

---

---

## **CHAPTER 3**

### **Materials and Methods**

---

---

### Materials and Methods

#### 3.1.Tools and Techniques:

To study the structural dynamics as well as interaction profile of S protein with its primary receptor ACE2 and other alternative or co-receptor, as well as Mpro 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:***

Molecular dynamics (MD) simulations play a crucial role in understanding the fundamental physics governing the structure and function of biological macromolecules. These simulations have been extensively employed to explore the structure and dynamics of macromolecules, such as proteins and nucleic acids. By bridging the gap between theoretical models and experimental observations, MD simulations replicate experiments, reveal unseen microscopic details, and provide deeper insights into experimental findings. In an MD simulation, the classical equations of motion are numerically solved in a stepwise manner, enabling a comprehensive analysis of molecular interactions. This approach is widely used to investigate the time-dependent behavior of a system composed of particles, typically atoms or molecules with known properties. MD simulations have a broad range of applications, including modeling molecular interactions between therapeutic drugs and biomolecules, simulating the microscopic mechanism of water droplet freezing, calculating the thermodynamic and rheological properties of various hydrocarbon mixtures and simulating the effects of a particular combination of organic molecules on properties that affect solar cell efficiency. Quantum MD simulations, representing a significant advancement over classical methods, provide valuable insights into various biological challenges [1].

##### **3.1.1.1. History of Simulation:**

The MD simulation method was first introduced in the late 1950s by Alder and Wainwright to investigate the interactions of hard spheres [2,3]. Their work led to several significant discoveries regarding the behavior of simple liquids. A major breakthrough in MD simulations occurred in 1964 when Rahman conducted the first simulation using a realistic potential for liquid argon [4]. Further progress was made in 1974 when Rahman and Stillinger performed the first MD simulation of a realistic system, specifically liquid

water [5]. In 1977, the first protein simulations were documented, including the simulation of the bovine pancreatic trypsin inhibitor (BPTI) [6]. Today, MD simulations are widely used to study solvated proteins, lipid systems, and protein-DNA complexes, addressing challenges such as ligand binding thermodynamics and the folding of small proteins. The field has expanded significantly with the introduction of combined quantum mechanical-classical simulations, which are essential for gaining insights into enzymatic processes within the context of entire proteins. Additionally, MD simulation techniques are extensively utilized in various experimental methods, including X-ray crystallography and NMR structure determination.

### 3.1.1.2. Theory of molecular dynamics simulation:

The MD simulation method is based on the equation of motion. Newton's second law of motion, expressed as  $F=ma$ , where 'm' represents the mass of a particle, 'a' denotes its acceleration, and 'F' is the applied force, serves as the basis for this method. However, if the force acting on each atom in the system is known, then one should be able to calculate the acceleration for each atom. Newton's equations of motion enable the prediction of an atom's spatial position over time. Although MD modeling techniques may require significant computational time and resources, continuous advancements in technology are reducing both costs and processing times. In protein-solvent simulations, temporal calculations now extend to the millisecond scale. The equation of motion based on Newton's law can be stated as: -

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

$$F = ma \dots \dots \dots (3.2)$$

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

In this equation,  $F_i$  defines the force exerted on particle  $i$ ,  $m_i$  as the mass of particle  $i$ ,  $a_i$  as the acceleration of particle  $i$  and  $\mu$  is a scalar potential function. The Newton's force,  $F_i$  can also be expressed as the potential energy gradient as:

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

By combining the two equations given,

$$-\nabla_i V = m_i \frac{d^2 r_i}{dt^2} (1+x)^n = 1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \dots \dots \dots (3.5)$$

Where,  $r_i$  is the position vector of particle  $i$  and  $\frac{d^2 r_i}{dt^2}$  is the second derivative of position with respect to time, which represents the acceleration of the particle. The system's potential energy, or  $V$ , is defined here. There is a connection between Newton's equation of motion and the potential energy derivative that is utilized to explain position changes in time. The main objective of the numerical integration for Newton's equation of motion is to find a formula that may define position  $r_i(t+\Delta t)$  at time  $t+\Delta t$  in relation to the locations that are known at time  $t$ . However, the Velocity Verlet technique employs the positions and accelerations at time  $t$  as well as the positions from time  $t-\Delta t$  to get the new locations at time  $t+\Delta t$ . The velocities in this procedure are not produced directly. Although their knowledge is not required for evolution, it is required to calculate the kinetic energy  $K$ . Throughout the route, the entire energy  $E=K+U$  should be preserved. The position of atom in every  $\Delta t$  time step:

$$r_i(t+\Delta t) = r_i(t) + v_i(t)\Delta t + \frac{1}{2}a_i(t)\Delta t^2 \dots \dots \dots (3.6)$$

The velocity is used as a half time step: -

$$r_i(t + \Delta t) = r_i\left(t - \frac{\Delta t}{2}\right) + r_i(t) \frac{\Delta t}{2} \dots \dots \dots (3.7)$$

The velocities can be computed from the  $\Delta t$  time step: -

$$r_i(t) = \frac{r_i\left(t+\frac{\Delta t}{2}\right)^2 + r_i\left(t-\frac{\Delta t}{2}\right)^2}{2} \dots \dots \dots (3.8)$$

It is significant that when kinetic energy is required at time  $t$ , then the velocity rescaling is required. Also, the required atomic positions are obtained from:

$$r_i(t + \Delta t) = r_i(t) + r_i(t + \Delta t) \frac{\Delta t}{2} \dots \dots \dots (3.9)$$

Where,  $r_i(t)$  is Position at time  $t$ ,  $r_i(t+\Delta t)$  is the position at time  $t+\Delta t$ ,  $v_i(t)$  is velocity at time  $t$  and  $a_i(t)$  is the acceleration at time  $t$ .

Force fields are utilized to describe the variations in bond lengths, bond angles, torsional movements, and non-bonding interactions, such as van der Waals and electrostatic forces, between atoms over time. Designed to mimic molecular geometry and capture specific characteristics of the studied structures, a force field consists of a collection of linked constants and equations.

### 3.1.1.3. Force Fields:

In mathematics, a force field (FF) represents an expression that defines how a system's energy varies based on the positions of its constituent particles. It comprises an analytical interatomic potential energy function,  $U(r_1, r_2, \dots, r_N)$ , along with a set of associated parameters. These parameters are generally derived from semi-empirical quantum mechanical methods, ab-initio calculations, or by fitting experimental data obtained through techniques such as X-ray, neutron, and electron diffraction, as well as infrared, NMR, Raman, and neutron spectroscopy. Within this framework, molecules are modeled as assemblies of atoms connected by simple elastic (harmonic) forces. Instead of using the true potential, the FF employs a simplified representation that remains valid within the simulation domain. Ideally, the system of interest should be complex enough to capture essential properties while remaining computationally efficient for rapid evaluation. Various force fields exist in the literature, each tailored to specific types of systems and differing in their level of complexity. A classical FF expression might look like this:

$$V(r^N) = \sum_{bonds} \frac{k_i}{2} (l_i - l_{i,0})^2 + \sum_{angles} \frac{k_i}{2} (\theta_i - \theta_{i,0})^2 + \sum_{torsions} \frac{V_n}{2} (1 + \cos(n\phi - \phi_0)) + \sum_{i=1}^N \sum_{j=i+1}^N \left( 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\epsilon_0 \epsilon_r r_{ij}} \right) \dots \dots \dots (3.10)$$

Here,

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

$k_i \rightarrow$  force constant,  $l_i \rightarrow$  Bond length,  $l_{i,0} \rightarrow$  Equilibrium bond length.


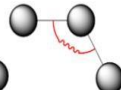
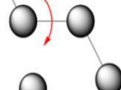
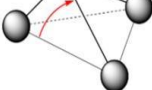

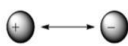
$\theta_i \rightarrow$  Bond angle,  $\theta_{i,0} \rightarrow$  Equilibrium bond angle.

$V_n \rightarrow$  Barrier height,  $n \rightarrow$  Periodicity of torsional rotation,  $\phi \rightarrow$  Torsion angle and  $\phi_0 \rightarrow$  Reference torsion angle.

$\epsilon_{ij} \rightarrow$  Depth of the potential well (interaction strength),  $\sigma_{ij} \rightarrow$  Distance where the potential is zero,  $r_{ij} \rightarrow$  Distance between particles i and j.

$q_i, q_j \rightarrow$  Charges on particles i and j,  $\epsilon_0 \rightarrow$  Permittivity of free space,  $\epsilon_r \rightarrow$  Relative permittivity (dielectric constant).

The potential energy function is mainly composed of bond lengths, bond angles, torsional rotations, and non-bonded interactions, including van der Waals forces and electrostatic interactions. **Figure 3.1** provides a graphical representation of the various interaction types.

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

**Figure 3.1.** Schematic illustration of the main contribution to the potential energy function (Taken from [7]).

The initial four terms in the equation represent the local or intramolecular contributions to the total energy, including bond stretching, angle bending, dihedral torsions, and improper torsions. The last two terms of the equation account for repulsive van der Waals interactions and Coulombic forces.

#### 3.1.1.4. Long range interactions: Ewald sum:

The Ewald Summation is a widely utilized technique for calculating electrostatic interactions in computer simulations of condensed-matter systems [8]. This method examines the errors introduced by truncating the infinite real-space and Fourier-space lattice sums in the formulation of the Ewald Sum. An optimal screening parameter of 7 is identified for the Fourier-space cutoff. Typically, it has been observed that the number of Fourier space vectors required to achieve a specific level of accuracy scales with 7/3. However, the proposed method offers a way to efficiently regulate computational parameters for Ewald sums, assess the accuracy of different Ewald-sum implementations, and compare various approaches. It remains one of the most frequently employed techniques for evaluating long-range interactions in MD simulations.

The fundamental concept underlying the Ewald sum is to analyze a charge distribution for each site's opposite sign. This additional charge distribution helps reveal interactions between nearby atoms. However, the observed interactions are short-range and can be precisely controlled using a cutoff scheme. To compensate for the additional charge

distribution, the equal charge distribution with the opposite sign and short-range interaction is averaged in the reciprocal space. As the electrostatic potential due to the screened charge is a quickly decreasing function of  $r$ , it is straightforward to calculate the contribution of a group of screened charges to the electrostatic potential at a specific location  $r_i$  using direct summation. The total potential energy for the long-range Coulomb interaction is given by the expression:

$$\mu_c = \mu_q(\alpha) - \mu_{self}(\alpha) + \Delta\mu(\alpha) \dots \dots \dots (3.11)$$

Where,  $\mu_c \rightarrow$  Corrected Chemical Potential,  $\mu_q(\alpha) \rightarrow$  Uncorrected Chemical Potential of the Charged System and  $\mu_{self}(\alpha) \rightarrow$  Self-Interaction Energy Correction.

In the **Equation 3.11**, the greater the value of  $\alpha$ , the sharper the distribution. However, a higher value narrows the screened potential range, allowing us to use a smaller cutoff radius. Additionally, the efficiency of the Ewald summation can be improved by incorporating the Fast Fourier Transform (FFT) in the reciprocal summation evaluation. Meanwhile, the Particle Mesh-based method applies a fixed cutoff for the direct space sum and utilizes an FFT-based approximation, which scales as  $N \log(N)$  for the reciprocal space summation.

### 3.1.1.5. Dealing with molecules: SHAKE algorithm:

The selection of the time step in a molecular system is restricted by the varying time scales associated with vibrational degrees of freedom, including bond vibrations, angle stretching, and torsional motion. Bonds involving hydrogen exhibit faster vibrational modes, limiting the integration time step to approximately 1 femtosecond (fs). However, by constraining these rapid degrees of freedom, a longer time step can be employed while solving for the unconstrained degrees of freedom. Since hydrogen bonds have the highest vibrational frequency, they can be constrained during MD simulations using the SHAKE algorithm, which was developed by Ryckaert et al. [9]. The SHAKE algorithm begins by applying the equations of motion to an atomic system without any initial constraints. Additionally, it utilizes the Lagrange multiplier formalism to ensure that bond lengths remain constant throughout the simulation. Assuming  $N_c$ , the constraint is given by:

$$\alpha_k = r_{k_1 k_2}^2 - R_{k_1 k_2}^2 = 0 \quad \text{Where } k = 1, 2, 3, \dots, N_c \dots \dots \dots (3.12)$$

The term  $\alpha_k$  is constraint function,  $R_{k_1 k_2}$  is considered a constrained distance and  $r_{k_1 k_2}$  is the squared instantaneous distance between the  $k_1$  and  $k_2$  atoms.

The modified constrained equation of motion is defined as:

$$m_i \frac{d^2 r_i(t)}{dt^2} = - \frac{\partial}{\partial r_i} [V(r_1 \dots r_N) + \sum_{k=1}^{N_c} \tau_k(t) \alpha_k(r_1 \dots r_N)] \dots \dots \dots (3.13)$$

In this case,  $m_i$  is referred to as the mass of the  $i^{\text{th}}$  particle,  $V(r_1 \dots r_N)$  is the Potential energy function,  $\tau_k$  is the Lagrange multiplier (unknown) for the  $k^{\text{th}}$  constraint and  $\alpha_k(r_1 \dots r_N)$  is the constraint function. However, by resolving  $N_c$  quadratic linked equations, the unknown multiplier in this modified constrained equation of motion can be determined. Finally, the motion equation has been discovered as shown below:

$$r_{k1}(t + \Delta t) = r_{k1}^{uc}(t + \Delta t) - 2(\Delta t)^2 m_{k1}^{-1} \tau_k(t) r_{k1k2}(t) \dots \dots \dots (3.14)$$

In the **Equation 3.14**,  $r_{k1}$  is corrected position of atom  $k_1$  and  $r^{uc}$  is the position updates with unconstrained force only. This method is however repetitive till the defined tolerance is specified.

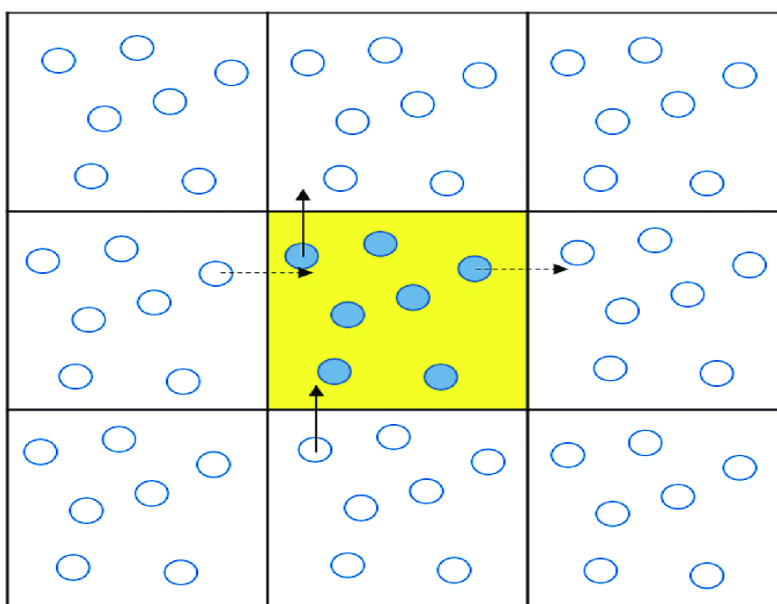
The SHAKE algorithm avoids explicit matrix inversion by iteratively adjusting particle coordinates until all constraints are met within a predefined tolerance. In addition to maintaining rigid bonds, constraint algorithms must also address constraint decay, which refers to deviations from ideal bond lengths caused by the accumulation of numerical errors. However, iterative algorithms inherently manage constraint decay by enforcing convergence within a specified tolerance at each time step. Regular corrections and verifications are performed to minimize deviations from the original bond distances. In contrast, non-iterative algorithms require an explicit mechanism to counteract constraint degradation, as they lack an inherent feedback system to detect and correct distance variations.

