

Molecular Dynamics Simulation Methods to Study Structural Dynamics of Proteins



Anil Kumar and Krishna Kumar Ojha

Abstract Molecular dynamics (MD) simulation is a computational technique for understanding the physical motions of atomic and molecular particles. In this approach, atoms and molecules interact for a defined time period, revealing information on the dynamic evolution of the system. Newton's equations of motion are used to determine the trajectories of atoms and molecules. The forces and potential energy between atoms and molecules are calculated using molecular mechanics force fields or interatomic potentials. The approach was originally created for applications in the field of theoretical physics; however, it is now used in other areas, including materials science, theoretical chemistry, computational biology, etc. This technique determines the time-dependent behaviour of a molecular system. MD simulation has been widely used to study the conformational changes of biomacromolecules to explore the structure and dynamics of proteins, nucleic acids, and their complexes. It has also been used to study the protein–ligand interactions, which are essential for various processes inside the cell, such as signal transduction, immune reaction, and gene regulation. The data help explore the regulatory mechanisms of various biological processes. MD studies also provide a theoretical background for drug design and discovery. Therefore, MD simulation has been extensively used by researchers in combination with biochemical and biophysical methods to obtain a dynamic understanding of biomolecular behaviour. This chapter discusses various MD simulation methods and how they are used to study the structural dynamics of proteins.

Keywords MD simulation · Force fields · Statistical mechanics · Classical mechanics · Periodic boundary condition · Energy minimization · Metadynamics · Umbrella sampling · Protein folding

A. Kumar (✉) · K. K. Ojha
Department of Bioinformatics, Central University of South Bihar, Gaya, India
e-mail: kumaranil@cub.ac.in

1 Introduction

In living beings, proteins carry out various cellular functions, including transport, cell signalling, and metabolic processes, such as catalysis. All of these processes rely heavily on the structural dynamics of the protein. The protein sequence folds in a specific manner to get a 3D conformation stabilized by various chemical interactions, including covalent and non-covalent interactions. The general approach to understanding the folding process of a protein is studying its unfolding behaviour. Spectroscopic techniques, such as circular dichroism (CD) and fluorescence spectroscopy, are most commonly used for understanding the forces and interactions involved in protein unfolding dynamics [1]. Understanding the protein folding/unfolding processes necessitates detailed atomic-level data, which could not be obtained using conventional wet-lab spectroscopic techniques. Recent years have seen the development of molecular dynamics (MD) simulation as a tool for understanding protein dynamics at the atomic level [2]. This method provides information about each atom as a function of time to characterize a molecule's dynamic behaviour. MD simulation has the merit of delivering time-dependent information regarding the folding and unfolding processes and inter-residue interactions [3].

In the late 1950s, Alder and Wainwright employed the MD method to investigate the interactions of hard spheres for the first time [4, 5]. Their discoveries shed light on the behaviour of simple liquids in various ways. Rahman made the following significant achievement in 1964 when he simulated liquid argon for the first time using a realistic potential [6]. Rahman and Stillinger's simulation of liquid water in 1971 was the first MD simulation study of a realistic system [7]. In 1977, the first protein simulation was performed [8]. MD simulation of solvated proteins, protein–DNA complexes, and lipids are very common in today's published reports, dealing with various challenges such as ligand binding thermodynamics and protein folding [9]. The number of simulation methodologies has exploded, and there is now a plethora of techniques for specific problems, such as mixed quantum-classical simulations for studying enzyme activities in the context of the entire protein. MD simulation approaches are also extensively utilized in experimental methods such as X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy to provide dynamic information about the proteins [10].

The field of MD simulation is rapidly expanding. The improvement of numerous approaches, such as force field advancement, sampling techniques, and superior processing power, has enabled us to do simulations in the microsecond to millisecond range with femtosecond coordinates [11]. MD simulation has the potential to shed light on a variety of biological problems. The use of MD simulation, on the other hand, necessitates the development of optimum models that closely resemble the cellular environment. As a result, the MD simulation will be more effective if more robust algorithms for modelling, docking, scoring, and energy calculations are developed [12].

In the last few decades, various approaches, including X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy (cryo-EM), have been used to

Table 1 RCSB PDB statistics as of December 25, 2022

RCSB PDB total entries (199,507)					
S. no.	Methods	Entries	S. no.	Types of molecules	Entries
1	X-ray	171,640	1	Protein only	173,443
2	Nuclear magnetic resonance (NMR)	13,878	2	Protein -NA	11,157
3	Electron microscopy (EM)	13,665	3	DNA only	2247
4	Multi-method	213	4	RNA only	1681
5	Others	111	5	Others	10,979

produce structures of a large number of biomacromolecules. However, there is still a significant disparity between the number of available protein sequences and the available protein structures. UniProtKB/TrEMBL's recent release comprises 22,95,80,745 sequence entries, whereas the protein data bank (PDB) only has 1,99,507 structures as of December 25, 2022. This shows that only a tiny part of all sequences have known structures. The PDB data as of December 25, 2022, are displayed in Table 1. As a result, protein structure prediction is critical for closing this huge gap. The recent development of high-end computing, such as DeepMind's AlphaFold, has been used to create models for half of the understudied (dark) human proteins [13]. It has also determined around 200 million protein structures from almost 1 million species, now available to scientists in DeepMind's database [14].

Though biomolecules are highly dynamic, most of the above approaches provide the structural information of a biomolecule in a static manner [15], such as a protein–small molecule complex presented as a static pose through molecular docking. MD simulation can aid large-scale computing to predict dynamic behaviour [5, 16]. Protein conformational dynamics serve a variety of functions, including transport, signalling molecules, sensors, and mechanical effectors, as well as interacting with the various substrates [17, 18].

In MD simulation, the protein and water molecules are used to create and mimic *in vivo* environments. Protein and water atoms move in femtosecond (fs) time scale. The forces on each atom are calculated using a force field. The force field includes bonded and non-bonded potential terms in potential energy functions. Newton's law of motion is used to update the velocity and coordinates of the systems, which are also updated in the trajectory with time. The first MD simulation of biological macromolecules was performed by McCammon et al. in 1977 for the bovine pancreatic trypsin inhibitor [8]. Later, researchers explored the role of thermal factor (β) in the internal movements of protein [19–21]. Aspects of mean square variations versus residue number were investigated in these studies. Subsequent advancements in MD simulation revealed a broad spectrum of nucleic acid and protein motions. Since the trajectories can store all the coordinates, they can provide an ensemble of conformations of any structure. From MD simulation data, principal component analysis (PCA) can also be performed [22, 23]. MD simulation provides information on macromolecular structural flexibility and abet in comprehending experimental

results, such as NMR parameter dynamics and the effect of solvent and temperature on the stability of a protein [24, 25]. For X-ray structure refinement and NMR structure determination, the simulated annealing method is commonly used [26].

