Chapter 2

2. Computational Approaches to Study Biomolecules

2.1 Introduction

Understanding the function of biological macromolecular systems such as proteins, nucleic acids, lipid membranes, carbohydrates, and their complexes is a major objective of current research by computational chemists and biophysicists (Sansom and Smith, 1998; Saxena et al., 2009). The behavior of such systems can be described in terms of the basic physical principles governing the interactions and motions of their elementary atomic constituents. The models are, thus, rooted in the fundamental laws of physics and chemistry, including electrostatics, quantum mechanics and statistical mechanics. The challenge now is in the development and application of methods, based on such well-established principles, to shed light on the structure and conformational flexibility, function, and properties of biological macromolecules and their complexes. Biological macromolecular systems of increasing size and complexity, including carbohydrates, nucleic acids, viruses, membrane proteins, and macromolecular assemblies, are now being investigated using these computational methods (Slater et al., 2009; Veluraja and Rao, 1980; Veluraja and Margulis, 2005).

Computational approaches are now used to facilitate the experimental determination of macromolecular structures by aiding in structural refinement based
on their X-ray and nuclear magnetic resonance (NMR). They are widely applied to determine the biomolecular structure and dynamics and it can also be used in situations where experimentally determined structures are not available.

In computational molecular simulations, the simplified description is a calculated potential energy surface which represents the molecule of interest. This energy is a function of atomic coordinates. The three major methods to simulate the molecular structures are Molecular mechanics (MM), Monte-Carlo (MC) and Molecular Dynamics (MD) simulation (Bonneau and Baker, 2001; Vasudevan and Balaji, 2001; Saiz et al., 2002; Norberg and Nilsson, 2003; Skeel et al., 2007; Senn and Thiel, 2009; Hagiwara et al., 2010; Durrant and McCammon, 2011).

2.2 Molecular Modelling

Molecular Modelling is a powerful methodology to represent, visualize and investigate the three dimensional structure of biomolecules. It is very important to specify the position of the atoms in a molecule in the system to a modeling program. Z-matrix and Cartesian (x,y,z) coordinates methods are most commonly used to represent the position of atoms in a molecule. The three dimensional structure of a molecule can be generated by knowing the internal parameter such as bond length, bond angle and torsional angle. Molecular graphics and computational chemistry are the two major classifications of molecular modeling. Molecular graphics is commonly used to visualize the chemical structures, molecular properties and manipulation in initial structure by modifying the torsional angles of chemical bonds and basic geometric calculations. The numerical properties such as energy
calculations of biomolecules can be studied through computational chemistry (Park and Khalili-Araghi, 2003; Lawrenz et al., 2011; Dabbagh et al., 2011).

### 2.3 Molecular Mechanics

Molecular mechanics is one of the computational techniques to study the molecular system and this is widely used by chemists, biophysicists and biochemists. Schleyer and Allinger are the major contributors to the development of molecular mechanics (Kollman et al., 2000). In molecular mechanics, the atoms of the molecules are treated as sphere and bonds between the particles are viewed as harmonic oscillators. Molecular mechanics expresses the total energy as a sum of Taylor series expansions for stretches for every pair of bonded atoms, and adds additional potential energy terms coming from bending, torsional energy, van der Waals energy, electrostatics, and cross terms:

\[
E = \sum_{\text{bonds}} E_{\text{stretch}} + \sum_{\text{angles}} E_{\text{bend}} + \sum_{\text{dihedrals}} E_{\text{torsion}} + \sum_{\text{pairs}} E_{\text{nonbond}} \quad \cdots \quad (2.1)
\]

Where, \( E_{\text{stretch}} \) - Energy contribution from bond stretching

\( E_{\text{bend}} \) - Energy contribution from angle bending

\( E_{\text{torsion}} \) - Energy contribution from torsional motion i.e. rotation around single Bonds

\( E_{\text{nonbond}} \) - Energy contribution from interaction between atoms or groups which are nonbonded.