### 3.1.1.6. Periodic Boundary conditions:

To understand periodic boundary conditions, we consider a system comprising  $N$  interacting particles at a temperature  $T$  and confined within a volume  $V$ . To ensure that the system remains bounded by its own replicas, periodic boundary constraints, similar to those in the 2D Ising model, must be applied. As a result, it can be seen that given a system of particles, when a particle exits the central box from one side, it must re-enter from the opposite side. As illustrated in **Figure 3.2**, the atoms of the molecules are arranged within an imaginary box surrounded by translated copies of their coordinates. In this configuration, multiple replicas of particle 3, present within the central box, can



theoretically interact with particle 1 inside the same box. Furthermore, when considering interactions between particles 1 and 3, it is logical to select the interaction corresponding to the shortest interatomic distance. This method is referred to as the nearest image convention. 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. Consequently, the number of particles within the core box remains unchanged. However, to manage non-bonded interactions, a non-bonded cutoff is typically employed, ensuring that each atom interacts with only one image of every other atom within the system.



**Figure 3.2.** Periodic boundary conditions in two dimensions. The simulation cell (dark color) is surrounded by translated copies of itself (light color) (Taken from [10])

### 3.1.1.7. Temperature and Pressure computation and control:

The initial temperature of the system is determined by coupling it to a Berendsen thermal bath [11]. This bath functions as a thermal energy reservoir, supplying or removing heat as needed to regulate the system's temperature. The system temperature  $T(t)$  that deviates from the bath temperature  $T_0$  is corrected according to:

$$\frac{dT(t)}{dt} = \frac{1}{\tau} \{T_0 - T(t)\} \dots \dots \dots (3.15)$$

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

$$\chi = \left[ 1 + \frac{\Delta t}{\tau T} \left[ \frac{T_0}{T(t)} - 1 \right] \right] \dots \dots \dots (3.16)$$

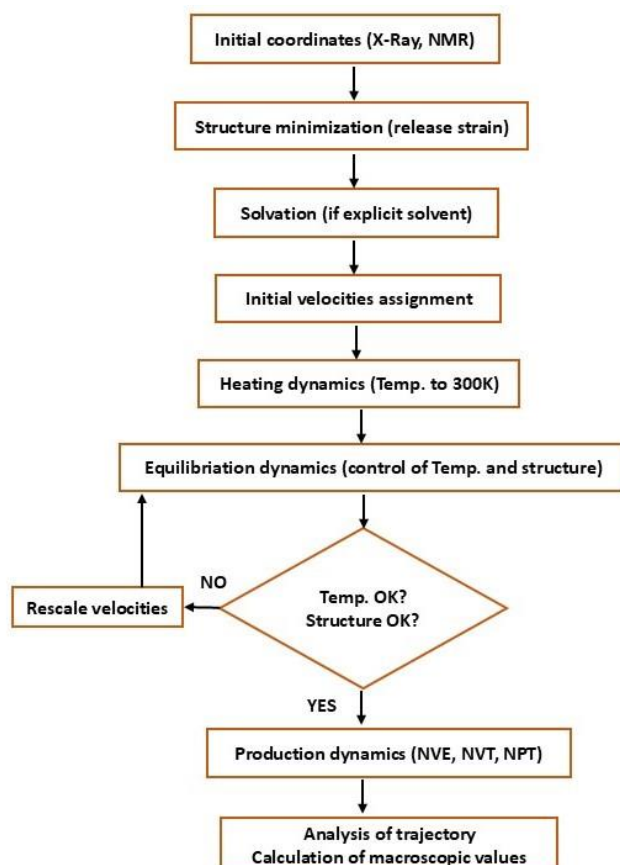
The strength of the coupling can be varied by changing the time constant  $\tau$ .

The pressure regulation method operates similarly to temperature control. By coupling the system to a barostat, the pressure can be stabilized at a constant value through periodic adjustments of the simulation cell size and atomic positions using a scaling factor  $\mu$ :

$$\mu = 1 - \omega \frac{\Delta t}{\tau p} (P - P_0) \dots \dots \dots (3.17)$$

where  $\omega$  represents the isothermal compressibility,  $\tau p$  represents the relaxation constant,  $P_0$  is the pressure of the barostat,  $P$  is the momentary pressure at time  $t$  and  $\Delta t$  is the time of step.

In this study, the standard simulation package AMBER14 is utilized [12,13]. MD simulations are performed using Particle Mesh Ewald Molecular Dynamics (PMEMD), one of the AMBER modules. The different steps involved in setting up and executing an MD simulation are illustrated schematically, as shown in **Figure 3.3**.



**Figure 3.3.** Schematic flowchart of steps involved in MD Simulation (Taken from [7]).

### 3.1.1.8. Water molecule models:

Computer simulations of biomolecular systems play a crucial role in research, offering insights into the structure, dynamics, and energetics of biomolecules that are often beyond the reach of experimental techniques. However, various molecular models have been developed to describe water in MD simulations. These models are characterized by factors such as site count, polarization effects, and structural flexibility (rigid or flexible). The significance of water models lies in their ability to accurately predict the physical properties of liquid water based on a hypothetical computational representation. This is because such models replicate the unknown structural characteristics of liquid water. The trade-off in selecting a model involves balancing computational complexity, the system size that can be feasibly simulated within a reasonable time frame, and accuracy. Despite the continuous advancement in computational power, limitations imposed by system size, time constraints, and model complexity persist. Among the various water models, 3-site models are the most commonly used in MD simulations due to their simplicity, thermodynamic accuracy, computational efficiency, and logical structure. These models allow the three atoms of the water molecule to interact through three specific sites, each with a designated point charge. Out of all the atoms, oxygen is the only one with Lennard-Jones characteristics that allow for interaction. Lennard-Jones interactions play a vital role in defining molecular size; at very close distances, they exhibit repulsive behavior, preventing the structural collapse of the system due to electrostatic forces. It is extremely attractive yet non-directional at intermediate distances and competes with directionally attractive electrostatic interactions. Several well-known 3-site models include the simple point charge (SPC), extended simple point charge (SPC/E), and transferable intermolecular potential three-point (TIP3P) models [14]. Despite their differences, all these models maintain a geometry that aligns with the experimentally known structure of a water molecule. In this study, the TIP3P water model is employed. The O-H bond length ( $r_{\text{OH}}$ ) and H-O-H bond angle ( $\theta_{\text{HOH}}$ ) in the TIP3P model used here are found to be consistent with experimental gas-phase values, measuring 0.9572 Å and 104.52°, respectively. **Figure 3.4** depicts the structure of a simple TIP3P water model.

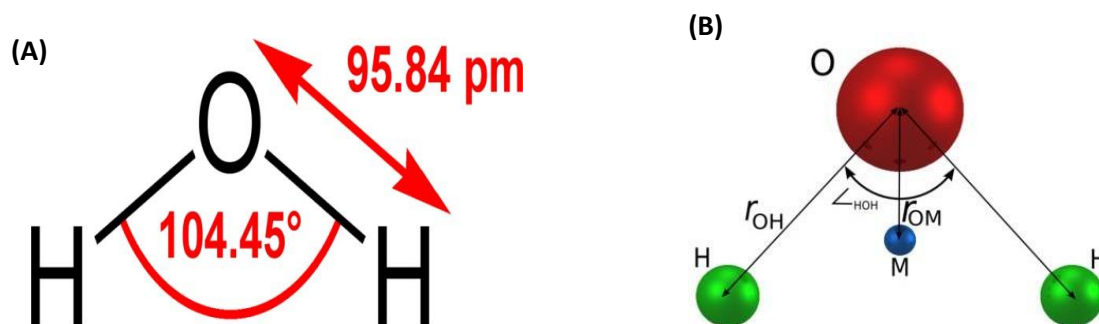


Figure 3.4. Schematic representation of TIP3P water model (Taken from [7])

### 3.1.1.9. Molecular Dynamics steps:

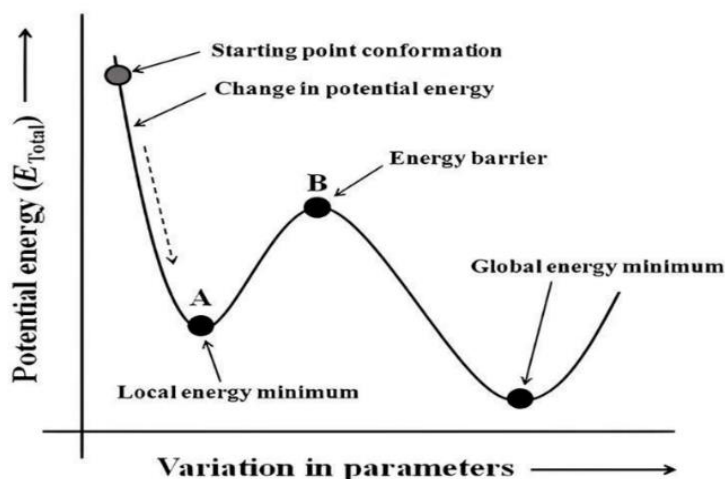
In order to execute the MD simulation for any given system, it involves the following four phases:

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

1. **Energy Minimization:** To initiate molecular dynamics, it is essential to identify a stable point or minimum on the potential energy surface by utilizing the force field assigned to the atoms of the system. Although these minimized structures represent the fundamental configurations associated with fluctuations occurring during dynamics, they serve as a valuable starting point for structural analysis [15,16]. At a point of minimal potential energy, the net force acting on each atom is zero. During both minimization and dynamics, constraints can be applied. These constraints may be derived from experimental data, such as nuclear Overhauser effects (NOEs) obtained from NMR experiments, or they may be enforced using a template to guide a ligand toward adopting a structure that closely resembles a target molecule. Minimization requires a function provided by the force field, along with an initial set of coordinates or an estimated structure. The magnitude of the first derivative helps determine both the direction and magnitude of the step (i.e., coordinate adjustment) necessary to approach a minimized configuration. Additionally, convergence can be rigorously defined in terms of both its magnitude and the size of the first derivative.

The process of molecular structure energy minimization consists of two main steps. The

first step involves constructing and evaluating an equation that expresses the system's energy as a function of its atomic coordinates for a given conformation. **Figure 3.5** illustrates the several stages that an energy-saving operation might go through.



**Figure 3.5.** A schematic representation of the different phases of a molecule during minimization of its energy (Adapted from [17])

Moreover, the energy minimization can be addressed using the following approach. It must be demonstrated that, given a function  $f$  and one or more independent variables ( $x_1, x_2, \dots, x_i$ ), the values of each independent variable can be found by taking the minimum value of  $f$ . For any variable, the first derivative of the function at its lowest point is 0, while its second derivatives are positive:

$$\frac{\partial f}{\partial x_i} = 0; \frac{\partial^2 f}{\partial^2 x_i} > 0 \dots\dots\dots (3.18)$$

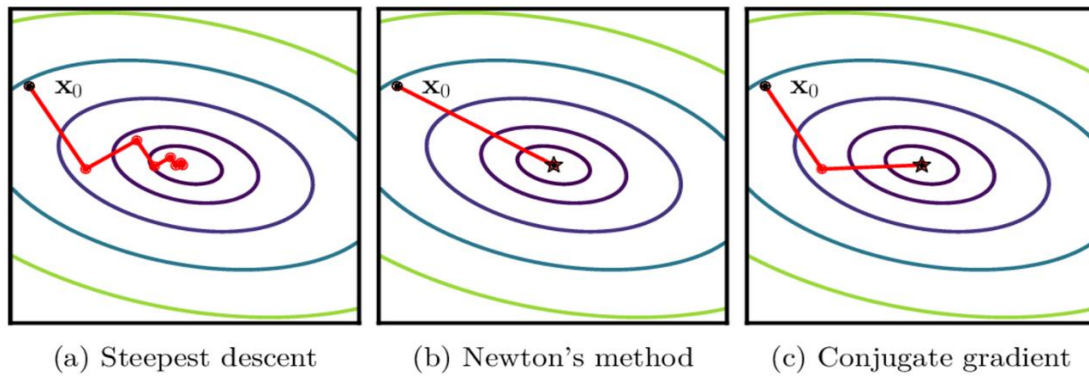
The direction of the initial derivative of the energy determines the location of the minimum, while the magnitude of the gradient indicates the steepness of the local slope. By enabling each atom to move in response to the applied force, the system's energy can be minimized when the force is equal to the negative gradient. Additionally, second derivatives provide insights into the curvature of the function, along with information that can help predict when the function will change direction, such as passing through a minimum or another stationary point.

In molecular modeling, the two most commonly employed techniques for first-order minimization are the steepest descent and conjugate gradient methods. When the derivatives are near zero, minimum energy converges. Prior to initiating an MD simulation, it is crucial to perform energy minimization on the structure to eliminate

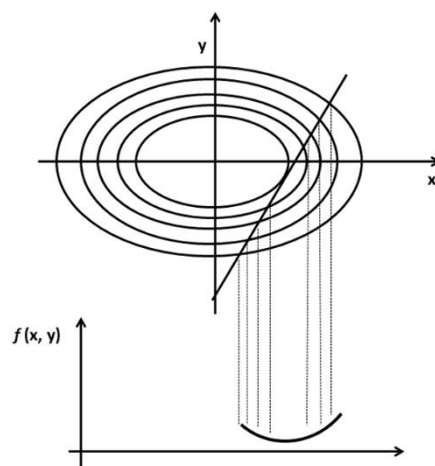
unfavorable interactions that could otherwise lead to structural distortions. The three primary minimization methods include Newton-Raphson, conjugate gradient, and steepest descent approaches.

**(i)The Steepest Descents Method:** This approach determines which path leads to the minimum by taking the first derivative, moving in a direction parallel to the net force. This direction is represented as a 3N-dimensional unit vector with 3N Cartesian coordinates. Once the direction of movement is established, the next step is to determine the distance to be travelled along the gradient.

**Figure 3.5** illustrates the two-dimensional energy surface [17]. Initially, the gradient follows a specific direction along a line. If we imagine slicing through the surface along this line, the function will first reach a minimum before increasing again, as depicted in **Figures 3.6 and 3.7** [18,19].