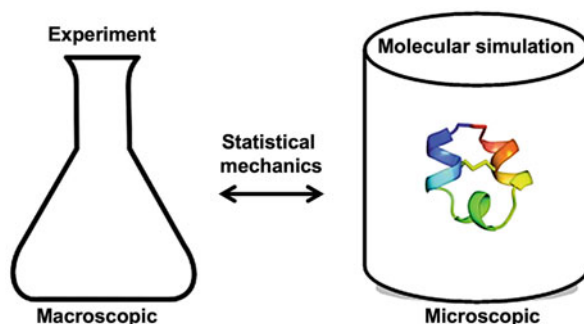
MD simulation can be used for computing the temporal evolution of atomic degrees of freedom by solving Newtonian equations of motion [27]. It allows researchers to observe atomic processes, such as chemical reactions and atomic diffusion, at the atomic time and length scales in large or complex systems. Analysis of repeated simulations, each run under various conditions, allows for the development of a model for a dynamic process [28]. It is among the essential tools for understanding biomolecules theoretically. This approach determines the time-dependent behaviour of a system. It provides precise information on the conformational and structural changes of proteins and nucleic acids [29]. Biomolecular processes occur over a wide range of time scales: side chain and loop motions are classified as local motions (0.01 to 5 Å range) that take 10^{-15} to 10^{-1} s to complete; rigid body motions (1 to 10 Å range) include helix, domain, and subunit motions that typically take 10^{-9} to 1 s; and helix-coil transitions and protein folding are examples of large-scale motions (>5 Å range) that take 10^{-7} to 10^4 s. Changes that occur in a short period are difficult to view using macroscopic experiments, but by simulating under physiological conditions computationally, the majority of the changes that happen in a short period of time can be visualized. MD simulation enables the investigation of complex biological systems, such as protein stability, protein folding, molecular recognition, ion transport, etc. It also allows researchers to investigate computer-aided drug design using structural information of biomolecules obtained through X-ray and NMR.

2 Statistical Mechanics

MD simulation generates microscopic data such as atomic positions and velocities. Using statistical mechanics, this data can be converted into macroscopic observables such as pressure, energy, and heat capacity [30]. Statistical mechanics is essential for MD simulation of biological systems. MD simulation is commonly used to investigate a system's macroscopic properties using microscopic simulations, such as the calculation of changes in the binding free energy of a candidate drug or to investigate the energetics and processes of conformational changes [31]. The mathematical formulae that correlate macroscopic properties with the motion of the atoms and molecules are provided by statistical mechanics. MD simulation, on the other hand, provides methods for solving particle equations of motion and evaluating these formulae [32]. MD simulation can also be used to investigate thermodynamic features as well as time-dependent (kinetic) processes.

Statistical mechanics is a discipline of physics that looks at macroscopic systems from a molecular perspective to deduce macroscopic phenomena from the properties of the molecules that make up the system and to forecast them [33]. Time-independent statistical averages are frequently used to connect the macroscopic

Fig. 1 MD simulation is commonly used to understand the macroscopic properties of a system using microscopic simulations via statistical mechanics



system to the microscopic system. In the following paragraphs, we will try to explain a few definitions of statistical mechanics to represent a physical system.

A thermodynamic state of a system is characterized by a set of parameters, such as temperature, pressure, and the number of particles, N . The equations of state and other fundamental thermodynamic equations can be used to calculate various thermodynamic properties. The atomic locations, q , and momenta, p , constitute the mechanical or microscopic state of a system, which can alternatively be considered coordinates in a multi-dimensional space (phase space). This space has $6N$ dimensions for a system of N particles. The state of the system is represented by G , a single point in phase space. A group of locations in a phase space that satisfies the criteria of a specific thermodynamic state is termed an ensemble. As a function of time, MD simulation generates a series of points in phase space that are part of the same ensemble and correspond to the various conformations and momenta of the system. There are descriptions of several different ensembles. An ensemble is a collection of all feasible systems with distinct microscopic states but the same macroscopic or thermodynamic state. There are several ensembles available for studying physical systems, such as microcanonical ensemble (NVE), canonical ensemble (NVT), isobaric-isothermal ensemble (NPT), and grand canonical ensemble (μ VT). A given number of atoms, a fixed volume, and a fixed energy characterize the thermodynamic state of NVE. This is similar to an isolated system. In NVT, the number of atoms, volume, and temperature are considered fixed. In NPT, the number of atoms, pressure, and temperature remains fixed. However, in μ VT, volume and temperature are fixed for a given chemical potential.

An experiment is frequently performed on a macroscopic sample containing a large number of atoms that sample a vast number of different conformations. Averages for experimental observables are defined using ensemble averages in statistical mechanics [34]. An ensemble average is a calculation that takes into account a large number of system copies at the same time (Fig. 1).

The ensemble average is computed as follows:

$$(A)\text{Ensemble} = \iint dp^N dr^N A(p^N, r^N) \rho(p^N, r^N)$$

where $A(p^N, r^N)$ is observable, defined as a function of the system's momenta (p) and locations (r). Integration is performed on all possible variables of r and p . The ensemble's probability density is given by

$$\rho(p^N, r^N) = \frac{1}{Q} \exp[-H(p^N, r^N)/k_B T]$$

where H represents the Hamiltonian, T is the temperature, k_B is Boltzmann's constant, and Q is the partition function.

$$Q = \iint dp^N dr^N \exp[-H(p^N, r^N)/k_B T]$$

This integral is highly difficult to calculate since it necessitates calculating all possible system states [35]. Because the points in an ensemble are generated sequentially in time in an MD simulation, the simulations must traverse through all conceivable states that match the specific thermodynamic constraints to calculate an ensemble average. Another method, which is used in MD simulations, is to calculate a temporal average of A , which is written as:

$$(A)\text{time} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{t=0}^{\tau} A(p^N(t), r^N(t)) dt \approx \frac{1}{M} \sum_{t=1}^M A(p^N, r^N)$$

where t is the time of simulation, M represents time steps, and $A(p^N, r^N)$ is the value of A at a particular instant.

MD simulation can compute only temporal averages, while the experimental observables are considered ensemble averages. The ergodic hypothesis, which is one of the fundamental principles of statistical mechanics, states that the temporal average equals the ensemble average [35]. The central premise is that if a system is allowed to grow indefinitely, it will pass through all possible states. As a result, one of the goals of an MD simulation is to create enough sample conformations to satisfy this equality [36]. Experimentally relevant data on structural and thermodynamic properties can be calculated with a reasonable resource of computing power. As the simulations have a defined time, it is important to sample enough phase space [37]. The average potential energy of the system is represented as

$$V = \langle V \rangle = \frac{1}{M} \sum_{i=1}^M V^i$$

where M represents the trajectory configurations and V^i is the potential energy of a particular configuration.

The average kinetic energy is expressed with the following equation:

$$K = \langle K \rangle = \frac{1}{M} \sum_{j=1}^M \left\{ \sum_{i=1}^N \frac{M_i}{2} v_i \cdot v_i \right\}_j$$

where M represents the number of configurations, N is the atoms number, m_i and v_i are the mass and velocity of the particle i , respectively. An MD simulation must last long enough to sample a large number of relevant conformations.