Molecular mechanics (MM) models are useful in studying structures, conformational energies, and other molecular properties, including vibrational
frequencies, conformational entropies and dipole moments. MM models also play an important role in the rational molecular design (Homeyer and Gohlke, 2012).

2.4 Molecular Dynamics Simulation

The Molecular Dynamics method was first introduced by Alder and Wainwright in the late 1950's (Alder and Wainwright, 1957, 1959) to study the solid-fluid transition in a system composed of hard spheres interacting by instantaneous collisions. Many important insights concerning the behavior of simple liquids emerged from their studies. The next major advance was in 1964, when Rahman carried out the first simulation using Lennard-Jones potential to describe both attractive and repulsive interaction potential in a system consisting of 864 argon atoms (Rahman, 1964). The first MD simulation for a realistic system of liquid water was done by Rahman and Stillinger in 1974 (Stillinger and Rahman, 1974). The first protein simulations appeared in 1977 with the simulation of the bovine pancreatic trypsin inhibitor (BPTI) (McCammon, et al, 1977). Today in the literature, one routinely finds MD simulations of solvated proteins, protein-DNA complexes as well as lipid systems addressing a variety of issues including the thermodynamics of ligand binding and the folding of small proteins. The number of simulation techniques has greatly expanded; there exist now many specialized techniques for particular problems, including mixed quantum mechanical - classical simulations that are being employed to study enzymatic reactions in the context of the full protein. MD simulation techniques are widely used in experimental procedures such as X-ray crystallography and NMR structure determination.
The molecular dynamics (MD) technique is used to follow the evolution of a large number of interacting particles by numerically integrating the classical equations of motion. The first step in performing MD simulation is to calculate the net force on each particle. The net force acted on the system has been found out by using the Newton’s second law of motion,

$$\vec{F} = m \vec{a}$$ (2.2)

The most commonly used software packages for MD simulations are AMBER (Case et al., 2005; Case et al., 2006), CHARMM (MacKerell et al., 1998), NAMD (Phillips et al., 2005) and GROMOS (Scott et al., 1999; Lins and Hunenberger, 2005).

**Time and size limitations of MD simulation**

Typical MD simulations can be performed on systems containing thousands or millions of atoms and the simulation may be run for picoseconds to hundreds of nanoseconds. However, the simulation is safe when the simulation time is much longer than the relaxation time of the system we are interested in.

**2.4.1 Molecular Interactions**

The force acting on the atoms can be usually derived from the potential energy \( U(r) \), where \( N \) represents the complete set of \( 3N \) atomic coordinates. The interaction between two free, uncharged atoms implies two primary effects. First, there is a short-ranged repulsion preventing overlap. Second, there is a long-range attraction arising from weak but favourable interactions due to induced dipole effects (dispersion forces). The Lennard-Jones (LJ) potential is commonly used to model these effects,
The other primary non-bonded interaction arises from the electrostatic interaction between pairs of particles given by coulombic potential,

\[ U(r_{ij}) = 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6} \right] \quad (2.3) \]

\[ U(r_{ij}) = \frac{1}{4\pi\epsilon_{0}} \frac{q_{i}q_{j}}{r_{ij}} \quad (2.4) \]

Here, \( q_{i} \) and \( q_{j} \) are the effective charges on each particle, \( \epsilon_{0} \) is the permittivity of free space, and \( \epsilon \) is the dielectric constant of the medium. In atomistic simulations with an explicit, polar solvent model, including \( \epsilon \) is not necessary. However, for simulations with either implicit or non-polar solvent models (including the mesoscopic models common in coarse-grained simulations), \( \epsilon \) is an effective dielectric constant that includes the screening effects due to the medium (e.g. \( \epsilon = 80 \) for water). Although the magnitude of the net charge is obvious for free ions, in molecules where charges are shared via bonds the value of the effective partial charges is a vital component of the force field. Unlike the LJ interaction that decays relatively quickly, the Coulombic interaction is long ranged. While a cut-off distance (beyond which contributions are not considered) is appropriate in one dimension, the long-ranged contributions are important in two or three dimensions.