**Figure 3.6.** Comparison of three optimization methods—(a) Steepest Descent, (b) Newton's Method, and (c) Conjugate Gradient—on a quadratic function. The contour plots illustrate the convergence behaviour of each technique, showing different iterative steps towards the minimum (indicated by a star). (Adapted from [18])



**Figure 3.7.** A line search is used to locate the minimum in the function in the direction of the gradient (Taken from [19])

**(ii) Minimization of Conjugate Gradients:** In energy minimization, the conjugate gradient method produces a series of directions that avoid both the steepest descent and oscillatory behavior in narrow valleys. Although the directions in this approach are conjugate, the gradients remain orthogonal at each point. For a quadratic function with  $M$  variables, the minimum can be reached in  $M$  steps using a set of conjugate directions. Beginning at point  $\mathbf{x}_k$ , the conjugate gradient approach proceeds in the direction  $\mathbf{v}_k$ . The gradient at the point and the preceding direction vector  $\mathbf{v}_{k-1}$  are used to determine  $\mathbf{v}_k$  [20].

$$\mathbf{v}_k = \mathbf{g}_k + \gamma_k \mathbf{v}_{k-1} \dots \dots \dots (3.19)$$

Here,  $\mathbf{v}_k \rightarrow$  Conjugate Direction at Iteration  $k$ ,  $\mathbf{g}_k \rightarrow$  Gradient at Iteration  $k$  (Residual Vector) and  $\gamma_k \rightarrow$  Conjugate Coefficient (Update Factor)

The scalar constant  $\gamma_k$  in the above equation is provided by:

$$\gamma_k = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2} \dots \dots \dots (3.20)$$

**(iii) The Newton-Raphson method:** The Newton-Raphson method utilizes both first and second derivatives. The curvature is utilized not just to use gradient information, but also to predict where the direction will change along the function's gradient. This technique demands significant computational power to achieve energy minimization. If additional water molecules are needed to enhance the system's solubility before minimization, they are introduced. A preheated water reservoir at the same temperature is used for solvation, ensuring the system is fully immersed. The water box surrounds the entire system, eliminating any water molecules that come into direct contact with proteins. The mathematical model of Newton-Raphson equation is as follows:

$$\mathbf{r}_{min} = \mathbf{r}_0 - \mathbf{A}_0 \mathbf{A}^{2-1} \cdot \nabla(\mathbf{r}_0) \dots \dots \dots (3.21)$$

where  $\mathbf{A}_0$  is the matrix of second partial derivatives of the energy with respect to the coordinates at  $\mathbf{r}_0$  (also known as the Hessian matrix),  $\mathbf{r}_{min}$  is the anticipated minimum,  $\mathbf{r}_0$  is an arbitrary starting point, and  $\nabla(\mathbf{r}_0)$  is the gradient of the potential energy at  $\mathbf{r}_0$ .

**2. Heating:** During the heating phase, Newton's equations of motion, which describe the system's time evolution, are numerically integrated. Initially, each atom in the system is assigned velocities corresponding to a temperature of 0 K. At short, predefined intervals, new velocities are assigned, gradually increasing the temperature in small increments. The simulation continues until the desired temperature, typically 300 K, is reached. As the system heats up, structural stresses are alleviated, and force constraints on different



subdomains of the simulation are progressively removed. Heating dynamics are typically conducted under constant volume conditions (NVT ensemble).

3. **Equilibration:** As a system progresses from its initial configuration, it undergoes an equilibration phase, during which equilibrium is achieved. Ideally, equilibration should continue indefinitely or at least until the values of monitored properties stabilize. Key measured properties include thermodynamic variables such as energy, temperature, and pressure, as well as structural characteristics. In liquid-state simulations, the initial structure often resembles a solid lattice. To ensure accurate results, it is crucial to allow the lattice to "melt" before beginning the production phase. Order parameters can be used to determine when the liquid state has been fully reached. These parameters measure the degree of order within a system. In a crystal lattice simulation, atoms typically remain in fixed positions, maintaining a high level of order. Conversely, in a liquid state, the constituent species exhibit significant movement, leading to translational disorder. MD involves solving the equations of motion for an atomic system. The solution to these equations provides the molecular trajectories and temporal evolution of atomic movements. Depending on the simulation temperature, MD allows for overcoming energy barriers and exploring various conformational possibilities. To initiate MD, initial velocities must be assigned, which is achieved using a random number generator constrained by the Maxwell-Boltzmann distribution. **Equation 3.22** is the formula of the system's internal energy. **Equation 3.23** is the formula for kinetic energy.

$$U = \left(\frac{3}{2}\right) N k_B T \dots\dots\dots (3.22)$$

$$U = \left(\frac{1}{2}\right) N m v^2 \dots\dots\dots (3.23)$$

Here,  $U \rightarrow$  Represents the total kinetic energy of all molecules in a system

$N \rightarrow$  Number of Particles (Molecules)

$k_B \rightarrow$  Boltzmann's Constant

$T \rightarrow$  Absolute Temperature (Kelvin)

$m \rightarrow$  Mass of a Single Molecule

$v^2 \rightarrow$  Mean Squared Velocity

The temperature of the system can be determined by averaging the velocities of all atoms present. Once the initial velocities are assigned, the Maxwell-Boltzmann distribution can be maintained throughout the simulation. After energy minimization, the system's temperature is effectively considered to be at absolute zero (0 K). Before reaching the target temperature, the system must first undergo initialization. To achieve this, velocities



are assigned at a low temperature, and molecular dynamics is performed following the equations of motion. Over multiple iterations, the temperature is gradually increased using scaling methods. Under atomic constraints, a controlled heating process is carried out over 20 ps, progressively raising the temperature from 0 K to 300 K. Among various temperature scaling techniques, velocity scaling is the most commonly used. For equilibration time steps of 1 fs, a minimum simulation run of 5 ps (5000 time steps) is typically required, though runs of 10 to 20 ps are often preferred. Following the heating phase, dynamic equilibration is conducted for an additional 100 ps to ensure system stability.

- 4. Production dynamics:** During the dynamics phase, thermodynamic averages are calculated, and new configurations are explored to better understand the system's behaviour over time. This stage, commonly referred to as production dynamics, is where these calculations and structural assessments take place. Throughout this phase, various thermodynamic parameters, such as temperature, pressure, and energy fluctuations, can be evaluated to ensure the stability and accuracy of the simulation. The duration of a production run depends on the complexity of the system and the desired level of accuracy. Typically, production runs extend over a timescale ranging from hundreds of picoseconds to several nanoseconds. Longer simulations allow for more accurate statistical averaging and a better representation of the system's equilibrium properties. These extended simulations help capture molecular interactions, conformational changes, and other essential dynamic behaviours, making them a crucial step in molecular modelling studies.

### ***3.1.2. Potential of mean force (PMF):***

The potential of mean force (PMF) [21] is a fundamental concept used to describe change in free energy as a function of inter- or intra-molecular coordinates in molecular systems. The reaction coordinate, which can represent the distance between two atoms or the torsion angle of a bond, is intrinsically associated with its corresponding distribution function. When a system is immersed in a solvent, the PMF accounts for both the solvent effects and the intrinsic interactions between the two particles involved. Additionally, the rate constant of a process can be determined from its transition state. There are multiple approaches to calculating the PMF, each offering different levels of computational complexity and accuracy. The simplest form of PMF is defined as the change in free energy along a reaction coordinate, typically represented by the variation in separation distance ( $r$ ) between two interacting particles [7]. It is denoted as:

$$A(r) = -k_{BT} \ln g(r) + \text{constant} \dots \dots \dots (3.24)$$

Here,  $A(r)$  is Potential of Mean Force (PMF)

$k_B$  is Boltzmann Constant

$g(r)$  is Radial Distribution Function (RDF)

Due to the logarithmic relationship between the potential of mean force (PMF) and the radial distribution function, even a minor variation in free energy can correspond to a significant change in  $g(r)$ , potentially altering it by an order of magnitude from its most probable value. However, in regions where this discrepancy is substantial, the radial distribution function is often inadequately sampled by MD simulations. This insufficient sampling leads to inaccuracies in PMF estimations.

To address this limitation, various enhanced sampling techniques have been developed, with umbrella sampling (US) being one of the most widely utilized methods [7]. Umbrella sampling helps improve sampling efficiency by applying biasing potentials, thereby ensuring a more accurate representation of free energy landscapes.

### 3.1.2.1. Umbrella Sampling (US):

Umbrella sampling (US) addresses the sampling problem by constraining the system to specific regions of its conformational space. This is done by modifying the potential function, allowing for adequate sampling of otherwise unfavorable states. By introducing biasing potentials, this technique ensures that rare or high-energy conformations are properly explored, leading to more accurate free energy calculations and overcoming the limitations of insufficient sampling in MD simulations. The expression for the modification of the potential function is:

$$P'(r^N) = P(r^N) + W(r^N) \dots \dots \dots (3.25)$$

Where,  $P'(r^N)$  is the biased probability distribution of the system under the influence of an added weighting function,  $P(r^N)$  is the true (unbiased) probability distribution of the system without any external bias and  $W(r^N)$  is a weighting function, which takes a quadratic form:

$$W(r^N) = k_W(r^N - r_0^N)^2 \dots \dots \dots (3.26)$$

Where,  $k_W$  is the force constant (or biasing strength),  $r_N$  is the current configuration of the system. For those configurations that are distant from equilibrium state  $r_0^N$  the weighting

function shall be large, hence a simulation by using the modified energy function  $P'(r^N)$  will be biased away from the configuration  $r_0^N$ , along with some relevant ‘reaction coordinate’ (RC). The resulting distribution will, of course, be non-Boltzmann. Torrie and Valleau [22] introduced a method for extracting the corresponding Boltzmann averages from non-Boltzmann distributions. The result is:

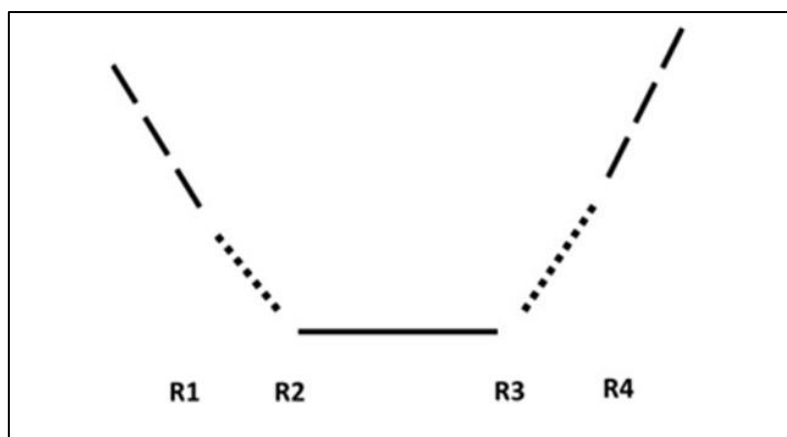
$$\langle A \rangle = \frac{\left\langle A(r^N) \exp\left[+\frac{W(r^N)}{k_B T}\right] \right\rangle_W}{\left\langle \exp\left[+\frac{W(r^N)}{k_B T}\right] \right\rangle_W} \dots\dots\dots (3.27)$$

Here,  $\langle A \rangle$  denoted the unbiased ensemble average of the observable  $A$  over the true probability distribution.  $A(r^N)$  denotes the observable of interest evaluated at configuration  $r^N$  and  $k_B$  is the Boltzmann constant.

Most of the time, an umbrella sampling calculation is done in stages. Each stage has a certain value for the coordinate and a certain value for the forcing potential  $W(r^N)$ . But if the forcing potential is very large, then the denominator in **Equation 3.27** is dominated by contributions from only a few configurations with especially high values of  $\exp [W(r^N)]$  and the average takes too long to converge.

### 3.1.2.2. Running the umbrella sampling calculations:

Once a stable starting structure is obtained, MD simulations can be performed on individual umbrella windows. An important consideration when determining the number of windows is ensuring that adjacent windows have sufficient overlap. This means that the configurations sampled in window 1 should partially coincide with those in window 2, and so forth. Similarly, the selection of the force constant plays a crucial role that it must be sufficiently large to confine sampling within a specific region of phase space, yet not excessively strong, as this could result in overly narrow windows that fail to overlap adequately (**Figure 3.8**).



**Figure 3.8.** Working principle of Umbrella Sampling. (Taken from [22]).

"\" = lower bound linear response region

"/" = lower bound linear response region

"..." = parabola

"\_" = flat region

Typically, the size of the windows and the constraints can be adjusted based on their position along the pathway. It is essential to ensure that each window undergoes sufficient simulation time to achieve convergence in sampling. To specify the harmonic restraint a reference file is employed where R1, R2, R3, R4 define a flat-welled parabola which becomes linear beyond a specified distance. Essentially between R1 and R2 it will be harmonic with force constant  $Rk_2$ , between R2 and R3 it will be flat and between R3 and R4 it will be harmonic with force constant  $Rk_3$ .

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

The Weighted Histogram Analysis Method (WHAM) [23] is an advanced extension of the conventional umbrella sampling (US) technique, offering several advantages over the standard approach. One of its key benefits is the enhanced connection between multiple simulations, allowing for more accurate estimations of free-energy differences by enabling multiple overlaps of probability distributions. In cases where three or more distributions contribute to the overlap region, the conventional method of generating a single distribution function by enforcing coincidence at a specific point fails to yield unique free-energy values.

The WHAM method incorporates an error estimation feature, providing researchers with objective guidance on the optimal placement and duration of subsequent simulations to improve accuracy. By considering all simulations that produce overlapping distributions, the WHAM approach effectively links these simulations through their shared histograms. Furthermore, the WHAM equations facilitate the construction of potential of mean force (PMF) profiles and free-energy calculations as a function of coupling parameters and/or temperature. This is particularly advantageous, as simulations performed at various temperatures can enhance conformational sampling, with results being interpolated or extrapolated to the target temperature [24].

### ***3.1.3. Molecular Docking:***

Predicting protein-protein and protein-small molecule interactions through computational methods remains one of the most challenging tasks in structural biology. Reliable and precise interaction predictions can significantly benefit various biological research fields, both in academia and industry. The primary challenge in protein-protein docking lies in correctly associating two interacting molecules, which relies on accurately identifying the residues involved in the target interaction. Several docking methodologies have been developed over time [25-29]. However, only a limited number of these algorithms are freely available for online use. 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.

Molecular docking is a widely used approach for modeling interactions at the atomic level, whether between a protein and a small molecule or between two proteins. This technique helps in understanding how small molecules behave at the target protein's binding site and in identifying the interface residues involved in protein-protein interactions [30]. The docking process consists of two primary stages. In the first stage, the ligand's position within the binding site is determined. In the second stage, the ligand conformers are ranked using a scoring function based on binding affinity. For effective docking, the scoring function must successfully prioritize the experimental binding mode as the most favourable among all generated conformations.

In this Thesis, we utilized the PatchDock server [31] for protein-small molecule docking, the HADDOCK server [32,33] for protein-protein docking, and the ClusPro server [34] for protein-peptide docking.

