search → energy refinement → total dockings.

The resulting docked structures are graded based on their energy values (E-values), with a lower value indicating a better-docked conformer.

2.4.1.2. SymmDock server

The SymmDock is a cyclically symmetric docking-driven algorithm server. It predicts the cyclically symmetric complex structure from the given structure of its asymmetric unit. This method is made up of Cn symmetry type complexes, which are complexes having rotational symmetry of order n about a symmetry axis. The value n here denotes the required number of asymmetric units in the output complex. The rotation angle for the symmetry complex is assumed to be 360/n°. The input for this docking procedure is a 3-D coordinate set of the asymmetric unit molecule and a number. This server, uses local feature matching to generate the candidate set of alterations. SymmDock limits its search to symmetric cyclic transformations of a particular order n. this server makes advantage of the unique properties of cyclically symmetric transformations in both its clustering and search strategies. Lists of complexes satisfying the cyclic symmetry constraints are generated as a result of this method's output. This approach, on the other hand, determines the symmetric transformation space for transformations that optimize the shape complementarity of the interface between nearby units. The user interface for SymmDock (http://bioinfo3d.cs.tau.ac.il/ SymmDock/), since it only accepts one molecule with symmetry order. Aside from that, the invocation and receipt outcomes remain the same; notably, the result notice with the necessary web link is emailed to the user.

2.4.2.2.1 Input

Two factors are required for the input: the asymmetric unit (i.e., the monomer) and the symmetry order (2 for dimer, 3 for trimer etc.). The first, asymmetric unit, may be supplied in PDB format or given by its PDB ID. For the symmetry order, it can be any number higher than 2. This server, on the other hand, exclusively anticipates cyclic symmetry. It is worth noting that a more complex repeating application of SymmDock can handle other symmetries as well. The user's email address is also asked for notification purposes.

2.4.2.2.2 Output

For SymmDock output, a web page displaying the projected solutions is prepared, and an email with a link to that page is delivered to the user. Instead of exhibiting only the pairwise interactions involved in the complex, the whole multimer is generated for each solution. For example, if the user request was to forecast a symmetric complex of order 12, each answer is a dodecamer of the asymmetric unit. The options for viewing and downloading solutions based on lower ranking solutions, as well as downloading a group of solutions are easy to access.

2.4.1.3. ClusPro web server

ClusPro (https://cluspro.org) is a web-based server that was established in 2004 and has since been significantly changed and expanded. This server is used for direct docking of two proteins that interact. The service requires two PDB-format protein files for docking. The server goes through three computational phases during docking as shown in **Figure 2.7**:

i. Employs rigid body docking to sample billions of conformations.
ii. RMSD-based grouping of the 1000 lowest energy structures is generated to locate the biggest clusters that will reflect the complex's most likely models.
iii. The energy minimization method is used to improve selected structures (**Figure 2.7)**. The rigid body docking stage employs PIPER, [348] a docking software based on the Fast Fourier Transform (FFT) correlation technique.

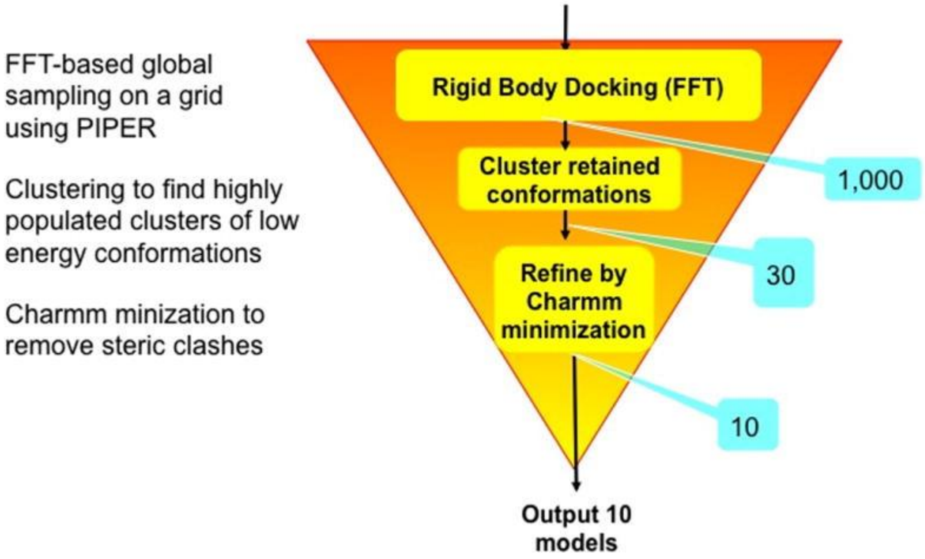

FFT-based global sampling on a grid using PIPER
Clustering to find highly populated clusters of low energy conformations
Charmm minization to remove steric clashes
Rigid Body Docking (FFT)
Cluster retained conformations
Refine by Charmm minimization
1,000
30
10
Output 10 models

Figure 2.7 Outline of the ClusPro algorithm. After each step, the number of structures retained is shown in a blue box. Taken from [346].