### 2.4.2 Bonded Interactions

In MD simulations, elements of chemical bonds are captured by implementing potentials to maintain bond lengths and bond angles. Although other forms are used, a
common choice for both is a harmonic potential such that the bond stretching $U(r_{ij})$ and bond bending $U(\theta_{ijk})$

$$U(r_{ij}) = \frac{1}{2} k_{ij} (r_{ij} - r_0)^2$$  \hspace{1cm} (2.5)$$

$$U(\theta_{ijk}) = \frac{1}{2} k_{ijk} (\theta_{ijk} - \theta_0)^2$$  \hspace{1cm} (2.6)$$

Here, $k_{ij}$ and $k_{ijk}$ are force constants, $\theta_{ijk}$ is the angle formed by the bonds joining atoms $i$, $j$, $k$, and $r_0$ and $\theta_0$ are the equilibrium separation and bond angle respectively. Hence, in this model, atoms are bonded together via Hookean springs while the bond angles oscillate around the equilibrium value. The energy associated with torsional angle ($\omega$) are given by

$$U(\omega) = \left( \frac{V_{n}}{2} \right) \left[ 1 + \cos(n\phi - \gamma) \right]$$  \hspace{1cm} (2.7)$$

In atomistic simulations, the various parameters are an essential part of the force field as they dictate the details of these interactions.

### 2.4.3 Time integration algorithm

The engine of a MD program is its time integration algorithm, required to integrate the equation of motion of the interacting particles and follow their trajectory (Schlick et al., 1999). Time integration algorithms are based on finite difference methods, where time is discretized on a finite grid, the time step $\Delta t$ being the distance between consecutive points on the grid.
There are two common algorithms, which are used to solve the equations of motion are Gear predictor-corrector and Verlet algorithm as both of these algorithms require only one evaluation of the forces at each time step. For computing particle trajectories using small time-steps Gear predictor-corrector method is more accurate than the Verlet algorithm (Grubmüller et al., 1991; Spreiter and Walter, 1999).

2.4.4 The Verlet algorithm

In molecular dynamics, Verlet algorithm is widely used to calculate the time integration (Spreiter and Walter, 1999). It uses the information from the current and previous time steps to advance the atomic positions. The basic idea is to write two third order Taylor expansions from t to \( t + \Delta t \) and \( t - \Delta t \). If \( v \) is the velocity, \( a \) the accelerations and \( b \) the third derivative of \( r \) with respect to \( t \), then

\[
\begin{align*}
  r(t + \Delta t) &= r(t) + v(t)\Delta t + \left( \frac{1}{2} \right) a(t)\Delta t^2 + \left( \frac{1}{6} \right) b(t)\Delta t^3 + O(\Delta t^4) \\
  r(t - \Delta t) &= r(t) - v(t)\Delta t + \left( \frac{1}{2} \right) a(t)\Delta t^2 - \left( \frac{1}{6} \right) b(t)\Delta t^3 + O(\Delta t^4)
\end{align*}
\]

By adding the above two expressions it gives,

\[
r(t + \Delta t) = 2r(t) - r(t - \Delta t) + a(t)\Delta t^2 + O(\Delta t^4)
\]

This is the basic form of the Verlet algorithm. Since we are integrating Newton’s equations, \( a(t) \) is just the force divided by the mass, and the force is in turn a function of the positions \( r(t) \),
A problem with this version of the Verlet algorithm is that velocities are not directly generated. While they are not needed for the time evolution, their knowledge is sometimes necessary. Moreover they are required to compute the kinetic energy $K$, whose evolutions is necessary to test the conservation of the total energy $E=K+V$. This is the most important test to verify that a MD simulation is proceeding correctly. One could compute the velocities from the positions by using,