3 Classical Mechanics

Newton's second law, $F = ma$ (where F is the force applied on the particle, m is the particle's mass, and a is the particle's acceleration), is the sole foundation of the traditional MD simulation. It is possible to determine each atom's acceleration in a system using the force acting on each atom [38]. A trajectory that depicts the locations, velocities, and accelerations of the particles over time is created after integrating the equations of motion [37]. This method can be used to calculate the average values of the particle properties. Since it is a deterministic method, the system's state can be calculated at any point in time, past or future, if the positions and motions of each atom are known. MD simulation can be time-consuming and expensive; computers, on the other hand, are becoming robust and cheaper. Up to the nanosecond time scale, simulations of solvated proteins can be calculated; nonetheless, simulations of the millisecond time scale have also been recorded using high performance computing. Newton's equation of motion is expressed as:

$$F_i = m_i a_i$$

where F_i represents the force acting on particle i , while m_i and a_i are the mass and acceleration, respectively. The force can also be described as a potential energy gradient.

$$F_i = - \nabla_i V$$

Combining these two equations result to:

$$- \frac{dV}{dr_i} = m_i \frac{d^2 r_i}{dt^2}$$

where V represents the system's potential energy. This equation can be used to relate the derivative of potential energy to changes in position as a function of time.

3.1 Newton's Second Law of Motion

$$F = m \cdot a = m \cdot \frac{dv}{dt} = m \cdot \frac{d^2x}{dt^2}$$

Considering the acceleration as constant

$$a = \frac{dv}{dt}$$

After integration, the expression for the velocity can be written as

$$v = at + v_0$$

since

$$v = \frac{dx}{dt}$$

after further integration

$$x = v \cdot t + x_0$$

Combining the above equation with the velocity, we get the below relation that gives the value of x at time t as a function of the initial position (x_0), the acceleration (a), and the initial velocity (v_0).

$$x = \frac{1}{2}a * t^2 + v_0 * t + x_0$$

The acceleration is calculated using the derivative of potential energy with respect to the position (r).

$$a = - \frac{1}{m} \frac{dE}{dr}$$

Therefore, the initial positions of the atoms, an initial velocity distribution, as well as the acceleration determined by the gradient of the potential energy function are all required to construct a trajectory [39]. The positions and velocities at time zero determine the positions and velocities at every other time (t), as the motion equations are deterministic. The initial positions can be taken from experimental structures, such as the protein's X-ray crystal structure or NMR structure. The initial velocity distribution is commonly derived from a random distribution with magnitudes that

conform to the requisite temperature and that are corrected to ensure that there is no overall momentum, which is represented by

$$p = \sum_{i=1}^N m_i v_i = 0$$

The probabilities of an atom having a velocity v_x in the x direction at a temperature T are determined by selecting velocities, v_i , at random from a Maxwell-Boltzmann or Gaussian distribution.

$$p(v_{ix}) = \left(\frac{m_i}{2\pi k_b T} \right)^{\frac{1}{2}} \exp \left[-\frac{1}{2} \frac{m_i v_{ix}^2}{k_b T} \right]$$

The temperature can be obtained as follows:

$$T = \frac{1}{(3N)} \sum_{i=1}^N \frac{|p_i|^2}{2m_i}$$

where N represents the number of atoms in the system.

3.2 Integration Algorithms

The potential energy is a function of all atoms in a system's atomic locations ($3N$). The equations of motion have no analytic solution due to the intricate nature of this function; they must be solved numerically [40]. Several numerical techniques have been devised to integrate the equations of motion, such as the Verlet algorithm, Leap-frog algorithm, Velocity Verlet, and Beeman's algorithm. While choosing an algorithm, one should consider that the algorithm should conserve energy and momentum. It should be computationally efficient and allow a long-time step for integration. The locations, velocities, and accelerations of all the integration techniques are assumed to be approximated by a Taylor series expansion:

$$r(t + \delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2 + \dots$$

$$v(t + \delta t) = v(t) + a(t)\delta t + \frac{1}{2}b(t)\delta t^2 + \dots$$

$$a(t + \delta t) = a(t) + b(t)\delta t + \dots$$

where r represents the position, v is the velocity (the first derivative with respect to time), and a is the acceleration (the second derivative with respect to time), etc.

3.2.1 Verlet Algorithm

To derive the **Verlet** algorithm, one can write:

$$r(t + \delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2$$

$$r(t - \delta t) = r(t) - v(t)\delta t + \frac{1}{2}a(t)\delta t^2$$

After summing the above two equations:

$$r(t + \delta t) = 2r(t) - r(t - \delta t) + \frac{1}{2}a(t)\delta t^2$$

The Verlet method calculates new positions at time $t + dt$ by combining locations and accelerations at time t with positions from time $t - dt$. There are no stated velocities in the Verlet algorithm. The Verlet algorithm is simple with minimal storage needs, but the disadvantage is that the algorithm is of moderate precision [41].

3.2.2 The Leap-Frog Algorithm

In this method, the velocities are calculated at time $t + 1/2dt$. Further, these are used to find the positions (r) at time $t + dt$. In this way, the velocities *leap* over the positions, and then the positions *leap* over the velocities [42].

$$r(t + \delta t) = r(t) + v\left(t + \frac{1}{2}\delta t\right)\delta t$$

$$v\left(t + \frac{1}{2}\delta t\right) = v\left(t - \frac{1}{2}\delta t\right) + a(t)\delta t$$

This approach has the advantage of explicitly calculating velocities; however, it has the disadvantage of not doing so simultaneously with the positions. The relationship can be used to approximate the velocities at time t .

$$v(t) = \frac{1}{2} \left[v\left(t - \frac{1}{2}\delta t\right) + v\left(t + \frac{1}{2}\delta t\right) \right]$$

3.2.3 The Velocity Verlet Algorithm

This algorithm returns positions, velocities, and accelerations at time t . Precision is not uncompromised.

$$r(t + \delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2$$

$$r(t + \delta t) = v(t) + \frac{1}{2}[a(t) + a(t + \delta t)]\delta t$$

3.2.4 Beeman's Algorithm

This algorithm is very similar to the Verlet algorithm.

$$r(t + \delta t) = r(t) + v(t)\delta t + \frac{2}{3}a(t)\delta t^2 - \frac{1}{6}a(t - \delta t)\delta t^2$$

$$v(t + \delta t) = v(t) + v(t)\delta t + \frac{1}{3}a(t)\delta t + \frac{5}{6}a(t)\delta t - \frac{1}{6}a(t - \delta t)\delta t$$

This algorithm has the advantage of providing a more accurate expression for velocities and better energy conservation [43]. The disadvantage is that more complex expressions increase the cost of the calculation.

4 Principle of MD Simulation

The Born-Oppenheimer approximation, which separates the slow atomic degrees of freedom from the fast motion of light electrons, is the core of MD simulation [43]. Binding in solids and molecules is due to the interaction of electrons with the atomic core, which is seen almost at rest by the electrons. This interaction also provides interatomic forces when the atom cores are treated as classical particles. While it was required to approximate these forces with appropriate interatomic potentials initially, the introduction of fast electronic computers and the Car-Parrinello method enabled the interaction to be treated on a first-principles basis, allowing predictive quantitative simulations [43].

