CHAPTER 2

COMPUTATIONAL METHODS USED FOR MOLECULAR MODELLING AND MOLECULAR DYNAMICS SIMULATIONS

2.1 Introduction

In the present day scenario, computational methods have a pivotal role in studying the conformations of carbohydrates (glycans) and protein–carbohydrate complexes. Glycans are having various conformations in solution state because of the higher flexibility about glycosidic linkages and the internal fluctuations. Because of these flexibilities, the experimental methods are inadequate to examine the multiple conformations observed in the solution state. These conformations are dynamic and change rapidly on the timescales of nanoseconds (ns) or microseconds (μs). Molecular Dynamics (MD) and Monte Carlo (MC) simulations can predict these conformational states and determine their relative probability of occurrence in the solution state. Molecular modelling techniques can also be applied to examine the carbohydrate conformations when they are bound to glycan binding proteins (GBP). The modelling studies can complement the data obtained from the experimental techniques like X-ray diffraction and NMR spectroscopy. However, experimental validation is the key to success the modelling studies. The relative binding energies of the ligand can be predicted by molecular modelling studies which can aid the selection and development of potential antagonists against diseases. The binding free energy of the glycan bound complexes can also be determined and the stability of complexes can be predicted using theoretical calculations. These computational
techniques have wide variety of applications such as structure based drug
design, rational drug design, designing cyborg lectins, engineering neolectins,
protein-protein, protein-carbohydrate and protein-DNA interactions, biological
recognition processes and the theoretical structure predictions.

2.2 Molecular Mechanics calculations

The aim of molecular mechanics (MM) calculations is to predict the
structural details and physical properties of the molecules of interest. Strain
energy, enthalpies of formation, entropies and the dipole moments are the
notable physical properties that can be calculated using molecular mechanics.
Based on the classical mechanics, MM is used to model the molecular and
supramolecular systems and to calculate their energies. MM is also used to
investigate the small molecules and larger assemblies of biomolecules with
thousands of atoms. To obtain the molecular structure in minimum energy
conformation, MM adjusts the energy through changes in bond lengths and bond
angles. The potential energy of all the systems in molecular mechanics is
calculated using force fields. The total energy of the system can be obtained by
calculating the energy due to bonded interactions, non-bonded interactions and
the internal forces as described in the Equations 2.1 - 2.3.

\[ E_{\text{total}} = E_{\text{bonded}} + E_{\text{non-bonded}} \]  
\[ \text{............... (2.1)} \]

where,

\[ E_{\text{bonded}} = E_{\text{bond}} + E_{\text{angle}} + E_{\text{torsion}} \]  
\[ \text{......... (2.2)} \]

\[ E_{\text{non-bonded}} = E_{\text{electrostatic}} + E_{\text{van der Waals}} \]  
\[ \text{........ (2.3)} \]

The bonded interactions are the summation of energy from stretching,
bending and torsions while the non-bonded interactions are the summation of electrostatic and van der Waals interactions. MM calculations are extensively used in understanding the biomolecular simulations in combination with other computational approaches like Quantum Mechanical (QM) calculations. Various softwares are used for molecular mechanics calculations such as AMBER (Assisted Model Building with Energy Refinement), DESMOND (To perform high-performance molecular dynamics simulations for biomolecular systems), GROMACS (GROningen MAchine for Chemical Simulations), CHARMM (Chemistry at HARvard Macromolecular Mechanics) and Discovery Studio (To simulate small and macromolecular systems) (Brooks et al, 1983; Case et al, 2012; Poltev, 2012).

2.3 Monte Carlo Simulations

Monte Carlo (MC) simulations are computational algorithms based on the repeated random sampling method for the numerical results, quantitative analysis and decision making. Optimization, numerical integration and the generation of draws from a probability distribution are the three distinct problem classes available in the Monte Carlo simulations. The application of Monte Carlo simulation to the conformations of carbohydrates is the less common when compared to MD simulations. Generation of conformer ensembles is common in both the MC and MD simulations. The new conformers are generated randomly in MC simulations while in MD simulations, the conformers are generated from the previous state of the conformers (Rao et al., 1998). The Metropolis algorithm is used in the most common implemented version of MC simulations, which ensures that the generated conformations obey the Boltzmann probability
distribution (Metropolis et al., 1953). The strain free conformation in which the conformers are free from close contacts between non-bonded atoms and free from bond angle deformations is used as initial conformation in the MC simulations. From this initial conformation, a new conformation is generated by a random change in the atomic coordinates. The energy difference between the first and second conformation is also considered in MC simulations. In addition to the above, the simulations are carried out either in the internal coordinate or the Cartesian coordinate space by changing either torsion angles or orthogonal coordinates x, y and z (Rao et al., 1998). Based on the deterministic approach, Monte Carlo methods can also be used in the understanding of biomolecular kinetics with both spatial and temporal resolutions (Caflisch, 1998).