### 3.1.3.1. PatchDock:

PatchDock is an online docking tool designed for rigid docking of molecules, including protein-protein and protein-drug interactions, while considering surface adaptations during intermolecular penetration [31]. Based on geometric molecular docking, it identifies transformations with high shape complementarity, minimizing steric clashes while maximizing interface regions. The method analyses concave, convex, and flat patches using the Connolly dot surface representation and matches complementary patches to generate docking configurations. Each candidate solution is evaluated based on geometric compatibility and atomic desolvation energy, with redundancy eliminated via the root mean square deviation (RMSD) clustering algorithm. PatchDock's efficiency stems from its rapid transformation search, which leverages local feature matching instead of exhaustive scanning of the six-dimensional transformation space. Advanced data structures and spatial pattern recognition techniques, such as geometric hashing and pose clustering, further enhance computational speed. The docking process is highly optimized, with a runtime of approximately 10 minutes for standard protein inputs ( $\approx 300$  amino acids). Rooted in the Kuntz algorithm for local shape feature matching [35], PatchDock ensures accurate docking by identifying high-probability molecular surface regions at binding sites, making it effective for docking large proteins with small drug molecules.

#### Input for PatchDock:

The docking algorithm requires two molecules in PDB format as input. These molecules can either be retrieved directly from the Protein Data Bank or uploaded by the user. In the latter case, the user only needs to provide the PDB code. If docking involves specific chains, the relevant chain ID(s) must be specified. Additionally, the user must enter a valid email address to receive the docking results. In addition, the docking request form has four more optional fields.

(i) **Clustering RMSD:** This positive value represents the root mean square deviation (RMSD) clustering radius in angstroms. It is primarily used in the final clustering step of the algorithm. The clustering RMSD value determines the minimum distance between any two solutions in the output. By default, this parameter is set to 4 Å.

(ii) **Complex Type:** PatchDock offers different parameter settings to optimize docking for various types of molecular interactions. If this field is left unspecified,

the software applies the default configuration. For enzyme-inhibitor complexes, the algorithm restricts the search space to enzyme cavities. In the case of antibody-antigen complexes, it focuses on the complementarity-determining regions of the antibody. For protein-small ligand docking, the parameter settings are adjusted specifically for small molecules.

**(iii) Potential Binding Sites for Ligands and (iv) Receptors:** If biological data indicate specific residues as part of a potential binding site, users can input this information into the algorithm. The list of residues associated with possible binding sites must be provided in an uploaded file. The format of this file should match the PDB structure, with each line containing the chain ID and residue index separated by a space. **Figure 3.9** illustrates the PatchDock user interface.

**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 [31]).

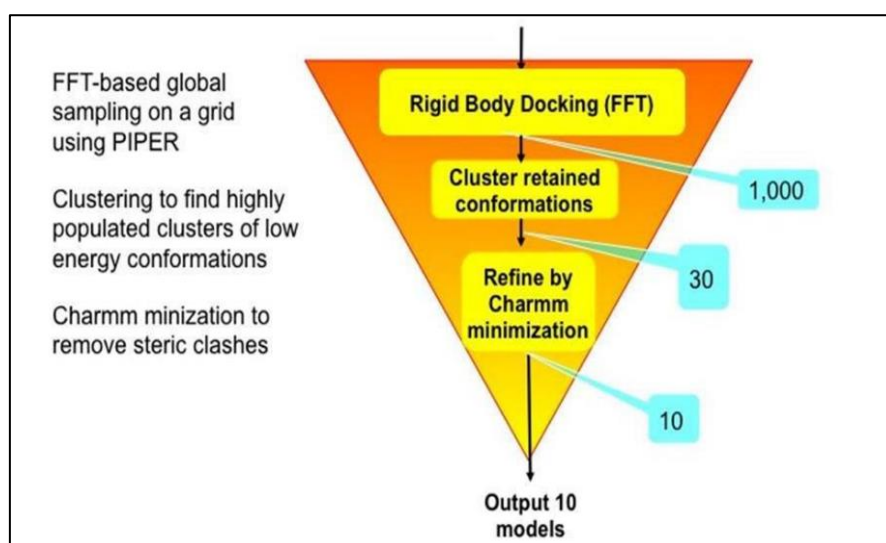
### Output for PatchDock:

The PatchDock server automatically generates the top 20 docking solutions, which are accessible via a link sent to the user's email (<https://bioinfo3d.cs.tau.ac.il/PatchDock/>). These solutions are typically displayed in a table format, showing key parameters such as the geometrical shape complementarity score, interface size, desolvation energy, and the rigid transformation details. Each docking solution includes a corresponding PDB file, which users can view or download via the provided URL. To enhance the accuracy of docking results, structures obtained from PatchDock can be further refined using FireDock, a web server designed for rapid optimization of molecular docking interactions

### 3.1.3.2. ClusPro web server:

ClusPro, a web-based docking server, was first introduced in 2004 [36, 37] and has since undergone extensive modifications and enhancements [38-40]. This platform enables the direct docking of two interacting proteins [34] by requiring input in the form of two PDB-format protein files. The docking process on ClusPro involves three main computational steps:

- (i) **Rigid body docking:** The server samples billions of possible conformations.
  - (ii) **Clustering based on RMSD:** Among the sampled structures, the 1000 lowest-energy conformations are grouped to identify the largest clusters, representing the most likely complex models.
  - (iii) **Energy minimization:** The selected structures are refined to enhance their accuracy.
- During the rigid body docking stage, ClusPro employs PIPER [41], a docking tool that utilizes the Fast Fourier Transform (FFT) correlation technique to efficiently explore molecular interactions (**Figure 3.10**).



**Figure 3.10.** Representation of the ClusPro algorithm, the number of structures retained after each step is shown in a blue box. (Taken from [34]).

### 3.1.3.3. Haddock server:

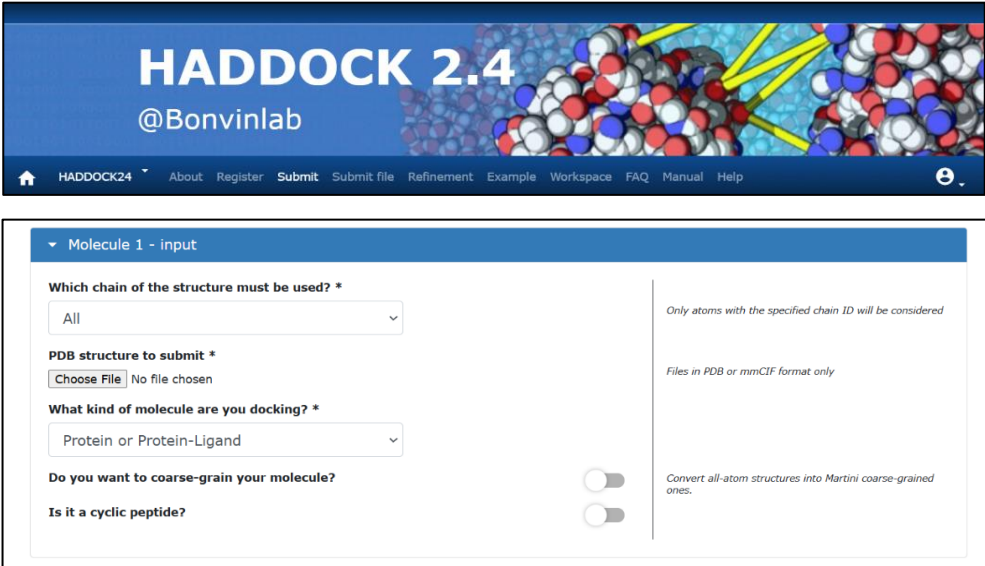
HADDOCK is a widely used docking program that employs a data-driven approach, integrating various types of experimental data to enhance docking accuracy. Its primary web interface is designed for ease of use, requiring only the structural input of individual



molecular components and a list of interacting residues. For advanced users, additional web interfaces provide access to a broader range of experimental data and allow customization of the docking process. The HADDOCK server operates on a dedicated computational cluster and utilizes the e-NMR GRID infrastructure for enhanced performance [32].

During docking, HADDOCK incorporates non-structural experimental data to guide the process through multiple stages, including rigid-body energy minimization, semi-flexible refinement, and water-based refinement. These refinement stages introduce flexibility, enabling movement in all DNA nucleotides and protein residues at the predicted interaction interface, thereby improving docking accuracy [33]. The Haddock user interface is shown step by step in **Figure 3.11**.

**(A) Input (For molecule 1):**



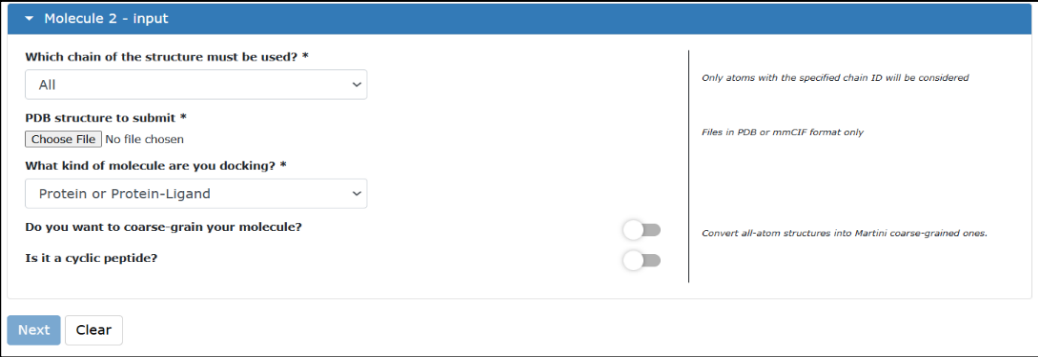
The screenshot shows the HADDOCK 2.4 web interface. The header includes the logo and navigation links: Home, HADDOCK24, About, Register, Submit, Submit file, Refinement, Example, Workspace, FAQ, Manual, and Help. The main content area is titled 'Molecule 1 - input' and contains the following fields:

- Which chain of the structure must be used? \***: A dropdown menu with 'All' selected.
- PDB structure to submit \***: A 'Choose File' button and the text 'No file chosen'.
- What kind of molecule are you docking? \***: A dropdown menu with 'Protein or Protein-Ligand' selected.
- Do you want to coarse-grain your molecule?**: A toggle switch.
- Is it a cyclic peptide?**: A toggle switch.

On the right side, there are two informational notes:

- 'Only atoms with the specified chain ID will be considered'
- 'Files in PDB or mmCIF format only'
- 'Convert all-atom structures into Martini coarse-grained ones.'

**(B) Input (For molecule 2):**



The screenshot shows the HADDOCK 2.4 web interface for Molecule 2. The header is the same as in (A). The main content area is titled 'Molecule 2 - input' and contains the same fields as in (A):

- Which chain of the structure must be used? \***: A dropdown menu with 'All' selected.
- PDB structure to submit \***: A 'Choose File' button and the text 'No file chosen'.
- What kind of molecule are you docking? \***: A dropdown menu with 'Protein or Protein-Ligand' selected.
- Do you want to coarse-grain your molecule?**: A toggle switch.
- Is it a cyclic peptide?**: A toggle switch.

On the right side, there are the same informational notes as in (A):

- 'Only atoms with the specified chain ID will be considered'
- 'Files in PDB or mmCIF format only'
- 'Convert all-atom structures into Martini coarse-grained ones.'

At the bottom left, there are 'Next' and 'Clear' buttons.

**Figure 3.11.** The Haddock user interface: (A) Entries required for the molecule 1 and (B) Entries required for the molecule 2

### 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 determine the binding free energy (BFE) of small ligands (small molecules) with receptor proteins or protein-protein complexes, researchers commonly employ the MM-PBSA and MM-GBSA methods [42-44]. These techniques, which rely on MD simulations of the protein-ligand complex, are widely recognized for their high reproducibility and accuracy.

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:

$$\Delta G = \Delta H - T\Delta S \dots\dots\dots (3.29)$$

wherein,  $\Delta H$  is the enthalpy,  $\Delta S$  represents entropy and  $T$  is the temperature of the system at 300 Kelvin.

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

$$\Delta G_{\text{bind}}/\Delta G_{\text{binding}} = \Delta G_{\text{complex, solv}} - (\Delta G_{\text{protein, solv}} + \Delta G_{\text{ligand, solv}}) \dots\dots (3.30)$$

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

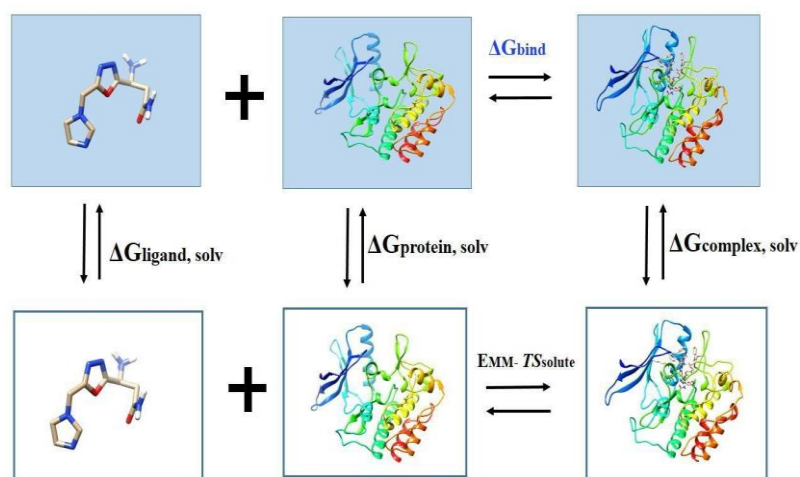
$$\Delta G_{\text{bind}}/\Delta G_{\text{binding}} = [E_{\text{MM}} + \Delta G_{\text{solvation}}] - T\Delta S_{\text{total}} \dots\dots\dots (3.31)$$

$$E_{\text{MM}} = E_{\text{intra}} + E_{\text{elec}} + E_{\text{vdW}} \dots\dots\dots (3.32)$$

$$E_{\text{internal}} = E_{\text{bond}} + E_{\text{angle}} - E_{\text{torsion}} \dots \dots \dots (3.33)$$

$$\Delta G_{\text{solvation}} = \Delta G_{\text{PB/GB solvation-elec}} + \Delta G_{\text{SASA,nonpolar}} \dots \dots \dots (3.34)$$

The molecular mechanics (MM) energy, denoted as  $E_{\text{MM}}$ , represents the energy derived from the force field in the absence of a solvent. The internal energy,  $E_{\text{internal}}$  (intra) comprises three intramolecular components:  $E_{\text{bond}}$ ,  $E_{\text{angle}}$ , and  $E_{\text{torsion}}$ . The intermolecular interaction energies include  $E_{\text{elec}}$ , which accounts for electrostatic interactions, and  $E_{\text{vdW}}$ , which represents van der Waals forces.  $\Delta G_{\text{solvation}}$  is the solvation free energy which comprises of the  $\Delta G_{\text{PB/GB, solvation-elec}}$  which denotes the electrostatic contribution to solvation free energy, calculated using GB/PB and the nonpolar solvation energy ( $\Delta G_{\text{SASA,nonpolar}}$ ) is determined based on the solvent-accessible surface area (SASA). Additionally,  $T$  and  $S_{\text{solute}}$  represent the temperature and the entropy of the solute, respectively. 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 [45]).