4.1 Periodic Boundary Condition

Periodic boundary conditions (PBCs) are a group of boundary conditions that are used to approximate a large (infinite) system with a small component called a unit cell. Simulations and mathematical modelling frequently employ PBCs. PBCs are used in MD simulation to eliminate finite-size boundary effects and to make the system similar to an infinite one at the expense of potential periodicity effects [43]. The existence of PBC ensures that every atom that exits a simulation box via the right-hand face must re-enter via the left-hand face. If we look at the face of the

simulation box opposite the one from where the protein is protruding in the case of a large protein, we will notice a hole in the solvent. The molecule(s) shift from where they were initially situated within the box since they are free to diffuse around in most simulations [44]. During the simulation, the box is not centred on anything. Molecules are not automatically made complete. Using PBCs to solve the surface-effects problem is an alternate and preferred method. PBCs can be approached in various ways, but we will stick to the minimum-image convention. We must first understand the concept of a unit cell before discussing PBCs. A unit cell is the simplest representation of a system. If we are imitating a crystal, we might pick a tiny cell with a few hundred atoms that match the desired crystal form. When simulating a gas, our unit cell could be a small volume containing several hundred gas molecules. We can start with a tiny unit cell volume, even if the purpose of the simulation is to obtain insight into bulk crystal or gas properties (easily $>10^{10}$ molecules). Then we can make neighbouring copies (images) of the unit cell that duplicate the contents of the unit cell in adjacent volumes. The images are a duplicate of the original simulation region and are used to lessen or eliminate border effects by providing an equivalent surrounding environment of atoms to every atom in the unit cell, independent of position in the unit cell. We can update the original unit cell's positions, forces, and velocity. Mirror replicas of the unit cell will be updated in the surrounding image cells. As a result, the atoms in the image cells have no physical significance on their own and are only constructs for PBCs.

The seeming “never-ending” aspect of the unit cell, that is, when an atom exits through a wall in the unit cell, it subsequently re-enters on the opposite side of the unit cell with the same velocity, is one of the simple and attractive effects of PBCs [45]. The layout of the image cells supports this continuity because as an atom departs the unit cell, the same atom's image may be seen entering the unit cell from an image.

4.2 Ewald Summation

Ewald summation is a technique for calculating long-range interactions in periodic systems, such as electrostatic interactions [46]. The total electrostatic energy of NN particles and their periodic images can be calculated using the following:

$$V = \frac{f}{2} \sum_{n_x} \sum_{n_y} \sum_{n_z} \sum_i^N \sum_j^N \frac{q_i q_j}{r_{ij,n}}$$

The box index vector is $(n_x, n_y, n_z) = n$, and the asterisk mark designates that terms with $i = j$ should be omitted in case $(n_x, n_y, n_z) = (0,0,0)$. The distance $r_{ij,n}$, as opposed to the minimum image, represents the actual distance between the charges. Although incredibly slow, this sum is conditionally convergent. Ewald summation was initially developed to determine the long-range interactions of the periodic

images in crystals. The goal is to split the single, slowly convergent sum into two components that swiftly converge and a constant term.

4.3 Particle Mesh Ewald (PME) Method

The Particle Mesh Ewald (PME) method is given by Tom Darden to enhance the reciprocal sum performance. The charges are interpolated to a grid instead of just adding wave vectors. Cardinal B-spline interpolation or smooth PME (SPME), is used in GROMACS [47]. Using a 3D FFT technique, the grid is then Fourier transformed, and the reciprocal energy term is calculated by summing the grid in k -space. The inverse transformation is used to calculate the potential at grid points, and interpolation factors are used to determine the forces working on each atom. In a medium to large systems, the PME technique is noticeably faster than standard Ewald summation. Ewald may still be preferable on relatively small systems to save time setting up grids and transforms. The PME direct space potential is moved by a constant in the Verlet cut-off scheme so that the potential is zero at the cut-off. In contrast to the Lennard-Jones potential, where all shifts add up, this shift is minor, and because the net system charge is almost zero, the total shift is also minimal. We nonetheless apply the shift to make the potential precisely equal to the integral of the force.

4.4 Thermostat in MD

By altering the system's temperature in some way, thermostats are intended to assist a simulation sample from the appropriate ensemble (i.e. NVT or NPT). We must first define what is meant by temperature. The "instantaneous (kinetic) temperature" in simulations is typically calculated from the system's kinetic energy using the equipartition theorem. In other words, the system's total kinetic energy is used to calculate the temperature [48]. The purpose of a thermostat is not to maintain a constant temperature because doing so would mean fixing the total kinetic energy, which is wrong and not what NVT or NPT are intended to do. Instead, it guarantees that a system's average temperature is correct.

Consider a glass of water placed in a space to understand this case. Consider estimating the kinetic energy of a few molecules in a small area of the glass by looking at them extremely closely [49]. Because there are so few particles, you would not anticipate the kinetic energy to be perfectly constant; instead, you would anticipate fluctuations in the kinetic energy. The fluctuations in the average decrease as you average across more and more particles, and when you ultimately consider the entire glass, you can conclude that it has a "constant temperature". Compared to a glass of water, MD simulations are quite small, which causes larger fluctuations [50]. Therefore, it would be fair to consider the role of the thermostat in this situation

to ensure that we have the proper average temperature and fluctuations of the correct size.

4.5 Solvent Models

A solvent model is a computer technique used in computational chemistry to predict the behaviour of solvated condensed phases. Simulations and thermodynamic calculations for reactions and processes that occur in solutions are made possible by solvent models [51]. Environmental, chemical, and biological processes are among them. Such computations can result in new predictions about the physical processes due to greater understanding. Generally, there are two groups of models: explicit and implicit models, each of which has advantages and disadvantages of its own [52]. Implicit models often have good computing efficiency and can provide a good description of the behaviour of the solvent, but they are unable to consider the local variations in solvent density near a solute molecule. When water is used as a solvent, the density fluctuation behaviour caused by solvent ordering around a solute is more common. Explicit models can provide a physical, spatially detailed description of the solvent but are frequently inefficient in terms of computational efficiency [53]. Although many of these explicit models may fail to replicate specific experimental results, this is often due to differences in fitting methods and parameterization.

4.6 Energy Minimization

Energy minimization is the process of arranging a group of atoms in space in such a way so that the net interatomic force acting on each atom is as close to zero as possible while it is stationary on the potential energy surface (PES) [53]. The atoms could combine to form a single molecule, an ion, a condensed phase, a transition state, or a combination of these.

5 Current Tools for Molecular Dynamics

Various tools are available for performing MD simulations both in proprietary and open-source domains. Some of them are discussed below:

5.1 *Gromacs*

Gromacs is an MD simulation software package designed primarily for simulating proteins, nucleic acids, and lipids. It was created in the University of Groningen's Biophysical Chemistry department, and it is now maintained by various contributors from research institutions all over the world. It is one of the widely used software available that can run on a computer with basic configuration as well as on high-end workstations. It is a freely available open-source software distributed under the General Public License (GNU) [54, 55].

5.2 *Amber*

A set of biomolecular simulation tools are included in Amber. It was started in the late 1970s, and a vibrant development community continues to maintain it. Two objects are being referred to by the term "Amber". First, it is a collection of molecular mechanical force fields for simulating biomolecules available in the public domain. Second, it is a collection of molecular simulation programs that also includes demonstrations. AmberTools21 and Amber20 are the two components of Amber. AmberTools21 can be used without Amber20, but not the other way around [56].