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} - (2.12)$$

However the error associated with this expression is of the order of $\Delta t^2$ rather than $\Delta t^4$. To overcome this difficulty, some variants of the Verlet algorithms has been developed (Swope et al., 1982). They give rise to exactly the same trajectory, and differ in what variables are stored in memory at what times. The Velocity Verlet scheme and Leap-frog algorithm are the variants of Verlet algorithm. In Velocity Verlet algorithm the positions, velocities and accelerations at time $t+\Delta t$ are obtained from the same quantities at time $t$ in the following way:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 -(2.13)$$

$$v\left(t + \frac{\Delta t}{2}\right) = v(t) + \frac{1}{2}a(t)\Delta t - (2.14)$$

$$a(t + \Delta t) = -\frac{1}{m}\nabla V(r(t + \Delta t)) - (2.15)$$
\[ v(t + \Delta t) = v\left(t + \frac{\Delta t}{2}\right) + \left(\frac{1}{2}\right) a(t + \Delta t) \Delta t - \ldots - \ldots - \ldots - \ldots - \ldots - (2.16) \]

Verlet algorithms are simpler to use and require less memory than the Gear predictor-corrector method (Allen and Tildesley, 1987)

Gear predictor-corrector method follows two steps and are (i) predictor (ii) corrector

Predictor step provides an initial approximation to the propagated solution and corrector yields a refined approximation.

2.5 Force field

The interactions between the particles in the system play an important role. The mathematical forms and parameters dictating these interactions and are called as “force field”. It is used in MD simulation to calculate the internal potential energy of a molecule. Consider a molecule as a collection of atoms held together by elastic forces. Now the forces can be written in terms of potential energy functions of various structural features such as bond lengths, bond angle, non bonded interactions etc. The force field is the combination of these potential energy terms. Hence force fields are also sometimes referred to as potentials. Thus the energy, \( E \), of a molecule in a force field arises from the deviations from the ideal structural features and so can be written

\[ E = E_s + E_b + E_{\omega} + E_{nb} - \ldots - \ldots - \ldots - \ldots - (2.17) \]

Here \( E \) is termed as the steric energy. \( E_s \) is the energy for bond stretching; \( E_b \) is the energy for bond angle bending, \( E_{\omega} \) is the torsional energy due to twisting about bonds and \( E_{nb} \) is the energy for non bonded interactions.
2.5.1. Force field parameter

Energy arises due to stretching and bending are calculated by applying Hookes law and is,

\[ E_s = \sum_{i}^{N} \frac{k_s}{2} (l_i - l_i^0)^2 \quad (2.18) \]

\[ E_b = \sum_{i<j}^{M} \frac{k_b}{2} (\theta_{ij} - \theta_{ij}^0)^2 \quad (2.19) \]

Where, \( N \) is the total number of bonds and \( M \) is the total number of bonds angles in the molecule. \( k_s \) and \( k_b \) are the force constants for stretching and bending respectively. \( l_i \) and \( \theta_{ij} \) are the actual bond length and bond angles. \( l_i^0 \) and \( \theta_{ij}^0 \) are ideal bond lengths and bond angles.

The energy due to torsion is usually expressed in terms of Fourier series,

\[ E_w = \sum \frac{1}{2} \left[ V_1 (1 + \cos \omega) + V_2 (1 + \cos 2\omega) + V_3 (1 + \cos 3\omega) + \cdots \right] \quad (2.20) \]

Where, the sum is over all unique sequences of bonded atoms. In general the series is truncated at the third term, \( V_1 \), \( V_2 \) and \( V_3 \) being chosen so that the resultant conformation agree well with experiment for a given group of molecules.