The electrostatic solvation energy can be determined using the PB and GB methods. The dielectric constants were set to 1 for the solute (inner region) and 80 for the solvent (water). The atomic charges and radii used in the calculation were consistent with those applied in MD simulations. From solvent accessible surface-area, the non-polar contribution ( $\Delta G_{\text{SASA}}$ ) to the solvation free energy was calculated by means of **Equation 3.35**.

$$\Delta G_{\text{SASA}} = \gamma \times \text{SASA} + b \dots \dots \dots (3.35)$$

Here, SASA is the solvent-accessible surface-area and  $\gamma$  is surface tension parameter. ' $\gamma$ ' is set as 0.005 kcal (mol<sup>-1</sup>Å<sup>-2</sup>) for PB and 0.0072 kcal (mol<sup>-1</sup>Å<sup>-2</sup>) for GB. ' $b$ ' is a parameterized value set as 0.92 kcal/mol for PB and 0 kcal/mol for the GB method. The probe radius of the solvent is set to 1.4 Å.

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

$$S = S_{\text{trans}} + S_{\text{rot}} + S_{\text{vib}} \dots \dots \dots (3.36)$$

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

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

According to the work of Gohlke et al. [46,57], AMBER14 incorporates various techniques to break down estimated free energy into specific residue contributions using either the Generalized Born (GB) or Poisson–Boltzmann (PB) models. Interactions for each residue can be analysed by considering only those in which at least one atom of the residue is involved, a process referred to as per-residue energy decomposition. In contrast, pairwise decomposition examines interactions between specific residue pairs by including only interactions where at least one atom from each residue is involved. These decomposition methods provide valuable insights into key interactions during free energy calculations [46,47].

However, the dielectric boundary between the protein and the surrounding solvent is inherently nonlocal, as it is influenced by the overall spatial arrangement of atoms. As a result, solvation free energies computed using the Generalized Born (GB) and Poisson–Boltzmann (PB) models are not strictly pairwise decomposable. Therefore, caution should be exercised when interpreting results from free energy decomposition.

Using the Per-Residue Decomposition Method via Python Script MMPBSA.py [48], we can determine the partial BFE contribution towards the amino acid residue Y ( $\Delta G^Y_{\text{bind}}$ ). The contribution of each residue to the overall BFE may be calculated using a per-residue based decomposition method [49-52]. We divide the terms in **Equation 3.35** first to get the  $\Delta G^Y_{\text{bind}}$  into its atomic contribution. **Equation 3.37** is used to calculate each atom's ( $a$ ) contribution to the overall electrostatic interaction energy:

$$E_{elec}^a = \frac{1}{2} \sum_{b \neq a} \frac{q_a q_b}{r_{ab}} \dots \dots \dots (3.37)$$

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 3.38** illustrates how the non-polar effects of solvents on BFE are expressed using each atom's (a) SASA

$$\Delta G_{nonpolar,solv}^a = \gamma \{ (SASA^{a,complex} - (SASA^{a,protein} + SASA^{a,ligand})) \} \dots \dots \dots (3.38)$$

Where,  $\Delta G_{nonpolar,solv}^a$ : The nonpolar solvation free energy change.

$SASA^{a,complex}$ : The solvent-accessible surface area of the protein-ligand complex after binding.

$SASA^{a,protein}$ : The solvent-accessible surface area of the unbound protein (before binding).

$SASA^{a,ligand}$ : The solvent-accessible surface area of the unbound ligand (before binding).

$\gamma$  is surface tension parameter and it is set to 0.0072 kcal/mol/Å<sup>2</sup> 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:

$$\Delta G_{elec,solv}^a = -\frac{1}{2} \sum_b q_a q_b \left( \frac{1}{f(r_{ab}, R_a, R_b)} \right) - \frac{1}{\epsilon_\omega r_{ab}} \dots \dots \dots (3.39)$$

$$f(r_{ab}, R_a, R_b) = \sqrt{r_{ab}^2 + R_a R_b e^{-r^2/ab} / (4R_a R_b)} \dots \dots \dots (3.40)$$

Here,  $\Delta G_{elec,solv}^a$  is the electrostatic solvation free energy contribution of atom a,  $q_a, q_b$  denotes the partial charges of atoms a and b,  $r_{ab}$  is the direct Coulomb distance between atoms a and b,  $R_a, R_b$  is the effective Born radii,  $\epsilon_\omega$  is the dielectric constant of the solvent and  $f(r_{ab}, R_a, R_b)$  denotes the Generalized Born function,

Evaluating the partial BFE contribution to amino acid residue Y using these contributions to each atom results in **Equation 3.41**, which is shown below:

$$\Delta G_{bind}^Y = \sum_{a \in Y} (E_{elec}^a + E_{vdw}^a + \Delta G_{nonpolar,solv}^a + \Delta G_{elec,solv}^a) \dots \dots \dots (3.41)$$

### 3.1.5. Binding free energy calculation using webserver

#### 3.1.5.1. Prodigy webserver:

PROtein binDIng enerGY prediction (PRODIGY) is a web server designed to estimate the binding affinity of protein–protein complexes based on their three-dimensional (3D) structure

[53]. This server utilizes a straightforward yet highly effective predictive model that relies on intermolecular contacts and features derived from the non-interface surface.

To use PRODIGY, users must provide the 3D structure of a protein–protein complex through one of the following methods:

1. Uploading the structure in PDB or mmCIF format.
2. Automatically retrieving it from the Protein Data Bank.
3. Uploading a compressed archive file (.tar, .tgz, .zip, .bz2, .tar.gz) to analyse multiple structures simultaneously (up to a 50 MB limit).

Additionally, users need to specify the chain identifiers of the interacting molecules. The server also allows setting the temperature for calculating the dissociation constant (default: 25°C) and provides an option to enter an email address, where a link to the results page will be sent. The output include:

1. the predicted value of the binding free energy ( $\Delta G$ ) in kcal/mol;
2. the predicted value of the dissociation constant ( $K_d$ ) in M calculated from  $\Delta G = RT \ln(K_d)$  where  $R$  is the idea gas constant (kcal/K/mol),  $T$  the temperature (K);
3. the number and type of intermolecular contacts within the 5.5 Å distance cutoff;
4. the percentages of charged and polar amino-acids on the non-interacting surface;
5. a downloadable table (.txt) of all residues occurring at the interface;
6. a compressed file with all the result files.

The PRODIGY server operates efficiently, completing predictions within seconds even for the largest protein–protein complexes [53]. An illustration of the input and output pages of PRODIGY is presented in **Figure 3.13**.

#### (A) Input

The screenshot displays the PRODIGY web interface. At the top, there is a navigation bar with links: Home, Manual, Method, Dataset, Example, and Reference. Below this, a banner reads "We are testing out a new interface for PRODIGY!" and "The new interface contains all features of the current PRODIGY webserver and new ones, such as automatic identification of ligands and chains! If you would like to try it out, please click here." Below the banner, there are three tabs: PRODIGY (selected), PRODIGY-LIGAND, and PRODIGY-CRYSTAL. The main form area contains the following fields and options:

- Structure(s)\*:** A text input field for PDB\_ID, an "OR" button, and a "Choose File" button (labeled "No file chosen").
- Interactor 1\*:** A text input field for ID\_chain(s) - Eg. A or A,B,...
- Interactor 2\*:** A text input field for ID\_chain(s) - Eg. C or C,D,...
- Temperature (in °C)\*:** A text input field with the value 25.0.
- Job ID:** A text input field with the placeholder "Custom tag - Optional".
- ☐ Notify me by email upon job completion!
- ☐ I'm not a robot (with a CAPTCHA image).
- Buttons: "Load Sample Data" and "Submit Prodigy".

At the bottom, a note states: "Once you click on the button, your job will be processed. If you have provided an email address, the link to the output page will be emailed to you."

**(B) Output**

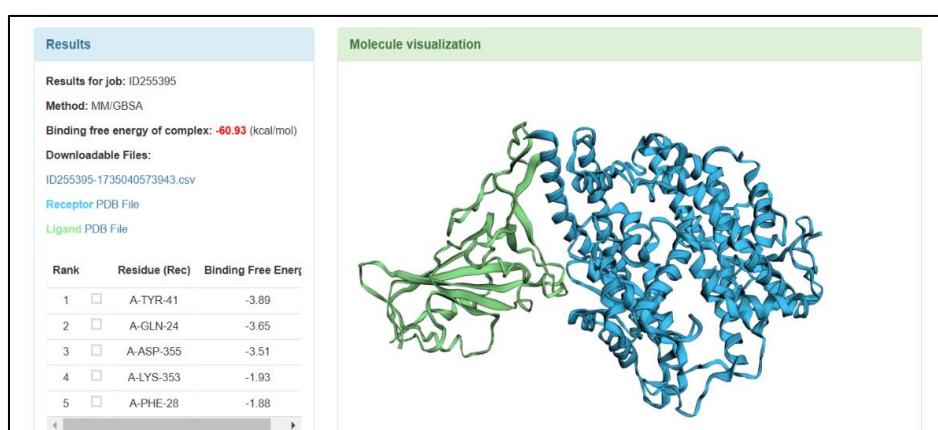
BINDING AFFINITY AND $K_D$ PREDICTION										
Protein-protein complex	$\Delta G$ (kcal mol <sup>-1</sup> )	$K_d$ (M) at °C	ICs charged-charged	ICs charged-polar	ICs charged-apolar	ICs polar-polar	ICs polar-apolar	ICs apolar-apolar	NIS charged	NIS apolar
model_000_00	-18.5	2.6e-14	7	16	44	9	41	40	22.75	36.77

**Figure 3.13.** Example output of PRODIGY showing the (A)input and (B) output page.

### 3.1.5.2. HawkDock webserver:

Protein-protein interactions (PPIs) are crucial for various cellular functions; however, accurately predicting their three-dimensional structures remains a challenging task. To address this, HawkDock, a freely accessible web server was developed to predict and analyse PPI structures. The HawkDock server seamlessly integrates the ATTRACT docking algorithm, the HawkRank scoring function, and MM/GBSA free energy decomposition analysis into a comprehensive, multi-functional platform [54].

MM/GBSA is utilized to estimate binding free energy and decompose the contributions of individual residues to the overall binding free energy of a protein-protein complex, facilitating structural analysis. When predicting key residues using the MM/GBSA approach, the most significant residues involved in PPIs were identified within the top 10 residues in approximately 81.4% of predicted models and 95.4% of crystal structures in the benchmark dataset [54]. An example of output page of HawkDock is shown in **Figure 3.14**.



**Figure.3.14.** Example output of HawkDock for the Spike-ACE2 complex



### ***3.1.6. Prediction of protein-protein / protein-ligand interaction using in-silico tools:***

Protein-protein interactions (PPIs) play a vital role in numerous cellular physiological processes and are also linked to various diseases [55-57]. Since PPIs exhibit considerable variability, a detailed examination of protein interfaces is crucial. The size of the protein contact area influences both the specificity and stability of these interactions. Protein-protein interaction sites are primarily formed through van der Waals interactions among nonpolar residues, driven by hydrophobic effects [58] and significant shape complementarity [59-61]. Additionally, electrostatic interactions between the interacting protein surfaces plays a crucial role in promoting the formation and stabilization of the complex. As the protein surfaces interact, the complex becomes more stable.

Accurately predicting PPIs is essential for advancing drug discovery and developing new therapeutic strategies. While external substances can interfere with PPIs, these interactions are fundamental to numerous biological processes, both beneficial and harmful. In modern drug discovery, the two key steps involve selecting a viable pharmacological target, gathering comprehensive information about it, and designing an appropriate ligand [62]. Understanding PPIs, therefore, facilitates the development of modulators specifically designed to target protein complexes effectively. In order to predict the interactions between protein-protein or proteins-ligands, this thesis employs three different tools namely PDBsum, LigPlot, and Mapiya server.

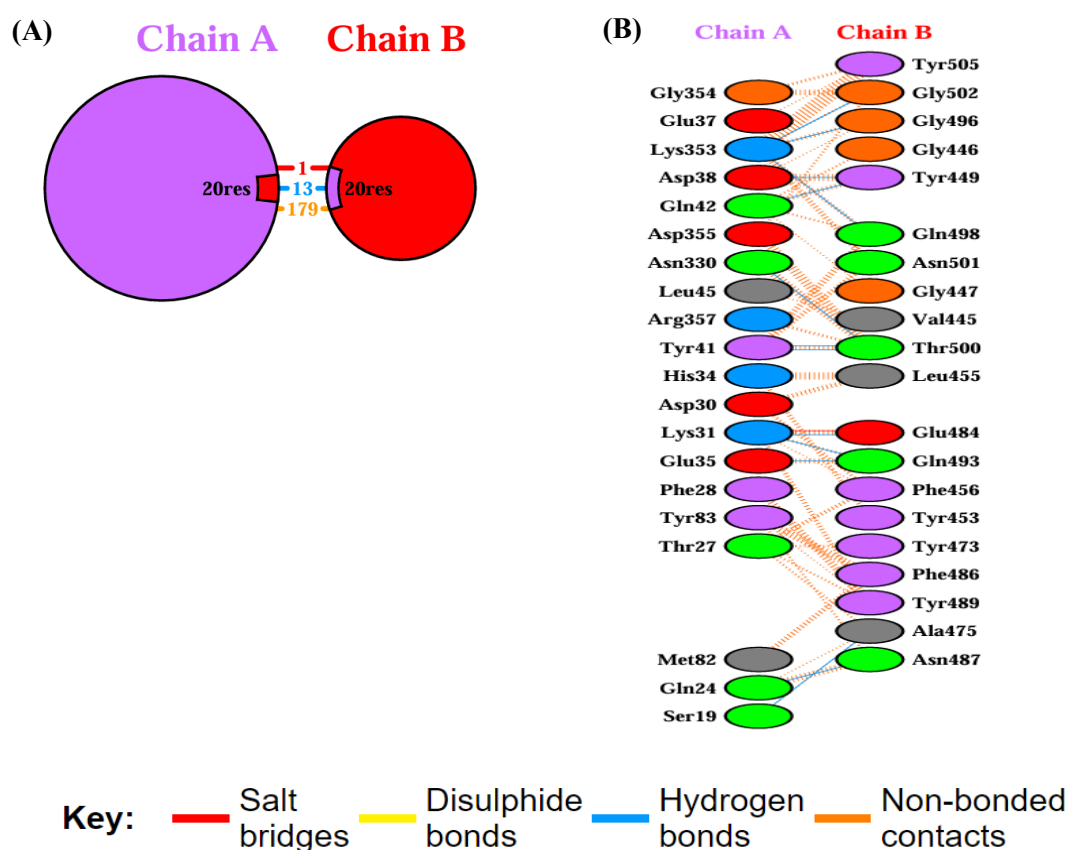
#### **3.1.6.1. PDBsum server:**

PDBsum is a web-based database [63,64] established in 1995, providing structural insights into all experimentally validated models published by the Protein Data Bank (PDB) [65,66]. The PDBsum server also offers a schematic representation of intermolecular interactions [63], aiming to visually depict structural data for each three-dimensional (3D) model as comprehensively as possible. It generates graphical diagrams of molecular components within each PDB entry, including protein, DNA, and RNA chains, ligands, and metal ions, along with their interactions. Over time, the server has been enhanced with new features, expanding its functionality.