2.5. Result analysis tools

2.5.1. PDBsum web server

PDBsum (http://www.ebi.ac.uk/pdbsum) [349, 350] is a web-based database that gives a visual summary of the relevant information about each macromolecular structure deposited at the Protein Data Bank (PDB). It includes photos of the structure, annotated plots of each protein chain's secondary structure, extensive structural analysis, summary PROCHECK results, and schematic diagrams of protein–protein, protein–small molecule, and protein–DNA interactions. RasMol scripts emphasize crucial structural properties such as protein domains, PROSITE patterns, and protein–protein/ligand interactions. PDBsum is updated anytime new structures are added to the PDB and is publicly available at http://www.biochem.ucl.ac.uk/bsm/pdbsum.

2.5.1.1 Protein-protein interactions

In PDBsum server lies a sub-program called LIGPLOT, where the interactions across protein–protein interfaces are shown in detail (**Figure 2.8**).

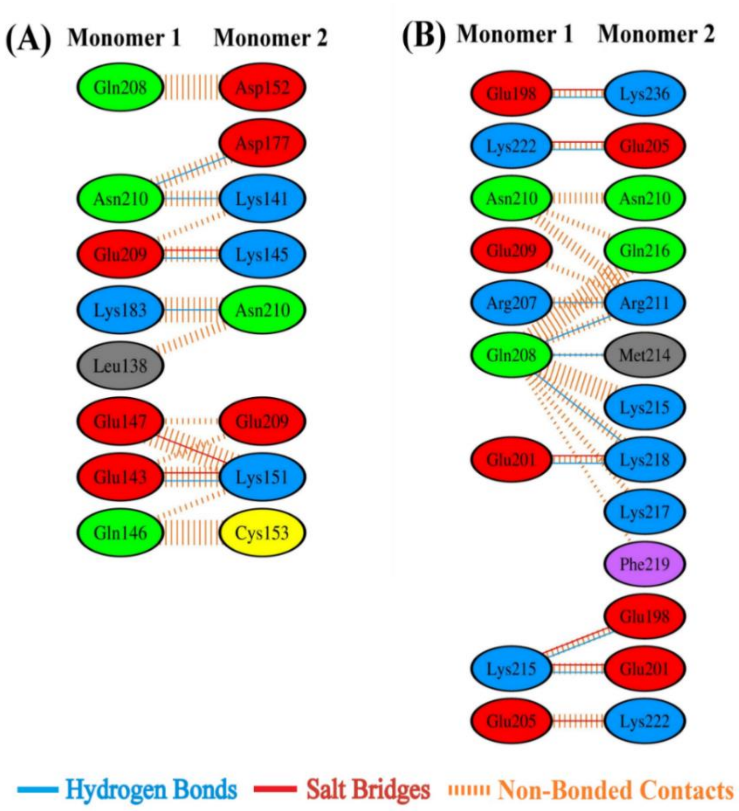


Figure 2.8 Protein–protein interaction diagrams obtained from PDBsum server. Taken from **[chapter 3].**

2.6. DNAproDB server

DNAproDB, developed by Jared et al. 2017, is a database, structure processing pipeline, and web-based visualization tool for analyzing DNA-protein complexes structurally. This database, in particular, comprises all structural and biochemical properties collected from the PDB. By building searches on a set of features using the search form, this data can be utilized to evaluate individual structures or to generate massive databases. the users can also upload their own models to analyze and visualize as well as interpret the results [351. 352].

In a nutshell, DNAproDB can aid the users with following concerns:

i. Look up features of DNA, proteins, or DNA-protein interactions in thousands of DNA-protein complex structures.
ii. Customizable, interactive visualizations help you see data in a new light. They can be used as a data exploration tool or exported for use in publications.
iii. Upload your own structures from experiments or simulations, and our pipeline will provide the same data and visualizations as any structure in our database.

Their built-in visualization tools enable both flexibility and interactivity, and the data user gets can be downloaded directly.

For our work, we considered following parameters:

i. Major groove residue interaction: at least 2.0 Å2 buried accessible surface area (BASA), 1 hydrogen bond and 1 van-der Waals (vdW) interaction;

ii. Minor groove residue interaction: 1.0 Å2 BASA, 1 hydrogen bond and 1 vdW interaction;

iii. Backbone-residues interaction: 5.0 Å2 BASA, 1 hydrogen bond and 1 vdW interaction;

2.7. 3D structure visualization tools

2.7.1 Visual molecular dynamics (VMD)

VMD is a piece of computer software that allows you to model and see molecules. VMD is primarily used to examine and analyze MD simulation results. Other tools are available for working with volumetric data, sequence data, and arbitrary graphical objects [316].

2.7.2 UCSF Chimera

Chimera, developed at the University of California, San Francisco (UCSF), is a sophisticated tool for interactive viewing and analysis of molecular structures and data, including density maps, supramolecular assemblies, sequence alignments, docking findings, and conformational ensembles. Chimera was developed with funding from the National Institutes of Health's Resource for Biocomputing, Visualization, and Informatics (RBV) [315].