5.3 *CHARMM*

CHARMM is a molecular simulation tool with extensive applicability to many-particle systems that supports multi-scale methods, including quantum mechanics/molecular mechanics (QM/MM), molecular mechanics/coarse-grained (MM/CG), a variety of implicit solvent models, and a large collection of energy functions. It targets biomolecules such as proteins, small molecules, nucleic acids, lipids, and carbohydrates found in solution, crystals, and membrane environments. CHARMM also have a wide range of applications for inorganic materials. CHARMM includes a comprehensive set of tools for analysis and model construction. It performs well on a variety of systems, such as GPUs and parallel clusters [57].

5.4 *NAMD*

NAMD is a parallel MD programme designed for the high-performance modelling of large biomolecular systems. It won the Gordon Bell Award in 2002, the Sidney Fernbach Award in 2012, and the Gordon Bell Prize in 2020. As NAMD is based on Charm++ parallel objects, it can scale from hundreds to over half a million cores for

larger simulations. NAMD uses the well-known molecular graphics application VMD to set up the simulation and analyse the trajectory, in addition to being file-compatible with AMBER, CHARMM, and X-PLOR. The source code for NAMD is freely available. NAMD is available as a self-built project or as downloadable binaries for many different platforms [58].

5.5 *HyperChem*

HyperChem is a powerful molecular modelling system with a reputation for excellence, adaptability, and user-friendliness. HyperChem provides access to several molecular modelling tools more than any other programme by fusing 3D visualization and animation with different computational approaches. It incorporates several computational techniques, including density functional theory, semi-empirical and ab-initio molecular orbital methods, MD, and MM. HyperChem includes HyperChem Data and HyperNMR. Among the frequently added features are CHARMM protein simulations, molecules in magnetic fields, compatibility with third-party applications, etc. Both large and small molecules can be used with HyperChem. It also supports scripting [59].

6 GUI-Based Software for MD Trajectories Analysis

The MD simulation output trajectories can be visualized using GUI-based software. The following are some of the most popular software:

6.1 *Visual Molecular Dynamics (VMD)*

A research group of theoretical biophysics at the University of Illinois created visual molecular dynamics (VMD) [60–62]. It is a highly effective tool for observing and investigating various biological systems, including nucleic acids, proteins, lipids, and carbohydrates. It supports a wide range of formats, such as PDB and GROMOS for biomolecules. It can handle a massive amount of data processing to display changes in a trajectory [63]. The molecules can be seen as animated, and the input trajectory can also be used to make a movie. It works with any operating system that has a basic computer configuration. It is included with NAMD as well. The additional features of VMDs include [64]:

1. It can be used to visualize macromolecules.
2. An amino acid and atoms can be chosen.
3. Structure alignment can be performed.

4. Support for user activity logs is present.
5. Support for the Raster3D format is available.
6. Ramachandran plots can be generated.
7. It supports various molecular image types.
8. It supports command lines.
9. It utilizes vectors and arrays.
10. It has JavaScript support.

6.2 *PyMOL*

Structural biologists extensively use the PyMOL software [65]. PyMOL can accept various file formats, including SDF, Mol2, PDB, etc. The trajectory can be imported, and the simulation results can be analysed on PyMOL. A surface view model can be generated. To further study the MD simulation results, several additional plug-ins are available. The user can use this tool to create high-quality figures as well as animated movies.

6.3 *Chimera*

UCSF Chimera is a sophisticated tool for molecular modelling systems that is free for academic usage [66]. Advanced UCSF ChimeraX is also freely available for academic use. The Gromacs and Amber trajectory formats are supported by Chimera 1.13.1 and later versions. Following the import of these trajectories, the user can create a movie with a time frame and produce attractive images. Aligning two or more structures is possible. The surface cavity analysis during the trajectory run can also be generated. It supports the command line option and has a variety of functions.

7 Other Advanced MD Simulation Methods

7.1 *Metadynamics*

An improved sampling technique known as metadynamics uses a set of collective variables (CVs) that specify transitions along a reaction coordinate to explain the system. The system's position in this CV space is established during the simulation, and then positive biasing Gaussian functions are added, modifying the system's Hamiltonian [67].

$$H = T + V + \sum V_{\text{GAUSSH}} = T + V + + \sum V_{\text{GAUSSH}}$$

As a result of the accumulation of these Gaussian functions in properly sampled regions of the CV space, the system can more easily navigate through regions of CV space that correspond to free energy maxima while simulating the unmodified Hamiltonian. The simulation can now examine the entire energy landscape. Knowing the sampling of the modified Hamiltonian and the deposited Gaussian functions allows one to retrieve the free energy surface of the unmodified Hamiltonian.

7.2 Umbrella Sampling

The purpose of any MD simulation is to sample all possible states in which a molecule may exist. Based on this method, the probability (free energy) for the molecule to be in any state can be determined. Often, some protein states are separated from others by very high energy barriers. Sometimes, it would take years of conventional MD simulation to go through all molecular states. Umbrella sampling allows us to accelerate the sampling by flattening those hills and ridges, which prevents MD simulation from accessing certain states. In umbrella sampling, the energy landscape is flattened by adding artificial umbrella potentials that are supposed to mirror and thus annihilate the real barriers. However, making an umbrella potential account for all degrees of freedom in the system would be difficult. Hence, the umbrella potential involves only a few (one to three) degrees of freedom, often called CVs or reaction coordinates. The sampling of a system is considered complete when it has visited all values of CVs for an accurate and unbiased calculation of state probabilities [68].

8 Structural Parameters to Analyse MD Simulation Data

8.1 Root Mean Square Deviation (RMSD)

The root mean square deviation (RMSD) is the Euclidean distance between the structure and a reference structure that measures the relaxation between the structures [69]. It is a common way to quantify the distance of structural coordinates. It determines how far apart, on average, a group of atoms, such as the protein's backbone atoms, are from one another [70]. Calculating the RMSD between two sets of atomic coordinates, such as two points in time from the trajectory, measures how much the protein structure has changed. It is possible to compute the RMSD for each residue, the backbone, the side chains, and C-alpha. It is calculated with reference to the simulation time [71]. A lower RMSD value indicates a very stable structure over the course of a simulation.

8.2 *Root Mean Square Fluctuation (RMSF)*

The average variation of a particle over time from a reference position is measured by the root mean square fluctuation (RMSF) [69]. As a result, RMSF examines the structural elements that deviate the most from their mean structure. The variability around each atom's average position is captured by the RMSF. This reveals information on the flexibility of the protein's various regions and relates to the crystallographic B-factors. It can be used to check whether the simulation findings are consistent with the crystal structure because one would typically anticipate similar profiles for the RMSF and the B-factors. Atoms in bends and coils fluctuate more than in helices and sheets; hence they have lower RMSF values, whereas bends and coils have higher RMSF values.

8.3 *Radius of Gyration (Rg)*

The measurement of the radius of gyration (R_g) indicates the shape and compactness of a molecule at a particular time. The gyrating radius is compared to the hydrodynamic radius that can be measured empirically [69]. Additionally, this provides the individual components that are equivalent to the eigenvalues of the inertia matrix. This means that the first component corresponds to the molecule's longest axis and the last to its shortest. The three axes effectively provide a global indication of the shape of the molecule [72].