The pairwise non bonded interactions are given by Lennard Jones potential and Coulombic potential energy.

\[ E_{\text{nonbond}} = \frac{A}{r^{12}} - \frac{B}{r^6} + \frac{q_i q_j}{r} \quad (2.21) \]
2.6 Boundary condition

When simulating a system of finite size, some thought must be given to the way the boundary of the system will be treated. The simplest choice is the vacuum boundary condition. When simulating a liquid, solution or solid rather than a molecule in the gas phase, it is common practice to minimize edge or wall effects by the application of periodic boundary conditions. If the irregularity of the system is incompatible with periodicity, edge or wall effects may be reduced by treating part of the system as an extended wall region in which the motion of the atoms is partially restricted.

The edge effect in the finite system can be minimized by using the periodic boundary conditions. The atoms of the system that is to be simulated are put into a cubic or more generally into any periodically space-filling shaped box which is treated as if it is surrounded by $26(3^3-1^3)$ identical translated images itself. The next layer of neighbor images of the central computational box contains $5^3-3^3-1^3 = 98$ boxes, and so on.

2.7 Strategy involved performing the MD simulation

A typical MD simulation consists of four stages and they are initialization, equilibration, production and analysis.

Initialization

To run the MD simulation of any system, it is necessary to initialize the coordinates of the atoms, their velocities and the target temperature for the simulation. The initial velocities are assigned with random directions and a fixed magnitude.
**Equilibration**

After the initialization of the MD simulation, it should take some period of time for the system to achieve equilibrium before collecting data. This equilibration involves the achievement of the correct partitioning of energy between kinetic and potential energy, as well as attaining a Maxwell-Boltzmann velocity distribution corresponding to the concerned temperature. The time needed for equilibration to occur is variable and it depends on the nature and size of the system being run. During this stage, the velocities are normally scaled to maintain a proper temperature. Equilibration of the system can be accelerated by first starting the simulation at a higher temperature and later cooling by rescaling the velocity.

**Production**

After the MD system reaches to the equilibrium state, it usually takes another period of time to collect data. In this production stage after equilibration, no velocity scaling for temperature control is involved, while the trajectories are written out in some interval to the external file for later analysis. The number of time steps in this production stage usually depends on the nature of the problem and the purpose of the simulation.

**Analysis**

This stage involves analyzing the information stored in the trajectory file during the production stage. The trajectory file contains the absolute Cartesian coordinates, velocities and various observables of the system such as energy, temperature, pressure etc. Depending on the different purpose of the simulation, the
trajectory information can be extracted and employed to analyze either the material properties or physical characteristics.

The trajectory files can be visualized through various visualization softwares. In this work VMD is used to view the trajectory files.

2.8 Applications of MD simulation

Computer simulation is widely used at the atomic level. MD and MC methods have found widespread use in structure determination, refinement and prediction (Gunsteren and Berendsen, 1990; Patel et al., 2008; Kang et al., 2012). MD simulation is used to study the protein folding, protein stability, conformational changes, conformation of carbohydrates, protein-carbohydrate interactions and ion transport in biological systems (Veluraja and Rao, 1980; Veluraja and Margulis, 2005; Buch et al., 2010; Veluraja et al., 2010; Wang and Hou, 2011). An especially elegant and important use of molecular dynamics simulations is in the study of modified ligands and mutant proteins. Also MD simulation is used in drug designing (Alonso et al., 2006; Salsbury Jr, 2010; Borhani and Sahw, 2011; Harvey and De Fabritiis, 2012).

2.9 Quantum Mechanical Calculation

Quantum mechanical methods are widely used to study the molecular structure and reactions of molecules with the help of statistical and quantum mechanics. In this method, the energy and properties of a molecule are obtained by solving the Schrodinger equation. The field of QM–MM simulations of chemical reactions has
grown considerably from the initial proposals of Warshel and Levitt in the 1970s (Warshel and Levitt, 1976) to a technique that can now deliver quantitatively accurate reaction pathways for reactions in the active sites of enzymes (Giese and York, 2004; Senn and Thiel, 2009; Bartok and Payne, 2010; Ob-Egbedi et al., 2011; Panchenko et al., 2011).