PDBsum was the first web-based platform to leverage emerging World Wide Web technologies to create a structured catalog of PDB information. Its primary goal was to serve as an extensive visual compendium of protein structures and their complexes. Originally developed at University College London in 1995, the server contained numerous structural representations not readily available elsewhere. In 2001, PDBsum was transferred to the European



Bioinformatics Institute, where it has since been continuously updated and integrated with other resources. Complementary resources include the Research Collaboratory for Structural Bioinformatics (RCSB) PDB (<http://www.rcsb.org>), and PDB Europe (PDBe) (<https://www.ebi.ac.uk/pdbe>) [65]. These platforms offer extensive coverage of all PDB entries, along with powerful tools for structural analysis. The results from the PDBsum server summarizes the number of interactions across any chosen interface (**Figure 3.15A**), and the **Figure 3.15B** plot illustrates a depth information about which residues are actually interacting across that interface.



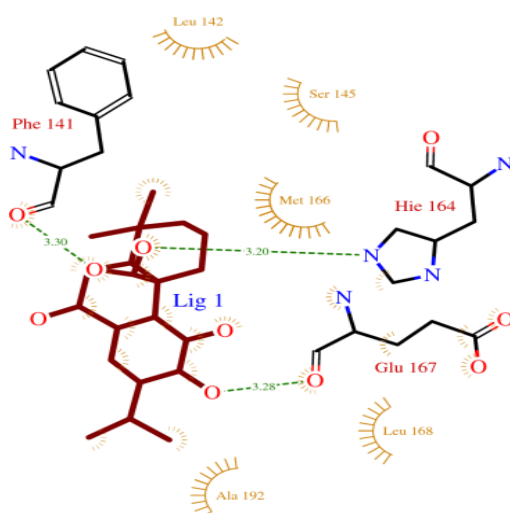
**Figure 3.15.** Protein–protein interaction diagrams in PDBsum (A) A schematic diagram showing the numbers of interactions across one of the interfaces, namely the A–B protein interface, and the numbers of residues involved in the Spike (Delta-plus)-ACE2 complex. (B) Detail of the individual residue–residue 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 Chapter 7B).

### 3.1.6.2. LigPlot Tool:

LigPlot+ [67] is a widely used program that generates a two-dimensional (2D) schematic representation of protein-ligand interactions, including hydrogen bonds and non-bonded

interactions between the ligand and surrounding protein residues. When a protein-ligand complex in PDB format is uploaded, LigPlot+ analyzes the molecular interactions and visually maps them for easier interpretation. A standalone version of LigPlot+ is also available for download and installation, allowing users to generate detailed protein-ligand interaction profiles independently. The program produces a PostScript file that displays intermolecular interactions such as hydrogen bonding, hydrophobic interactions, and atom accessibility, along with their interaction strengths. These representations can be viewed in either color or black and white, depending on user preferences. One of the key advantages of LigPlot+ is its universal applicability, as it supports interaction mapping for all ligands without restrictions. This makes it an essential tool for structural biology and drug discovery studies.

**Figure 3.16** illustrates an example of LigPlot+ results for the Mpro-Rosmanol complex, depicting the hydrophobic interactions and hydrogen bonds.



**Figure 3.16.** LigPlot analysis showing hydrophobic interaction, hydrogen bonds between Main Protease and Rosmanol molecule. The hydrogen bond interactions are represented by dashed lines. The amino acid residues involved in the hydrophobic interactions are shown as starbursts (Taken from Chapter 5)

### 3.1.6.3. Mapiya - contact maps:

The study of biomolecular interactions is fundamental to understanding how biomolecules function. Contact maps serve as an effective tool for deciphering the complex network of biomolecular interactions [68]. Over time, computer-aided visualization tools for contact maps have been developed, facilitating their application in validating contact prediction

methods [69,70], reconstructing three-dimensional (3D) protein structures [71-74], and performing surface-based interaction analyses [75,76].

Mapiya is an interactive and user-friendly web service designed for biomolecular structure analysis, specifically for proteins and their complexes, including protein-protein and protein-nucleic acid (RNA/DNA) interactions. It integrates several features from existing tools while offering additional structure- and sequence-based analyses. Mapiya's core functionality revolves around contact maps, providing multiple visualization options for detailed molecular interaction analysis [77,78].

To use Mapiya, the only required input is a biomolecular structure file in PDB format. This is marked as the fourth step on the main Mapiya page. The first three steps are optional and allow users to customize default settings and perform additional calculations. These optional steps include:

- (i) **'Select options'** – enabling modification of default settings, such as the contact cutoff value;
- (ii) **'Fix structure'** – utilizing the PDBfixer package to correct structural issues and fill gaps in PDB input files;
- (iii) **'Biological assembly'** – reconstructing biological assemblies based on information from the input PDB file.

The primary steps involved in using the Mapiya server to analyse biomolecular interactions of proteins and their complexes (protein-protein, and protein-RNA/DNA) are depicted in **Figure 3.17**.

The screenshot displays the MAPIYA Contact Map Server interface. It features a top navigation bar with the logo and title. Below this, there are three main sections: '1. Select options', '2. Fix structure', and '3. Biological assembly'. Each section contains a brief description of its function. To the right of these sections is a '4. Upload file' section. At the bottom, there is a large '4. Drop files here' area with a 'Download structure from the rcsb.org' button. The interface is clean and user-friendly, with clear instructions and a logical flow.

**Figure 3.17.** Showing the main steps of Mapiya server to analyze the biomolecular interactions of proteins and their complexes (protein-protein, and protein with RNA and/or DNA)

Mapiya is capable of analyzing both intramolecular and intermolecular interactions among various biomolecules, including proteins, peptides, and nucleic acids (RNA and DNA). Users can upload a structural file directly from their local system or retrieve it from the Protein Data Bank (PDB) for analysis [77].

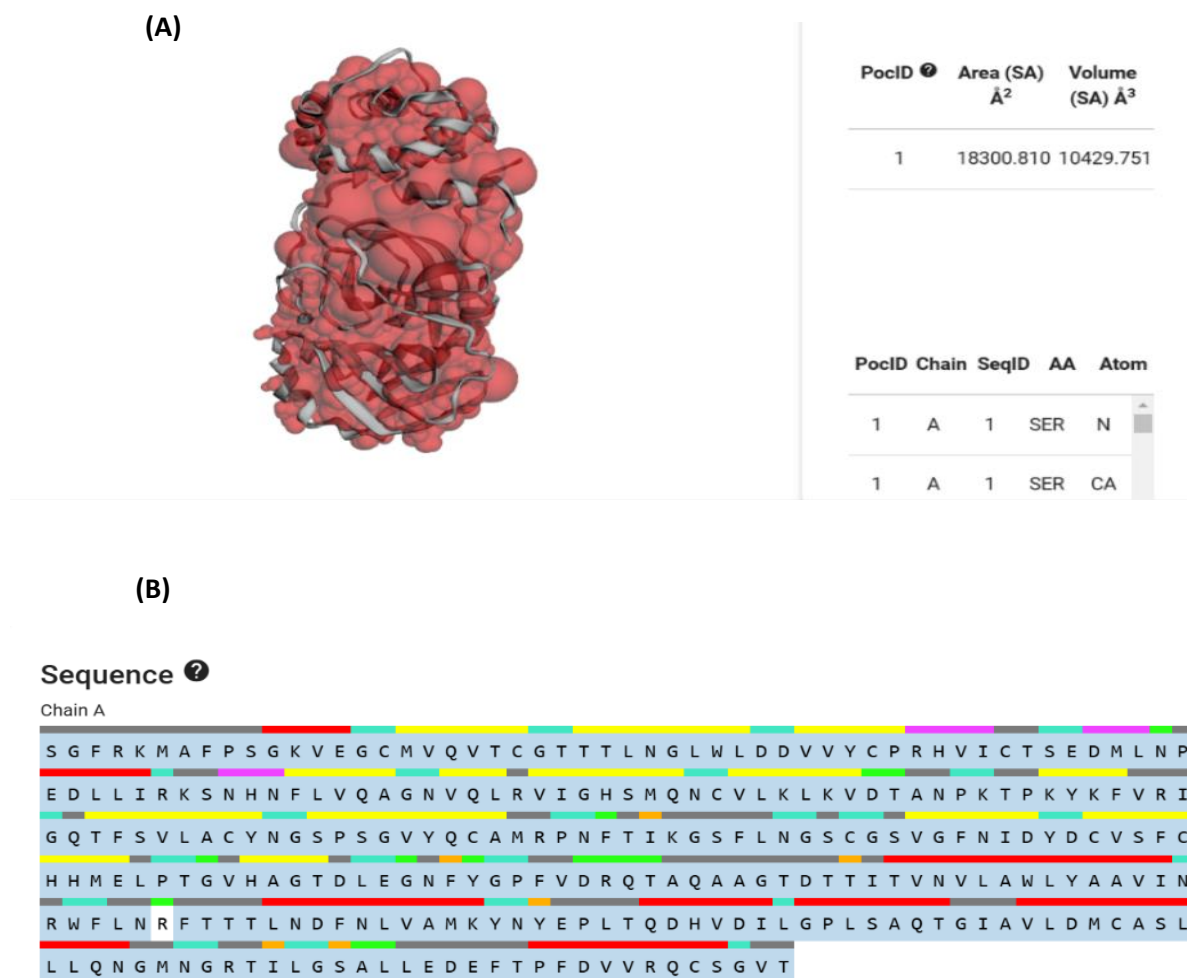
### ***3.1.7. Prediction of binding pocket analysis using CASTp Server:***

Protein structures are highly complex, comprising internal cavities, cross pathways, and surface pockets. These structural features serve as the foundation for various biological functions, including ligand binding, DNA interactions, and enzymatic activity. Understanding and quantifying these topographic characteristics is essential for deciphering the relationship between protein structure and function [79], engineering proteins with specific properties [80], and designing therapeutics targeting protein structures [81].

The Computed Atlas of Surface Topography of Proteins (CASTp) utilizes the alpha shape method [82], a concept from computational geometry, to identify and analyze protein topographic features. This approach allows for precise measurement of area, volume, and imprint calculations [83-89]. CASTp offers essential functionalities, such as detecting and characterizing protein structure channels, pockets, and cavities. Additionally, by incorporating pre-computed topographic features of biological assemblies from the Protein Data Bank (PDB) and providing negative volume imprints, the service has significantly enhanced its capabilities.

A refined user interface has improved the readability and accessibility of the server. The default probe radius for pre-computed results is set to **1.4 Å**, a standard for solvent-accessible surface area calculations. However, users can specify a custom probe radius for tailored analysis. CASTp precisely defines all atoms involved in forming protein structures, including surface pockets, internal cavities, and cross channels, while also computing their exact areas and volumes (**Figure 3.18**).

These calculations are performed analytically using both the solvent-accessible surface model (Richards' surface) [88] and the molecular surface model (Connolly's surface) [89]. The CASTp server also provides topographic feature imprints [103], which can be downloaded directly and visualized using UCSF Chimera.



**Figure 3.18.** The primary CASTp server user interface. (A)The panel in the pocket and (B) sequence panels

### 3.1.8. Predicting Physicochemical Properties, secondary structure, Intrinsically Disordered regions, and Mutation Effects on Protein Stability:

Comprehensive information on the physicochemical properties of a protein, secondary structure prediction, intrinsically disordered regions, and the impact of single-point mutations on stability and function of a given protein was predicted using online servers such as:

1. **I-Mutant 2.0:** This support vector machine-based tool predicts changes in protein stability resulting from single-point mutations [90].
2. **SIFT (Sorting Intolerant from Tolerant):** SIFT analyzes the effects of single-point mutations on protein function by evaluating their tolerance or intolerance scores [91].
3. **PONDR VLXT:** This tool estimates the percentage of intrinsic disorder in a protein. It integrates three predictors: two trained on X-ray characterized terminal

disordered regions and one trained on variously characterized long disordered regions [92].

4. **ExPASy ProtParam:** This online tool calculates physicochemical properties such as instability index and aliphatic index, providing insights into protein stability [93].
5. **GOR IV:** The GOR IV server is an advanced bioinformatics tool designed to predict protein secondary structures from amino acid sequences. Utilizing an information theory-based approach, it evaluates position-dependent scoring matrices to classify regions into alpha-helices, beta-sheets, or coils. By analyzing sequences provided in FASTA format, the server generates both graphical and tabular results, facilitating research in structural biology, protein function analysis, drug discovery, and protein engineering. As a freely accessible online resource, GOR IV is extensively applied in homology modeling and protein structure prediction, making it a valuable tool in computational biology [94].

### 3.1.9. 3-D structure visualization tools:

(i) **Visual Molecular Dynamics (VMD):** VMD is a computational tool designed for molecular modeling and visualization [95]. Its primary function is to facilitate the visualization and analysis of MD simulation results. Additionally, VMD supports various data types, including volumetric data, sequence information, and graphical objects, making it a versatile tool in computational biology.

(ii) **UCSF Chimera:** UCSF Chimera is a highly flexible software used for the interactive visualization and analysis of molecular structures along with related data such as conformational ensembles, density maps, sequence alignments, supra-molecular assemblies, and docking results [96]. Developed by the Resource for Biocomputing, Visualization, and Informatics (RBVI) with support from the National Institutes of Health (NIH), Chimera serves as a powerful tool in computational chemistry and structural biology.

(iii) **ArgusLab:** ArgusLab is a molecular modeling and drug design software specifically developed for Windows-based operating systems. Created by Mark Thompson, a research scientist at the Pacific Northwest National Laboratory, ArgusLab provides tools for

molecular graphics, docking studies, and computational chemistry applications, making it useful for rational drug design and molecular interaction studies [97].

### 3.1.10. 3-D structure modelling tools:

**1. I-TASSER web server:** Based on the Critical Assessment of Protein Structure Prediction (CASP) rankings, I-TASSER [98] has been recognized as the most effective automated server for predicting 3D protein structures. The server generates five distinct models, from which the most reliable structure is determined using the C-score calculated from the relative clustering structural density and consensus significance. To assess the accuracy of the selected model, Template Modeling (TM) score and Root Mean Square Deviation (RMSD) are utilized, providing insights into the structural quality and similarity to the native conformation.

### 3.1.11. Analysis of trajectories:

At this stage, the recorded coordinates and velocities of the system are utilized for further analysis, which requires MD trajectory files. When integrated with visualization tools like VMD, these simulations enable the time-dependent observation of structural parameters, facilitating a deeper understanding of atomic-level conformational changes. By means of the ptraj and cpptraj modules of AMBER14, parameters such as Root Mean Square Deviation (RMSD), Root Mean Square Fluctuation (RMSF), and Radius of Gyration (Rg) can be analyzed.

#### (i) Root Mean Square Deviation (RMSD):

RMSD is a measurement used to determine a structure's deviation from a certain conformation.