8.4 *Solvent Accessible Surface Area (SASA)*

The area of the protein that is accessible to solvent is known as solvent accessible surface area (SASA), which can be further divided into a hydrophilic and hydrophobic SASA [73, 74]. The SASA of the expanded form of the protein is higher than that of the folded globular protein. It is known that when a temperature of a system rises, proteins begin to unfold and expose their hydrophobic interiors to the solvent. SASA consequently increases upon unfolding. Additionally, the SASA can be used to estimate the free energy of solvation together with a few empirical parameters [75].

8.5 *Hydrogen Bonds*

The number of internal hydrogen bonds with a protein or external hydrogen bonds between a protein and its surrounding solvent is another characteristic that can be

informative [76, 77]. The distance between a donor-H acceptor pair and the donor-H acceptor angle can be used to determine the presence or absence of a hydrogen bond. Hydrogen bonds are vital for maintaining protein secondary structures; therefore, simulations of protein folding must adequately represent the hydrogen bond interactions. Hydrogen bonding is treated as a non-bonded interaction in modern classical force fields where electrostatic interactions are predominant. Atomic charges, on the other hand, are fixed and are established in a mean-field fashion in the frequently utilized non-polarizable force fields. The native structure cannot be appropriately populated when the non-polarizable AMBER force field is used in the folding simulations of small peptides. When the polarization effect is added to the simulation using either the on-the-fly charge fitting or the polarizable hydrogen bond model, the native structure becomes more prominent in the free energy landscape. These results emphasize how crucial the electrostatic polarization effect is for simulating proteins [78].

9 Summary

MD simulation has been a popular method for understanding the dynamic representation of any biological system for the last few decades [79–81]. In recent years, GPU-based high computational capability systems have significantly reduced the time required in the MD simulation of biological macromolecules. It is a handy technique for understanding molecular interactions such as protein–protein and protein–ligand interactions, as well as protein folding. It creates the cell-like environment around the macromolecules by considering pH and the surrounding molecules such as water, lipids, ions, as well as co-enzymes. It provides atomic-level interaction details that offer insights into how molecules function. Tools such as MM-PBSA can be used to predict the free energy of binding, various energy constituents, and contribution to binding with a small molecule at the residue level. The implementation of the QM and MM method in the MD simulation has improved the accuracy of these predictions. MD simulation can thus be used to investigate the dynamics of a biological system by selecting an appropriate model and physical conditions.

References

1. S.M. Kelly, N.C. Price, The application of circular dichroism to studies of protein folding and unfolding. *BBA-Protein Struct. M.* **1338**, 161–185 (1997)
2. U. Mayor, C.M. Johnson, V. Daggett, A.R. Fersht, Protein folding and unfolding in microseconds to nanoseconds by experiment and simulation. *Proc. Natl. Acad. Sci.* **97**, 13518–13522 (2000)
3. H.A. Scheraga, M. Khalili, A. Liwo, Protein-folding dynamics: overview of molecular simulation techniques. *Annu. Rev. Phys. Chem.* **58**, 57–83 (2007)

4. B.J. Alder, T.E. Wainwright, Phase transition for a hard sphere system. *J. Chem. Phys.* **27**, 1208–1209 (1957)
5. B.J. Alder, T.E. Wainwright, Studies in molecular dynamics. I. General method. *J. Chem. Phys.* **31**, 459–466 (1959)
6. A. Rahman, Correlations in the motion of atoms in liquid argon. *Phys. Rev.* **136**, A405 (1964)
7. A. Rahman, F.H. Stillinger, Molecular dynamics study of liquid water. *J. Chem. Phys.* **55**, 3336–3359 (1971)
8. J.A. McCammon, B.R. Gelin, M. Karplus, Dynamics of folded proteins. *Nature* **267**, 585–590 (1977)
9. A. Khandelwal, V. Lukacova, D. Comez, D.M. Kroll, S. Raha, S. Balaz, A combination of docking, QM/MM methods, and MD simulation for binding affinity estimation of metalloprotein ligands. *J. Med. Chem.* **48**, 5437–5447 (2005)
10. M. Billeter, G. Wagner, K. Wüthrich, Solution NMR structure determination of proteins revisited. *J. Biomol. NMR* **42**, 155–158 (2008)
11. C. Park, J. Jung, G.J. Yun, Multiscale micromorphic theory compatible with MD simulations in both time-scale and length-scale. *Int. J. Plast.* **129**, 102680 (2020)
12. M.C. Zwier, L.T. Chong, Reaching biological timescales with all-atom molecular dynamics simulations. *Curr. Opin. Pharmacol.* **10**, 745–752 (2010)
13. J.L. Binder, J. Berendzen, A.O. Stevens, Y. He, J. Wang, N.V. Dokholyan, T.I. Oprea, AlphaFold illuminates half of the dark human proteins. *Curr. Opin. Struct. Biol.* **74**, 102372 (2022)
14. E. Callaway, “The entire protein universe”: α I predicts shape of nearly every known protein. *Nature* **608**, 15–16 (2022)
15. R.O. Dror, R.M. Dirks, J.P. Grossman, H. Xu, D.E. Shaw, Biomolecular simulation: a computational microscope for molecular biology. *Annu. Rev. Biophys.* **41**, 429–452 (2012)
16. V. Rajendran, R. Shukla, H. Shukla, T. Tripathi, Structure-function studies of the asparaginyl-tRNA synthetase from *Fasciola gigantica*: understanding the role of catalytic and non-catalytic domains. *Biochem. J.* **475**, 3377–3391 (2018)
17. T. Pandey, R. Shukla, H. Shukla, A. Sonkar, T. Tripathi, A.K. Singh, A combined biochemical and computational studies of the rho-class glutathione s-transferase sll1545 of *synechocystis* PCC 6803. *Int. J. Biol. Macromol.* **94**, 378–385 (2017)
18. A. Sonkar, H. Shukla, R. Shukla, J. Kalita, T. Pandey, T. Tripathi, UDP-N-acetylglucosamine enolpyruvyl transferase (MurA) of *acinetobacter baumannii* (AbMurA): structural and functional properties. *Int. J. Biol. Macromol.* **97**, 106–114 (2017)
19. B. Brooks, M. Karplus, Harmonic dynamics of proteins: normal modes and fluctuations in bovine pancreatic trypsin inhibitor. *Proc. Natl. Acad. Sci.* **80**, 6571–6575 (1983)
20. A.T. Brünger, C.L. Brooks 3rd, M. Karplus, Active site dynamics of ribonuclease. *Proc. Natl. Acad. Sci.* **82**, 8458–8462 (1985)
21. J. Smith, S. Cusack, U. Pezzecca, B. Brooks, M. Karplus, Inelastic neutron scattering analysis of low frequency motion in proteins: a normal mode study of the bovine pancreatic trypsin inhibitor. *J. Chem. Phys.* **85**, 3636–3654 (1986). <https://doi.org/10.1063/1.450935>
22. C.C. David, D.J. Jacobs, Principal component analysis: a method for determining the essential dynamics of proteins, in *Protein Dynamics*, ed. by D. Livesay, (Springer, Heidelberg, 2014), pp. 193–226
23. A. Wolf, K.N. Kirschner, Principal component and clustering analysis on molecular dynamics data of the ribosomal L11 23S subdomain. *J. Mol. Model.* **19**, 539–549 (2013)
24. F. Colonna-Cesari, D. Perahia, M. Karplus, H. Eklund, C.I. Brädén, O. Tapia, Interdomain motion in liver alcohol dehydrogenase. Structural and energetic analysis of the hinge bending mode. *J. Biol. Chem.* **261**, 15273–15280 (1986)
25. L. Nilsson, G.M. Clore, A.M. Gronenborn, A.T. Brünger, M. Karplus, Structure refinement of oligonucleotides by molecular dynamics with nuclear overhauser effect interproton distance restraints: application to 5' d (CGTACG) 2. *J. Mol. Biol.* **188**, 455–475 (1986)