2.9.1 Gaussian

Ab initio calculations (ab initio is a latin word gives the meaning of “from the start”, i.e. from the first principles) are based on the Schrodinger equation. This is one of the fundamental equations of modern physics and describe about how the electrons are behave in a molecule. The ab initio method solves the Schrodinger equation for a molecule and gives us an energy and wave function. The wave function is a mathematical function that can be used to calculate the electron distribution.

The wave function and energy of the system can be calculated by approximation methods (Bartok and Payne, 2010). Hartree-Fock (HF) method is the most commonly used approximation method (Slater, 1930), in which all electrons are assumed to move independently of each other, thus dividing the problem into a group of independent one-electron sub-problems. The Moller–Plesset (MP) perturbation theory (Moller and Plesset 1934) is commonly used to correct the errors introduced by the HF simplification. Perturbation theory is a mathematical way to solve a complex mathematical problem by first solving a simple related problem (HF approximation) and then iteratively (the number of steps is the order of the model) adding correction terms (electron-electron interaction terms) to solve the original problem.
The second-order model (MP2) is widely used, but third- (MP3) and fourth-order (MP4) models can also be used when higher precision is desired.

Density functional theory (DFT) is an alternative theory where the energy is estimated using functionals (i.e. functions of functions) of the electron density instead of the wave functions. This makes the DFT approach computationally fast. Because of difficulties in modeling some intermolecular interactions accurately, hybrid approaches that borrow some terms from HF or MP theories have been developed. The B3LYP (Becke, three-parameter, Francesco Strino 19 Lee–Yang–Parr) theory (Becke 1993; Kim and Jordan 1994; Lee et al., 1988) is one of the most commonly used DFT approaches. In Hartree-Fock method, the N-body wave function of the system is approximated by a single slater determinant. Gaussian is a widely used software package to perform the QM calculations and is initially released by John Pople in 1970 as Gaussian 70. Gaussian 09 is the most recent release of Gaussian.

To calculate the molecular geometries, rates and equilibria, spectra and other physical properties, one can use molecular mechanics, ab initio, semiempirical and density functional methods and molecular dynamics. Computational chemistry is widely used in the pharmaceutical industry to explore the interactions of potential drugs with biomolecules, for example by docking a candidate drug into the active site of an enzyme. Using computational chemistry we can calculate the molecular geometry such as shape of the molecule includes bond length, bond angles and torsional angles, energy of the molecules and transition states and chemical reactivity (Giese and York, 2004; Subramanian et al., 2010). Gaussian is used to calculate single point energy, geometry optimization, frequency and thermodynamical analysis, reaction path following (Yamaguchi et al., 2012).
2.9.2 Basis sets in Gaussian

The complete basis functions are used to represent the molecular orbitals. In most cases, the basis set simply consists of the relevant exponents and coefficients of Gaussian functions. The smallest basis set is called as Minimal Basis (MB) set. This basis set has one basis function per occupied atomic orbital on the atom. The most common minimal basis set is STO-3G (Hehre et al., 1969; Collins et al., 1976) which means that three contracted Gaussians are used to replace one Slater type orbital. The problem with the minimal basis description is that it doesn’t allow the atoms to change their shape very much. If H has only one function, then it does not have any degrees of freedom to adjust to adapt to a different bonding situation.

The difficulty of the single basis function in minimal basis set has been improved with the additional basis functions per occupied atomic orbital and is called as double zeta (DZ) basis sets. However valance orbitals are involving in chemical bonding hence it is necessary to add the additional functions for the valence orbitals and not for the core orbitals. In practice most DZ basis set are called as valence double zeta (VDZ) basis set. The VDZ function of H has two basis functions (1s and 2s). The most commonly used VDZ basis set in gaussian are 3-21G (Binkley, et al., 1980; Gordon, et al., 1982; Dobbs and Hehre, 1987) and 6-31G (Ditchfield, et al., 1971; Hehre, et al., 1972; Hariharan, et al., 1974; Gordon, 1980).