2.4 Molecular Dynamics simulations

Molecular Dynamics (MD) simulations are important tools widely used to understand the structure, function and dynamics of biological macromolecules. MD simulation studies are extended to investigate the conformations of carbohydrates and their complexes (McCormon et al., 1977; Karplus and McCammon, 2002; Veluraja and Margulis, 2005; Fadda and Woods, 2010; Veluraja et al., 2010; Priyadarzini et al., 2012). In MD simulations, the dynamical behavior of the molecules is determined by the forces acting on the individual atoms. In a different way, the MD is defined as the time dependent behavior of the molecular systems is computed in this principal theoretical study. Newton’s second law or the equation of motion (Newtonian mechanics) is used to simulate the molecules over a period of time (ps, ns and µs). The Newton’s second law of motion is the following:
\[ m_i \frac{d^2 r_i}{dt^2} = F_i(t) \quad i = 1,2,3,\ldots,n \quad \text{............ (2.4)} \]

Where \( m \) is the mass and \( t \) is the time, \( r \) is the positional vector of the atom, \( F \) is the force and \( n \) is the total number of atoms. The total potential energy of a system \( (V_{\text{tot}}) \) is a function of the positional vectors \( (r) \) of the constituent atoms. Hence the force \( (F) \) can easily be calculated, because the force is negative gradient of the potential can be referred to the derivative of the potential energy function with respect to the atomic coordinates. To obtain a new set of positions and velocities, equation 2.4 is numerically integrated over a small period of time (femtosecond) with a given set of initial positions and velocities. The initial set of positions are generally obtained either from X-ray crystallography or generated from the standard geometry (bond length, bond angle and torsion angle). The velocities are assigned from a Maxwellian distribution corresponding to a specific temperature. This procedure is iterated to get a trajectory of the motion of the molecules (Rao et al., 1998).

MD simulations are used to determine the three dimensional structure of macromolecules and refine the data obtained from experimental techniques like X-ray diffraction and NMR spectroscopy (Bruenger and Karplus, 1991; Torda and Gunsteren, 1991; Rao et al., 1998). The simulation results provide a dynamic picture of the molecular system instead of the static picture from the energy minimization studies. The energy minimization step is also available in the simulations as a first step to avoid the system of bad or close interatomic contacts (steric clashes) and to reduce the bond deformations and angle bending strain. The free energy changes and the equilibrium thermodynamic properties in the molecular systems can be determined from the ensemble of conformations or
simply trajectories generated by the simulations. Periodic boundary condition (PBC) is a significant feature in MD simulations of which the molecular system functions as it is part of a much larger system yet its properties represent the behavior at the larger scale. The atoms are placed in a cubic box or any finite space which could periodically fill the infinite volume. PBC can be used in the computer simulations and mathematical models mostly used to avoid the edge effects.

2.4.1 Time integration algorithm in molecular dynamics simulations

The time integration algorithm is an engine of a molecular dynamics program required to integrate the equation of motion of the interacting particles and the simulation trajectories. Various numerical integration algorithms are developed for integrating the equations of motion to the advantages like accuracy, stability, speed and computing time economy. Some of the notable integration algorithms are a) the basic Verlet algorithm, b) Verlet Leap-frog algorithm and c) Velocity Verlet algorithm. To select the best algorithm, the significant criteria such as conservation of energy and momentum, computationally efficient and the permission of long time step for integrations are considered. All the integration algorithms assume the positions, velocities and accelerations can be approximated by a Taylor series expansion:

\[
\begin{align*}
\mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v}(t) \Delta t + \frac{1}{2} \mathbf{a}(t) \Delta t^2 + \ldots \\
\mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \mathbf{a}(t) \Delta t + \frac{1}{2} \mathbf{b}(t) \Delta t^2 + \ldots \\
\mathbf{a}(t + \Delta t) &= \mathbf{a}(t) + \mathbf{b}(t) \Delta t + \ldots
\end{align*}
\]

\[\ldots \ldots \text{(2.5)}\]

where \( \mathbf{r} \) is the position, \( \mathbf{v} \) is the velocity and \( \mathbf{a} \) is the acceleration.
a) The basic Verlet algorithm

The Verlet algorithm uses positions and accelerations at time $t$ and the positions from time $t - \Delta t$ to calculate new positions at time $t + \Delta t$. The Verlet algorithm uses no explicit velocities. Its straightforwardness and modest storage requirements are the major advantages of Verlet algorithm while, the disadvantage is that the algorithm is of moderate precision. To derive the basic Verlet algorithm, consider the following two equations,

$$ r (t + \Delta t) = r(t) + v(t) \Delta t + \frac{1}{2} a(t) \Delta t^2 + ... $$ ..................................(2.6)

$$ r (t - \Delta t) = r(t) - v(t) \Delta t + \frac{1}{2} a(t) \Delta t^2 + ... $$ ..................................(2.7)

and then summing the equations 2.6 and 2.7, gives

$$ r (t + \Delta t) = 2 r(t) - r(t - \Delta t) + a(t) \Delta t^2 $$ ..................................(2.8)

Equation 2.8 is the basic form of Verlet algorithm.