26. D.A. Case, M. Karplus, Dynamics of ligand binding to heme proteins. *J. Mol. Biol.* **132**, 343–368 (1979)
27. P. Banáš, P. Jurečka, N.G. Walter, J. Šponer, M. Otyepka, Theoretical studies of RNA catalysis: hybrid QM/MM methods and their comparison with MD and QM. *Methods* **49**, 202–216 (2009)
28. S. Ahmadi, L. Barrios Herrera, M. Chehelamirani, J. Hostaš, S. Jalife, D.R. Salahub, Multiscale modeling of enzymes: QM-cluster, QM/MM, and QM/MM/MD: a tutorial review. *Int. J. Quantum Chem.* **118**, e25558 (2018)
29. A.W. Götz, M.A. Clark, R.C. Walker, An extensible interface for QM/MM molecular dynamics simulations with AMBER. *J. Comput. Chem.* **35**, 95–108 (2014)
30. R.C. Tolman, *The Principles of Statistical Mechanics* (Courier Corporation, New York, 1979)
31. R.H. Fowler, *Statistical Mechanics* (Cambridge University Press, Cambridge, 1967)
32. Y. Levin, R. Pakter, F.B. Rizzato, T.N. Teles, F.P. Benetti, Nonequilibrium statistical mechanics of systems with long-range interactions. *Phys. Rep.* **535**, 1–60 (2014)
33. M. Takano, K. Nagayama, A. Suyama, Investigating a link between all-atom model simulation and the Ising-based theory on the helix–coil transition: equilibrium statistical mechanics. *J. Chem. Phys.* **116**, 2219–2228 (2002)
34. M.J. Uline, D.W. Siderius, D.S. Corti, On the generalized equipartition theorem in molecular dynamics ensembles and the microcanonical thermodynamics of small systems. *J. Chem. Phys.* **128**, 124301 (2008)
35. J.R. Ray, H. Zhang, Correct microcanonical ensemble in molecular dynamics. *Phys. Rev. E* **59**, 4781 (1999)
36. A. Johnson, T. Johnson, A. Khan, Thermostats in molecular dynamics simulations. *UMass.* **1**, 29 (2012)
37. P.V. Coveney, S. Wan, On the calculation of equilibrium thermodynamic properties from molecular dynamics. *Phys. Chem. Chem. Phys.* **18**, 30236–30240 (2016)
38. R.D. Gregory, *Classical Mechanics* (Cambridge University Press, Cambridge, 2006)
39. T. Kibble, F.H. Berkshire, *Classical Mechanics* (World Scientific Publishing Company, Singapore, 2004)
40. H. Grubmüller, H. Heller, A. Windemuth, K. Schulten, Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Mol. Simul.* **6**, 121–142 (1991)
41. Q. Spreiter, M. Walter, Classical molecular dynamics simulation with the velocity Verlet algorithm at strong external magnetic fields. *J. Comput. Phys.* **152**, 102–119 (1999)
42. W.F. Van Gunsteren, H.J. Berendsen, A leap-frog algorithm for stochastic dynamics. *Mol. Simul.* **1**, 173–185 (1988)
43. R.W. Pastor, B.R. Brooks, A. Szabo, An analysis of the accuracy of Langevin and molecular dynamics algorithms. *Mol. Phys.* **65**, 1409–1419 (1988)
44. J. Shimada, H. Kaneko, T. Takada, Efficient calculations of Coulombic interactions in biomolecular simulations with periodic boundary conditions. *J. Comput. Chem.* **14**, 867–878 (1993)
45. W. Tian, L. Qi, X. Chao, J. Liang, M. Fu, Periodic boundary condition and its numerical implementation algorithm for the evaluation of effective mechanical properties of the composites with complicated micro-structures. *Compos. Part B* **162**, 1–10 (2019)
46. A.Y. Toukmaji, J.A. Board Jr., Ewald summation techniques in perspective: a survey. *Comput. Phys. Commun.* **95**, 73–92 (1996)
47. J. Kolafa, J.W. Perram, Cutoff errors in the Ewald summation formulae for point charge systems. *Mol. Simul.* **9**, 351–368 (1992)
48. E.A. Koopman, C.P. Lowe, Advantages of a Lowe–Andersen thermostat in molecular dynamics simulations. *J. Chem. Phys.* **124**, 204103 (2006)
49. E. Rosta, N.-V. Buchete, G. Hummer, Thermostat artifacts in replica exchange molecular dynamics simulations. *J. Chem. Theory Comput.* **5**, 1393–1399 (2009)
50. R. Kutteh, R.B. Jones, Rigid body molecular dynamics with nonholonomic constraints: molecular thermostat algorithms. *Phys. Rev. E* **61**, 3186–3198 (2000). <https://doi.org/10.1103/PhysRevE.61.3186>