In the presence of an external field, molecules exhibit polarizations. But VDZ basis set does not include the polarization of the molecules. So we add polarization functions, which are functions with one higher angular momentum than the highest occupied basis function and this description is called as double zeta polarization
(DZP) basis set. 6-31G*, 6-31G** cc-pVDZ are belongs to DZP basis set. Here * indicates that polarization functions are included on the heavy atoms except H.

2.10 Softwares used in this study

AMBER (Assisted Model Building by Energy Refinement)

AMBER is the collection of set of programs that allows the user to perform MD simulation particularly on biomolecules (Case et al., 2006; Perez et al., 2007). AMBER uses many force fields which includes (i) non-polarizable (ff99) and polarizable (ff02) protein force fields with improved torsional parameters for peptides and proteins. (ii) all-atom force field (ff03) (iii) an extension of general amber force field (gaff). The preparatory programs in AMBER are LEap and ANTECHAMBER. Of which, LEap is used to create a new system in Amber, or to modify old systems. ANTECHAMBER is the main program which is used if the system contains more than standard nucleic acids and proteins. This is used to prepare the input files for LEap.

SANDER is the basic energy minimization and molecular dynamics program. PTRAJ and MM-PBSA are the analysis programs in AMEBR. The collected trajectories are analyzed using these programs.

CHARMM (Chemistry at HARvard Molecular Mechanics) is a highly versatile and widely used molecular simulation program (Brooks et al., 2009). In this study AMBER9 is used for complex dynamics.
NAMD

NAMD is a Nano scalable Molecular Dynamics (Phillips et al., 2005) and has been used to perform the MD simulations. NAMD is a parallel, object-oriented molecular dynamics program and is designed for high performance simulation of large biomolecular system (Kale et al., 1999). In this project work the dynamical behavior of sialic acid and its acetylated derivatives are studied by performing the 10ns MD simulation using NAMD.

VMD

Visualization Molecular Dynamics (VMD) package is used to analyze the collected trajectories from MD simulations (Humphrey et al., 1996). VMD has been developed by the Theoretical and Computational Biophysics group at the Beckman Institute for Advanced Science and Technology of the University of Illinois at Urbana-Champaign. The initial version of original VMD is called as VRChem, and was developed in 1992 by Mike Krogh, Bill Humphrey, and Rick Kufrin (Nelson et al., 1995; Nelson et al., 1996). The first full featured VMD was developed in1998 by John Stone. The first version of VMD for the platform was released in 1999. In 2001, Justin Gullingsrud, Paul Grayson, and John Stone added support for haptic feedback devices and further developed the interface between VMD and for performing interactive molecular dynamics simulations (Stone et al., 2001). In subsequent developments, Jordi Cohen, Gullingsrud, and Stone entirely rewrote the graphical user interfaces, added built-in support for display and processing of volumetric data and the first use of openGL shading language. VMD is primarily developed as a tool for viewing and analyzing the results of molecular dynamics simulations, but it also includes tools for working with volumetric data, sequence data, and arbitrary graphics.
objects. Molecular scenes can be exported to external rendering tools such as POV – Ray, Renderman, Tachyon, VRML and many others. Users can run their own TCL and Python scripts within VMD as it includes embedded Tcl and Python interpreters (Dalke and Schulten, 1997). VMD is available free of charge, and includes source code, but it is under a free software license.

**Gaussian 03**

Gaussian is a package developed primarily for electronic structure calculations, although it will perform some molecular mechanics and molecular dynamics calculations. In this project, Gaussian 03 package is used to perform molecular geometry optimization using the basis set HF/6-31G*.

**MOLSCRIPT**

The MOLSCRIPT (Kraulis, 1991) program produces plots of protein and no-protein structures using several kinds of representation. MOLSCRIPT program uses, the three dimensional coordinate file as a input file.