It is described as:

$$RMSD = \left[ \frac{\sum_N (R_i - R_i^0)^2}{N} \right]^{1/2} \dots\dots\dots (3.42)$$

where  $R_i$  is the vector location of particle  $i$  (the target atom) in the snapshot,  $R_i^0$  is the coordinate vector for reference atom  $i$ , and  $N$  is the total number of atoms/residues taken into account in the computation. Using backbone atoms and the simulation's first frame as a reference, the RMSD was calculated. The RMSD is the product of the number of locations (i), the number of strands (j), and the number of angular parameters (k). The value of  $N$  in the above equation denotes the total number of variables needed to compute the RMSD. A radial vector of length  $r$  in the structure

space denoted by the RMSD absolute magnitude is the calculated RMSD.

**(ii) 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):

$$RMSF = \left( \frac{1}{T} \sum_{t=1}^T (r_i(t) - r_i^{ref})^2 \right)^{1/2} \dots\dots\dots (3.43)$$

In **Equation 3.43**, The time period over which one intends to average is known as  $T$ ,  $r_i(t)$  is the Instantaneous Position of Atom  $i$  at Time  $t$  and  $r_i^{ref}$  as the reference position of particle  $i$ . The reference position will be the time-averaged position of the same particle  $i$ , i.e.  $r_i^{ref} = \bar{r}_i$ .

**Difference between RMSD and RMSF:** In MD simulations, the structural variations of biomolecules are commonly assessed using RMSD and RMSF. RMSD measures the overall deviation between two structures for a given set of atoms, while RMSF quantifies fluctuations around an average position for each atom, residue, or a collection of structures over a trajectory. It is possible to have an RMSD of zero with a non-zero RMSF for every atom, or a high RMSD with a low RMSF if a significant conformational change occurs, followed by minimal atomic fluctuations.

**(iii) Radius of Gyration (Rg):**

The radius of gyration is computed to assess the structure's compactness.:

$$R_g = \left( \frac{\sum_i |r_i|^2 m_i}{\sum_i m_i} \right)^{1/2} \dots\dots\dots (3.44)$$

In **Equation 3.44**,  $m_i$  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.



### 3.1.12. Bibliography

- [1]. Kar, R. K. Benefits of hybrid QM/mm over traditional classical mechanics in pharmaceutical systems. *Drug Discovery Today*, 28(1): 103374, 2023. <https://doi.org/10.1016/j.drudis.2022.103374>
- [2]. Alder, B. J., and Wainwright, T. E. Phase Transition for a Hard Sphere System. *The Journal of Chemical Physics*, 27(5):1208, 1957. <https://doi.org/10.1063/1.1743957>
- [3]. Alder, B. J., and Wainwright, T. E. Studies in Molecular Dynamics. I. General Method. *The Journal of Chemical Physics*, 31:459, 1959. <https://doi.org/10.1063/1.17303761959>
- [4]. Rahman, A. Correlations in the Motion of Atoms in Liquid Argon. *Physical Review*, 136(2A): A405, 1964. <https://doi.org/10.1103/PhysRev.136.A405>
- [5]. Stillinger, F.H., and Rahman, A. Improved Simulation of Liquid Water by Molecular Dynamics. *The Journal of Chemical Physics*, 60(4): 1545-1557, 1974. <https://doi.org/10.1063/1.1681229>
- [6]. McCammon, J.A., Gelin, B.R., and Karplus, M. Dynamics of folded proteins. *Nature*, 267(5612): 585-590, 1977. <https://doi.org/10.1038/267585a0>
- [7]. Andrew, R. L. Molecular modeling principles and applications. Dorling Kindersley (India) Pvt. Ltd., U.P. India, 2nd edition, 2001.
- [8]. Hummer, G. The numerical accuracy of truncated Ewald sums for periodic systems with long-range Coulomb interactions. *Chemical Physics Letters*, 235: 297, 1995. [https://doi.org/10.1016/0009-2614\(95\)00117-M](https://doi.org/10.1016/0009-2614(95)00117-M)
- [9]. Ryckaert, J. P., Ciccotti, G., and Berendsen, H. J. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n- alkanes. *Journal of Computational Physics*, 23(3): 327-341, 1977. [https://doi.org/10.1016/0021-9991\(77\)90098-5](https://doi.org/10.1016/0021-9991(77)90098-5)
- [10]. Katiyar, R. S., and Jha, P. K. Molecular simulations in drug delivery: Opportunities and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 8(4): e1358, 2018. <https://doi.org/10.1002/wcms.1358>
- [11]. Anderson, H. C. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4): 2384, 1980. <https://doi.org/10.1063/1.439486>
- [12]. Case, D.A., Ben-Shalom, I.Y., S.R. Brozell, D.S. Cerutti, T.E. Cheatham, III, V.W.D. Cruzeiro, T.A. Darden, R.E. Duke, D. Ghoreishi, M.K. Gilson, H. Gohlke, A.W. Goetz, D. Greene, R Harris, N. Homeyer, Y. Huang, S. Izadi, A. Kovalenko, T. Kurtzman, T.S. Lee,

S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D.J. Mermelstein, K.M. Merz, Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D.R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C.L. Simmerling, J. Smith, R. SalomonFerrer, J. Swails, R.C. Walker, J. Wang, H. Wei, R.M. Wolf, X. Wu, L. Xiao, D.M. York and P.A. Kollman. AMBER 2018, University of California, San Francisco, 2018.

[13]. Darden, T., York, D., and Pedersen, L. Particle mesh Ewald: An  $N \cdot \log(N)$  method for Ewald sums in large systems. *The Journal of chemical physics*, 98(12):10089–10092, 1993.

<https://doi.org/10.1063/1.464397>

[14]. Jorgensen, W. L., and Jenson, C. Temperature dependence of TIP3P, SPC, and TIP4P water from NPT Monte Carlo simulations: Seeking temperatures of maximum density. *Journal of Computational Chemistry*, 19(10): 1179–1186, 1998.

[https://doi.org/10.1002/\(SICI\)1096-987X\(19980730\)19:10%3C1179::AID-JCC6%3E3.0.CO;2-J](https://doi.org/10.1002/(SICI)1096-987X(19980730)19:10%3C1179::AID-JCC6%3E3.0.CO;2-J)

[15]. Hagler, A. T. Theoretical simulation of conformation, Energetics, and dynamics of peptides. *Conformation in Biology and Drug Design*, 213–299, 1985.

<https://doi.org/10.1016/b978-0-12-304207-1.50011-4>

[16]. Struthers, R. S., Hagler, A. T., and Rivier, J. Design of peptide analogs. *ACS Symposium Series*, 239–261, 1984. <https://doi.org/10.1021/bk-1984-0251.ch01>

[17]. Seifoori, S., Ebrahimi, F., Parrany, A. M., and Liaghat, G. H. Dynamic analysis of single-layered graphene sheet subjected to a moving nanoparticle: A molecular dynamics study. *Materials Science and Engineering: B*, 285:115956, 2022.

<https://doi.org/10.1016/j.mseb.2022.115956>

[18]. Yabe, H., and Sakaiwa, N. A new nonlinear conjugate gradient method for unconstrained optimization. *Journal of the Operations Research Society of Japan*, 48(4): 284–296, 2005.

<https://doi.org/10.15807/jorsj.48.284>

[19]. Nocedal, J., and Wright, S. J. Numerical optimization. *Springer Series in Operations Research and Financial Engineering*, 1999. <https://doi.org/10.1007/b98874>

[20]. Kirkwood, J. G. Statistical mechanics of fluid mixtures. *The Journal of Chemical Physics*, 3(5): 300–313, 1935. <https://doi.org/10.1063/1.1749657>

[21]. Roux, B. The calculation of the potential of mean force using computer simulations. *Computer Physics Communications*, 91(1): 275–282, 1995. [https://doi.org/10.1016/0010-4655\(95\)00053-I](https://doi.org/10.1016/0010-4655(95)00053-I).

[22]. Torrie, G. M., and Valleau, J. P. Nonphysical sampling distributions in Monte Carlo

free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2): 187-199, 1977. [https://doi.org/10.1016/0021-9991\(77\)90121-8](https://doi.org/10.1016/0021-9991(77)90121-8)

[23]. Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *Journal of computational chemistry*, 13(8): 1011-1021, 1992. <https://doi.org/10.1002/jcc.540130812>

[24]. Massova, I. and Kollman, P. A. Computational alanine scanning to probe protein protein interactions: a novel approach to evaluate binding free energies. *Journal of the American Chemical Society*, 121(36):8133-8143, 1999. <https://doi.org/10.1021/ja990935j>

[25]. Camacho, C. J., Gatchell, D. W., Kimura, S. R., and Vajda, S. Scoring docked conformations generated by rigid-body protein-protein docking. *Proteins*, 40(3): 525-537, 2000. [https://doi.org/10.1002/1097-0134\(20000815\)40:3<525::aid-prot190>3.0.co;2-f](https://doi.org/10.1002/1097-0134(20000815)40:3<525::aid-prot190>3.0.co;2-f)

[26]. Goldman, B. B., and Wipke, W. T. QSD quadratic shape descriptors. 2. Molecular docking using quadratic shape descriptors (QSDock). *Proteins*, 38(1): 79-94, 2000. [http://dx.doi.org/10.1002/\(SICI\)1097-0134\(20000101\)38:13.0.CO;2-U](http://dx.doi.org/10.1002/(SICI)1097-0134(20000101)38:13.0.CO;2-U)

[27]. Gardiner, E. J., Willett, P., and Artymiuk, P. J. Protein docking using a genetic algorithm. *Proteins*, 44(1): 44-56, 2001. <https://doi.org/10.1002/prot.1070>

[28]. Chen, R., and Weng, Z. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins*, 47(3): 281-294, 2002. <https://doi.org/10.1002/prot.10092>

[29]. Gray, J. J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C. A., and Baker, D. Protein-protein docking with simultaneous optimization of rigid body displacement and side-chain conformations. *Journal of molecular biology*, 331(1): 281-299, 2003. [https://doi.org/10.1016/s0022-2836\(03\)00670-3](https://doi.org/10.1016/s0022-2836(03)00670-3)

[30]. Jain, A. N. Scoring functions for protein-ligand docking. *Current Protein and Peptide Science*, 7(5): 407-420, 2006. <https://doi.org/10.2174/138920306778559395>

[31]. Duhovny, D., Nussinov, R., and Wolfson, H. J. Efficient unbound docking of rigid molecules. In *Algorithms in Bioinformatics: Second International Workshop, WABI 2002 Rome, Italy*, Springer, Berlin, Heidelberg, 185-200, 2002. [https://doi.org/10.1007/3-540-45784-4\\_14](https://doi.org/10.1007/3-540-45784-4_14)

[32]. de Vries, S., van Dijk, M. & Bonvin, A. M. The HADDOCK web server for data-driven biomolecular docking. *Nature Protocols*, 5(5): 883-897, 2010. <https://doi.org/10.1038/nprot.2010.32>

[33]. van Dijk, M., van Dijk, A. D., Hsu, V., Boelens, R., and Bonvin, A. M. Information-driven protein-DNA docking using HADDOCK: it is a matter of flexibility. *Nucleic Acids*

*Research*, 34(11): 3317-3325, 2006. <https://doi.org/10.1093/nar/gkl412>

[34]. Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., and Vajda, S. The ClusPro web server for protein–protein docking. *Nature protocols*, 12(2): 255–278, 2017. <https://doi.org/10.1038/nprot.2016.169>

[35]. Sato, R., Harada, R. and Shigeta, Y. The binding structure and affinity of photodamaged duplex DNA with members of the photolyase/cryptochrome family: A computational study. *Biophysics and physicobiology*, 15:18-27, 2018. [https://doi.org/10.2142/biophysico.15.0\\_18](https://doi.org/10.2142/biophysico.15.0_18)

[36]. Comeau, S. R., Gatchell, D. W., Vajda, S., and Camacho, C. J. ClusPro: a fully automated algorithm for protein-protein docking. *Nucleic Acids Research*, 32: W96–99, 2004. <https://doi.org/10.1093/nar/gkh354>

[37]. Comeau, S. R., Gatchell, D. W., Vajda, S., and Camacho, C. J. ClusPro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics*, 20(1): 45–50, 2004. <https://doi.org/10.1093/bioinformatics/btg371>

[38]. Comeau, S. R., Kozakov, D., Brenke, R., Shen, Y., Beglov, D., and Vajda, S. ClusPro: performance in CAPRI rounds 6–11 and the new server. *Proteins*, 69(4): 781–785, 2007. <https://doi.org/10.1002/prot.21795>

[39]. Kozakov, D., Hall, D. R., Beglov, D., Brenke, R., Comeau, S. R., Shen, Y., Li, K., Zheng, J., Vakili, P., Paschalidis, I. C. h., and Vajda, S. Achieving reliability and high accuracy in automated protein docking: ClusPro, PIPER, SDU, and stability analysis in CAPRI rounds 13–19. *Proteins*, 78(15): 3124–3130, 2010. <https://doi.org/10.1002/prot.22835>

[40]. Kozakov, D., Beglov, D., Bohnuud, T., Mottarella, S. E., Xia, B., Hall, D. R., and Vajda, S. How good is automated protein docking? *Proteins*, 81(12): 2159–2166, 2013. <https://doi.org/10.1002/prot.24403>

[41]. Kozakov, D., Brenke, R., Comeau, S.R. and Vajda, S. PIPER: an FFT based protein docking program with pairwise potentials. *Proteins*, 65(2): 392–406, 2006. <https://doi.org/10.1002/prot.21117>

[42]. Kollman, P. A., Massova, I., Reyes, C., Kuhn, B., Huo, S., Chong, L., Lee, M., Lee, T., Duan, Y., Wang, W., Donini, O., Cieplak, P., Srinivasan, J., Case, D. A., and Cheatham, T. E., 3<sup>rd</sup>. Calculating structures and free energies of complex molecules: combining molecular

mechanics and continuum models. *Accounts of chemical research*, 33(12): 889-897, 2000.