51. P. Ferrara, J. Apostolakis, A. Caffisch, Evaluation of a fast implicit solvent model for molecular dynamics simulations. *Proteins* **46**, 24–33 (2002). <https://doi.org/10.1002/prot.10001>
52. H. Nguyen, D.R. Roe, C. Simmerling, Improved generalized born solvent model parameters for protein simulations. *J. Chem. Theory Comput.* **9**, 2020–2034 (2013). <https://doi.org/10.1021/ct3010485>
53. A. Malevanets, R. Kapral, Mesoscopic model for solvent dynamics. *J. Chem. Phys.* **110**, 8605–8613 (1999). <https://doi.org/10.1063/1.478857>
54. D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A.E. Mark, H.J.C. Berendsen, GROMACS: fast, flexible, and free. *J. Comput. Chem.* **26**, 1701–1718 (2005). <https://doi.org/10.1002/jcc.20291>
55. S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M.R. Shirts, J.C. Smith, P.M. Kasson, D. van der Spoel, B. Hess, E. Lindahl, GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* **29**, 845–854 (2013). <https://doi.org/10.1093/bioinformatics/btt055>
56. D.A. Case, T.E. Cheatham, T. Darden, H. Gohlke, R. Luo, K.M. Merz, A. Onufriev, C. Simmerling, B. Wang, R.J. Woods, The Amber biomolecular simulation programs. *J. Comput. Chem.* **26**, 1668–1688 (2005). <https://doi.org/10.1002/jcc.20290>
57. B.R. Brooks, C.L. Brooks, A.D. MacKerell, L. Nilsson, R.J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caffisch, L. Caves, Q. Cui, A.R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R.W. Pastor, C.B. Post, J.Z. Pu, M. Schaefer, B. Tidor, R.M. Venable, H.L. Woodcock, X. Wu, W. Yang, D.M. York, M. Karplus, CHARMM: the biomolecular simulation program. *J. Comput. Chem.* **30**, 1545–1614 (2009). <https://doi.org/10.1002/jcc.21287>
58. J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kalé, K. Schulten, Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **26**, 1781–1802 (2005). <https://doi.org/10.1002/jcc.20289>
59. M. Froimowitz, HyperChem: a software package for computational chemistry and molecular modeling. *BioTechniques* **14**, 1010–1013 (1993)
60. W. Humphrey, A. Dalke, K. Schulten, VMD: visual molecular dynamics. *J. Mol. Graph.* **14**, 33–38., 27–28 (1996). [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5)
61. J. Hsin, A. Arkipov, Y. Yin, J.E. Stone, K. Schulten, Using VMD: an introductory tutorial. *Curr. Protoc. Bioinformatics* **Chapter 5**, Unit 5.7 (2008). <https://doi.org/10.1002/0471250953.bi0507s24>
62. B. Knapp, N. Lederer, U. Omasits, W. Schreiner, vmdICE: a plug-in for rapid evaluation of molecular dynamics simulations using VMD. *J. Comput. Chem.* **31**, 2868–2873 (2010). <https://doi.org/10.1002/jcc.21581>
63. S. Falsafi-Zadeh, Z. Karimi, H. Galehdari, VMD DisRg: new user-friendly implement for calculation distance and radius of gyration in VMD program. *Bioinformatics* **8**, 341–343 (2012). <https://doi.org/10.6026/97320630008341>
64. I.V. Likhachev, N.K. Balabaev, O.V. Galzitskaya, Available instruments for analyzing molecular dynamics trajectories. *Open Biochem. J.* **10**, 1–11 (2016). <https://doi.org/10.2174/1874091X01610010001>
65. D. Seeliger, B.L. de Groot, Ligand docking and binding site analysis with PyMOL and autodock/Vina. *J. Comput. Aided Mol. Des.* **24**, 417–422 (2010). <https://doi.org/10.1007/s10822-010-9352-6>
66. E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, T.E. Ferrin, UCSF chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.* **25**, 1605–1612 (2004). <https://doi.org/10.1002/jcc.20084>
67. D. Hollas, L. Šišťák, E.G. Hohenstein, T.J. Martínez, P. Slavíček, Nonadiabatic Ab initio molecular dynamics with the floating occupation molecular orbital-complete active space configuration interaction method. *J. Chem. Theory Comput.* **14**, 339–350 (2018). <https://doi.org/10.1021/acs.jctc.7b00958>

68. G.M. Torrie, J.P. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling. *J. Comput. Phys.* **23**, 187–199 (1977). [https://doi.org/10.1016/0021-9991\(77\)90121-8](https://doi.org/10.1016/0021-9991(77)90121-8)
69. R. Brüschweiler, R.M.S.D. Efficient, Measures for the comparison of two molecular ensembles. Root-mean-square deviation. *Proteins* **50**, 26–34 (2003). <https://doi.org/10.1002/prot.10250>
70. O. Carugo, How root-mean-square distance (r.m.s.d.) values depend on the resolution of protein structures that are compared. *J. Appl. Cryst.* **36**, 125–128 (2003). <https://doi.org/10.1107/S0021889802020502>
71. R.K. Pathak, M. Baunthiyal, R. Shukla, D. Pandey, G. Taj, A. Kumar, In silico identification of mimicking molecules as defense inducers triggering jasmonic acid mediated immunity against alternaria blight disease in brassica species. *Front. Plant Sci.* **8**, 609 (2017). <https://doi.org/10.3389/fpls.2017.00609>
72. M.I. Lobanov, N.S. Bogatyreva, O.V. Galzitskaia, Radius of gyration is indicator of compactness of protein structure. *Mol Biol (Mosk)* **42**, 701–706 (2008)
73. E. Durham, B. Dorr, N. Woetzel, R. Staritzbichler, J. Meiler, Solvent accessible surface area approximations for rapid and accurate protein structure prediction. *J. Mol. Model.* **15**, 1093–1108 (2009). <https://doi.org/10.1007/s00894-009-0454-9>
74. Y. Mazola, O. Guirola, S. Palomares, G. China, C. Menéndez, L. Hernández, A. Musacchio, A comparative molecular dynamics study of thermophilic and mesophilic β -fructosidase enzymes. *J. Mol. Model.* **21**, 228 (2015). <https://doi.org/10.1007/s00894-015-2772-4>
75. J.A. Marsh, S.A. Teichmann, Relative solvent accessible surface area predicts protein conformational changes upon binding. *Structure* **19**, 859–867 (2011). <https://doi.org/10.1016/j.str.2011.03.010>
76. M.S. Weiss, M. Brandl, J. Sühnel, D. Pal, R. Hilgenfeld, More hydrogen bonds for the (structural) biologist. *Trends Biochem. Sci.* **26**, 521–523 (2001). [https://doi.org/10.1016/s0968-0004\(01\)01935-1](https://doi.org/10.1016/s0968-0004(01)01935-1)
77. C.N. Pace, H. Fu, K. Lee Fryar, J. Landua, S.R. Trevino, D. Schell, R.L. Thurlkill, S. Imura, J.M. Scholtz, K. Gajiwala, J. Sevcik, L. Urbanikova, J.K. Myers, K. Takano, E.J. Hebert, B.A. Shirley, G.R. Grimsley, Contribution of hydrogen bonds to protein stability. *Protein Sci.* **23**, 652–661 (2014). <https://doi.org/10.1002/pro.2449>
78. Y. Gao, Y. Mei, J.Z.H. Zhang, Treatment of hydrogen bonds in protein simulations, in *Advanced Materials for Renewable Hydrogen Production, Storage and Utilization*, ed. by J. Liu, (InTech, Rijeka, 2015). <https://doi.org/10.5772/61049>
79. R. Shukla, T. Tripathi, Molecular dynamics simulation in drug discovery: opportunities and challenges, in *Innovations and Implementations of Drug Discovery Strategies in Rational Drug Design*, ed. by S.K. Singh, (Springer, Singapore, 2021), pp. 295–316. https://doi.org/10.1007/978-981-15-8936-2_12
80. R. Shukla, T. Tripathi, Molecular dynamics simulation of protein and protein-ligand complexes, in *Computer-Aided Drug Design*, ed. by D.B. Singh, (Springer, Singapore, 2020), pp. 133–161. https://doi.org/10.1007/978-981-15-6815-2_7
81. K. Prince, S. Sasidharan, N. Nag, T. Tripathi, P. Saudagar, Integration of spectroscopic and computational data to analyze protein structure, function, folding, and dynamics, in *Advanced Spectroscopic Methods to Study Biomolecular Structure and Dynamics*, ed. by P. Saudagar, T. Tripathi, (Academic Press, San Diego, 2023), pp. 483–520