<https://doi.org/10.1021/ar000033j>

[43]. Hou, T., Wang, J., Li, Y., and Wang, W. Assessing the performance of the MM/PBSA and MM/GBSA methods. 1. The accuracy of binding free energy calculations based on molecular dynamics simulations. *Journal of chemical information and modeling*, 51(1): 69-82, 2010. <https://doi.org/10.1021/ci100275a>

[44]. Wang, J., Hou, T. and Xu, X. Recent advances in free energy calculations with a combination of molecular mechanics and continuum models. *Current Computer Aided Drug Design*, 2(3): 287-306, 2006. <https://doi.org/10.2174/157340906778226454>

[45]. Sato, R., Harada, R. and Shigeta, Y. The binding structure and affinity of photodamaged duplex DNA with members of the photolyase/cryptochrome family: A computational study. *Biophysics and physicobiology*, 15: 18-27, 2018. [https://doi.org/10.2142/biophysico.15.0\\_18](https://doi.org/10.2142/biophysico.15.0_18)

[46]. Metz, A., Pflieger, C., Kopitz, H., Pfeiffer-Marek, S., Baringhaus, K. H., and Gohlke, H. Hot spots and transient pockets: predicting the determinants of small molecule binding to a protein-protein interface. *Journal of chemical information and modeling*, 52(1): 120-133, 2011. <https://doi.org/10.1021/ci200322s>

[47]. Gohlke, H., Kiel, C., and Case, D. A. Insights into protein-protein binding by binding free energy calculation and free energy decomposition for the Ras-Raf and Ras-RalGDS complexes. *Journal of molecular biology*, 330(4): 891-913, 2003. [https://doi.org/10.1016/s0022-2836\(03\)00610-7](https://doi.org/10.1016/s0022-2836(03)00610-7)

[48]. Miller III, B. R., McGee Jr, T. D., Swails, J. M., Homeyer, N., Gohlke, H., and Roitberg, A. E. MMPBSA.py: an efficient program for end-state free energy calculations. *Journal of chemical theory and computation*, 8(9): 3314-3321, 2012. <https://doi.org/10.1021/ct300418h>

[49]. Ahmad, R., Brandsdal, B. O., Michaud-Soret, I., and Willassen, N. P. Ferric uptake regulator protein: Binding free energy calculations and per-residue free energy decomposition. *Proteins*, 75(2): 373-386, 2009. <https://doi.org/10.1002/prot.22247>

[50]. Chen, J., Zhang, S., Liu, X., and Zhang, Q. Insights into drug resistance of mutations D30N and I50V to HIV-1 protease inhibitor TMC-114: free energy calculation and molecular dynamic simulation. *Journal of molecular modeling*, 16(3): 459-468, 2010. <https://doi.org/10.1007/s00894-009-0553-7>

- [51]. Khosa, S., Frieg, B., Mulnaes, D., Kleinschrodt, D., Hoepfner, A., Gohlke, H., and Smits, S. H. Structural basis of lantibiotic recognition by the nisin resistance protein from *Streptococcus agalactiae*. *Scientific reports*, 6(1): 18679–18692, 2016. <https://doi.org/10.1038/srep18679>
- [52]. Cele, F. N., Ramesh, M., and Soliman, M. E. Per-residue energy decomposition pharmacophore model to enhance virtual screening in drug discovery: a study for identification of reverse transcriptase inhibitors as potential anti-HIV agents. *Drug design, development and therapy*, 10: 1365–1377, 2016. <https://doi.org/10.2147/DDDT.S95533>
- [53]. Xue, L. C., Rodrigues, J. P., Kastiris, P. L., Bonvin, A. M., and Vangone, A. PRODIGY: a web server for predicting the binding affinity of protein–protein complexes. *Bioinformatics*, 32(23): 3676–3678, 2016. <https://doi.org/10.1093/bioinformatics/btw514>
- [54]. Weng, G., Wang, E., Wang, Z., Liu, H., Zhu, F., Li, D., and Hou, T. HawkDock: a web server to predict and analyze the protein-protein complex based on computational docking and MM/GBSA. *Nucleic Acids Research*, 47(W1): W322–W330, 2019. <https://doi.org/10.1093/nar/gkz397>.
- [55]. Cohen, F. E., and Prusiner, S. B. Pathologic conformations of prion proteins. *Annual Review of Biochemistry*, 67(1): 793–819, 1998. <https://doi.org/10.1146/annurev.biochem.67.1.793>
- [56]. Selkoe, D. J. The cell biology of  $\beta$ -amyloid precursor protein and presenilin in Alzheimer's disease. *Trends in cell biology*, 8(11): 447–453, 1998. [https://doi.org/10.1016/s0962-8924\(98\)01363-4](https://doi.org/10.1016/s0962-8924(98)01363-4)
- [57]. Loregian, A., Marsden, H. S., and Palu, G. Protein–protein interactions as targets for antiviral chemotherapy. *Reviews in medical virology*, 12(4): 239–262, 2002. <https://doi.org/10.1002/rmv.356>
- [58]. Keskin, O., Gursoy, A., Ma, B., and Nussinov, R. Principles of protein–protein interactions: What are the preferred ways for proteins to interact? *Chemical Reviews*, 108(4): 1225–1244, 2008. <https://doi.org/10.1021/cr040409x>
- [59]. Janin, J. Protein–protein recognition. *Progress in Biophysics and Molecular Biology*, 64 (2–3): 145–166, 1995. [https://doi.org/10.1016/S0079-6107\(96\)00001-6](https://doi.org/10.1016/S0079-6107(96)00001-6)
- [60]. Jones, S., and Thornton, J. M. Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences*, 93(1): 13–20, 1996. <https://doi.org/10.1073/pnas.93.1.13>



- [61]. Janin, J., and Chothia, C. The structure of protein-protein recognition sites. *The Journal of Biological Chemistry*, 265(27): 16027–16030, 1990. [https://doi.org/10.1016/s0021-9258\(17\)46181-3](https://doi.org/10.1016/s0021-9258(17)46181-3)
- [62]. Archakov, A. I., Govorun, V. M., Dubanov, A. V., Ivanov, Y. D., Veselovsky, A. V., Lewi, P., and Janssen, P. Protein-protein interactions as a target for drugs in proteomics. *Proteomics*, 3(4): 380–391, 2003. <https://doi.org/10.1002/pmic.200390053>
- [63]. Laskowski, R. A., Jablonska, J., Pravda, L., Vařeková, R. S. and Thornton, J.M. PDBsum: Structural summaries of PDB entries. *Protein Science: a publication of the Protein Society*, 27(1): 129-134, 2018. <https://doi.org/10.1002/pro.3289>
- [64]. Laskowski, R. A. PDBsum new things. *Nucleic Acids Research*, 37: D355– D359, 2009. <https://doi.org/10.1093/nar/gkn860>
- [65]. Rose, P. W., Prlić, A., Bi, C., Bluhm, W. F., Christie, C. H., Dutta, S., Green, R. K., Goodsell, D. S., Westbrook, J. D., Woo, J., Young, J., Zardecki, C., Berman, H. M., Bourne, P. E., and Burley, S. K. The RCSB Protein Data Bank: Views of Structural Biology for Basic and Applied Research and Education. *Nucleic Acids Research*, 43: D345-D356, 2014. <https://doi.org/10.1093/nar/gku1214>
- [66]. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shinyalov, I. N., and Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research*, 28(1): 235–242, 2000. <https://doi.org/10.1093/nar/28.1.235>
- [67]. Laskowski, R. A. and Swindells, M. B. LigPlot+: multiple ligand–protein interaction diagrams for drug discovery. *Journal of Chemical Information and Modeling*, 51 (10): 2778–2786, 2011. <https://doi.org/10.1021/ci200227u>
- [68]. Emerson, I. A., and Amala, A. Protein contact maps: a binary depiction of protein 3D structures. *Physica. A: Statistical Mechanics and its Application*, 465:782–791, 2017. <https://doi.org/10.1016/j.physa.2016.08.033>
- [69]. Kozma, D., Simon, I., and Tusnady, G. E. CMWeb: an interactive on-line tool for analysing residue-residue contacts and contact prediction methods. *Nucleic Acids Research*, 40: W329–W333, 2012. <https://doi.org/10.1093/nar/gks488>
- [70]. Sánchez Rodríguez, F., Mesdaghi, S., Simpkin, A. J., Burgos- Mármol, J. J., Murphy, D. L., Uski, V., Keegan, R. M., and Rigden, D. J. ConPlot: web-based application for the visualization of protein contact maps integrated with other data. *Bioinformatics*, 37(17):2763–2765, 2021. <https://doi.org/10.1093/bioinformatics/btab049>

- [71]. Konopka, B. M., Ciombor, M., Kurczynska, M., and Kotulska, M. Automated procedure for contact-map-based protein structure reconstruction. *Journal of Membrane Biology*, 247(5):409–420, 2014. <https://doi.org/10.1007/s00232-014-9648-x>
- [72]. Vassura, M., Margara, L., Di Lena, P., Medri, F., Fariselli, P., and Casadio, R. FT-COMAR: fault tolerant three-dimensional structure reconstruction from protein contact maps. *Bioinformatics*, 24(10):1313–1315, 2008. <https://doi.org/10.1093/bioinformatics/btn115>
- [73]. Vehlow, C., Stehr, H., Winkelmann, M., Duarte, J. M., Petzold, L., Dinse, J., and Lappe, M. CMView: interactive contact map visualization and analysis. *Bioinformatics*, 27(11):1573–1574, 2011. <https://doi.org/10.1093/bioinformatics/btr163>
- [74]. Badaczewska-Dawid, A.E., Kolinski, A., and Kmiecik, S. Computational reconstruction of atomistic protein structures from coarse-grained models. *Computational Structural Biotechnology Journal*, 18:162–176, 2020. <https://doi.org/10.1016/j.csbj.2019.12.007>
- [75]. Ribeiro, J., Ríos- Vera, C., Melo, F., and Schüller, A. Calculation of accurate interatomic contact surface areas for the quantitative analysis of non-bonded molecular interactions. *Bioinformatics*, 35(18):3499–3501, 2019. <https://doi.org/10.1093/bioinformatics/btz062>
- [76]. Ribeiro, J., Melo, F., and Schüller, A. PDIViz: analysis and visualization of protein–DNA binding interfaces. *Bioinformatics*, 31(16): 2751–2753, 2015. <https://doi.org/10.1093/bioinformatics/btv203>
- [77]. Badaczewska-Dawid, A. E., Nithin, C., Wroblewski, K., Kurcinski, M., and Kmiecik, S. MAPIYA contact map server for identification and visualization of molecular interactions in proteins and biological complexes, *Nucleic Acids Research*, 50(W1): W474–W482, 2022. <https://doi.org/10.1093/nar/gkac307>
- [78]. Eastman, P., Swails, J., Chodera, J. D., McGibbon, R.T., Zhao, Y., Beauchamp, K. A., Wang, L. P., Simonett, A.C., Harrigan, M.P., Stern, C. D., Wiewiora, R. P., Brooks, B. R., and Pande, V. S. OpenMM 7: rapid development of high-performance algorithms for molecular dynamics. *PLoS Computational Biology*, 13(7): e1005659, 2017. <https://doi.org/10.1371/journal.pcbi.1005659>



- [79]. Forli, S., Huey, R., Pique, M. E., Sanner, M. F., Goodsell, D. S., and Olson, A. J. Computational protein–ligand docking and virtual drug screening with the autodock suite. *Nature Protocols*, 11(5): 905–919, 2016. <https://doi.org/10.1038/nprot.2016.051>
- [80]. Toh, S., Holbrook-Smith, D., Stogios, P. J., Onopriyenko, O., Lumba, S., Tsuchiya, Y., Savchenko, A. and McCourt, P. Structure-function analysis identifies highly sensitive strigolactone receptors, *Striga. Science*, 350(6257): 203–207, 2015. <https://doi.org/10.1126/science.aac9476>
- [81]. Reetz, M.T. Directed evolution of promiscuity: Artificial enzymes as catalysts in organic chemistry. *Directed Evolution of Selective Enzymes*, 237–266 2016. <https://doi.org/10.1002/9783527655465.ch7>
- [82]. Brouwer, J. M., Lan, P., Cowan, A. D., Bernardini, J. P., Birkinshaw, R. W., van Delft, M. F., Sleebs, B. E., Robin, A. Y., Wardak, A., Tan, I. K., Reljic, B., Lee, E. F., Fairlie, W. D., Call, M. J., Smith, B. J., Dewson, G., Lessene, G., Colman, P. M., and Czabotar, P. E. Conversion of BIM-BH3 from activator to inhibitor of Bak through structure-based design. *Molecular Cell*, 68(4): 659–672, 2017. <https://doi.org/10.1016/j.molcel.2017.11.001>
- [83]. Edelsbrunner, H. and Mucke, E.P. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1): 43–72, 1994. <https://doi.org/10.1145/174462.156635>
- [84]. Ebalunode, J.O., Ouyang, Z., Liang, J. and Zheng, W. Novel approach to structure-based pharmacophore search using computational geometry and shape matching techniques. *Journal of Chemical Information and Modeling*, 48(4): 889–901, 2008. <https://doi.org/10.1021/ci700368p>
- [85]. Tian, W., and Liang, J. On quantification of geometry and topology of protein pockets and channels for assessing mutation effects. *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2018: 263–266, 2018. <https://doi.org/10.1109/bhi.2018.8333419>
- [86]. Liang, J., Edelsbrunner, H., and Woodward, C. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein Science*, 7(9): 1884–1897, 1998. <https://doi.org/10.1002/pro.5560070905>

- [87]. Edelsbrunner, H., Facello, M., and Liang, J. On the definition and the construction of pockets in macromolecules. *Discrete Applied Mathematics*, 88(1-3): 83–102, 1998. [https://doi.org/10.1016/S0166-218X\(98\)00067-5](https://doi.org/10.1016/S0166-218X(98)00067-5)
- [88]. Lee, B. and Richards, F. M. The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55(3): 379–400, 1971. [https://doi.org/10.1016/0022-2836\(71\)90324-x](https://doi.org/10.1016/0022-2836(71)90324-x)
- [89]. Connolly, M. Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221(4612): 709–713, 1983. <https://doi.org/10.1126/science.6879170>
- [90]. Capriotti, E., Fariselli, P., and Casadio, R. I-mutant2.0: Predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Research*, 33: W306–10, 2005. <http://doi.org/10.1093/nar/gki375>
- [91]. Ng, P. C., and Henikoff, S. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Research*, 31(13): 3812–3814, 2003. <http://doi.org/10.1093/nar/gkg509>
- [92]. Xue, B., Dunbrack, R. L., Williams, R. W., Dunker, A. K., and Uversky, V. N. Ponder fit: A meta-predictor of intrinsically disordered amino acids. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, 1804(4): 996–1010, 2010. <http://doi.org/10.1016/j.bbapap.2010.01.011>
- [93]. Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R. D., and Bairoch, A. ExPASy: The Proteomics Server for in-depth protein knowledge and analysis. *Nucleic Acids Research*, 31(13): 3784–3788, 2003. <http://doi.org/10.1093/nar/gkg563>
- [94]. Garnier, J., Gibrat, J. F., and Robson, B. GOR method for predicting protein secondary structure from amino acid sequence. *Methods in Enzymology*, 266: 540–553, 1996. [https://doi.org/10.1016/S0076-6879\(96\)66034-0](https://doi.org/10.1016/S0076-6879(96)66034-0)
- [95]. Humphrey, W., Dalke, A. and Schulten, K. VMD: visual molecular dynamics. *Journal of molecular graphics*, 14(1): 33–38, 1996. [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5)
- [96]. Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., and Ferrin, T. E. UCSF Chimera—a visualization system for exploratory research and

analysis. *Journal of computational chemistry*, 25(13): 1605-1612, 2004.  
<https://doi.org/10.1002/jcc.20084>

[97]. Bitencourt-Ferreira, G. and de Azevedo, W.F. Molecular Docking Simulations with ArgusLab. *Methods in Molecular Biology*, 2053(2019): 203–220, 2019.  
[http://dx.doi.org/10.1007/978-1-4939-9752-7\\_13](http://dx.doi.org/10.1007/978-1-4939-9752-7_13).

[98]. Roy, A., Kucukural, A., and Zhang, Y. I-TASSER: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4): 725-738, 2010.  
<https://doi.org/10.1038/nprot.2010.5>