b) Leap-frog Verlet algorithm

In Leapfrog Verlet algorithm, the velocities are first calculated at time $t + \frac{1}{2} \Delta t$; which are used to determine the positions, $r$, at time $t + \Delta t$. In this way, the velocities leap over the positions and the positions leap over the velocities. The advantage of this algorithm is that the velocities are explicitly calculated, whereas the disadvantage is that they are not calculated at the same time as the positions. The velocities at time $t$ can be approximated by the relationship,

$$ \begin{align*}
    r (t + \Delta t) &= r(t) + v(t + \frac{1}{2} \Delta t) \Delta t \\
    v (t + \frac{1}{2} \Delta t) &= v(t - \frac{1}{2} \Delta t) + a(t) \Delta t \\
\end{align*} $$ ..................................(2.9)

And velocities at time $t$ can be calculated by approximating the relationship,

$$ v (t) = \frac{1}{2} [v(t - \frac{1}{2} \Delta t) + v(t + \frac{1}{2} \Delta t)] $$ ..................................(2.10)

Equation 2.10 is the functional form Leapfrog Verlet algorithm.
c) Velocity Verlet algorithm

The velocity Verlet algorithm provides both the atomic positions and velocities at the same instant of time, and for this reason may be regarded as the most complete form of Verlet algorithm. The basic equations are as follows.

\[
\begin{align*}
\mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + v(t) + \frac{1}{2} a(t) \Delta t^2 \\
v(t + \Delta t) &= v(t) + \frac{1}{2} [a(t) + a(t + \Delta t)] \Delta t
\end{align*}
\]

Equation 2.11 is the functional form of velocity Verlet algorithm.

2.4.2 The potential and force field used in molecular dynamics simulations

The main ingredient of molecular dynamics simulations is a model for the physical system. Also the ability of a simulation to yield specific information about a system is highly dependent on the quality of the potential energy function that is used to describe the interactions between the smallest entities in the simulation. The mathematical formula for calculating the forces is always based on the potential energy functions. One of the most commonly used empirical potential is the Lennard-Jones potential and is given by the expression,

\[
u(r_{ij}) = 4\varepsilon \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6}
\]

\[
\text{......................... (2.12)}
\]

A collection of parameters defining the spheres and interaction potentials is referred to as force field. The force field allows calculating the potential energy of the system and is generally based on a pair-wise sum over all the different sphere pairs present in the system. For example, the AMBER force field is based on the following parameters:
\[ V = V_{\text{bonded}} + V_{\text{non-bonded}} \]  \hspace{1cm} \text{(2.13)}

\[ V_{\text{bonded}} = V_{\text{bonds}} + V_{\text{angles}} + V_{\text{dihedrals}} \]  \hspace{1cm} \text{(2.14)}

\[ V_{\text{non-bonded}} = V_{\text{electrostatic}} + V_{\text{van der Waals}} \]  \hspace{1cm} \text{(2.15)}

The substitutions of equations 2.14 and 2.15 into equation 2.13, gives the following expression,

\[ V = \sum_{\text{bonds}} k_{r_0} (r - r_0)^2 + \sum_{\text{angles}} k_{\theta_0} (\theta - \theta_0)^2 + \sum_{\text{dihedrals}} [1 + \cos(n\phi - \pi)] 
+ \sum_{\text{non-bonded pairs}} \left[ \frac{A}{r_{ij}^6} - \frac{B}{r_{ij}^4} + \frac{C}{r_{ij}^6} \right] \]  \hspace{1cm} \text{(2.16)}

The basic functional form of a force field encapsulates both bonded terms relating to atoms that are linked by covalent bonds and non-bonded terms describing the long-range electrostatic and van der Waals forces which are shown in Figure 2.1.

**Figure 2.1.** The basic terms included in the functional form of a potential. Covalent bonds are indicated by solid lines and non-bonded interactions by a dashed line.
The bond and angle terms are usually modeled as harmonic oscillators in force fields that do not allow bond breaking. Morse potential gives a more realistic description of a covalent bond at higher stretching. The functional form for the rest of the bonded terms is highly variable. Proper dihedral potentials are usually included. Additionally, the improper torsional may be added to enforce the planarity of aromatic rings and other conjugated systems and the cross-terms that describe coupling of different internal variables such as angles and bond lengths. Some force fields also include explicit terms for hydrogen bonds. The non-bonded terms are most computationally intensive because they include many more interactions per atom. A widely used solution to this problem is to limit interactions to pair-wise energies (Paton and Goodman, 2009). The van der Waals term is usually computed with a Lennard-Jones potential and the electrostatic term with Coulomb's law. This accounts for the greater degree of agreement to the experimental observations. Particle Mesh Ewald (PME) is a computational method widely used for determining the forces in a system of particles especially computing the long-range interactions in the periodic systems. In this present work, PME is used to calculate the long range electrostatic interactions (Darden et al., 1993).

2.5 A historical perspectives on molecular dynamics simulations

Alder and Wainwright introduced the first molecular dynamics simulations to understand the interactions of hard spheres in the late 1950's (Alder and Wainwright, 1957; Alder and Wainwright, 1959). Then the predominant contribution was made by Rahman who has carried out the simulations on liquid argon using a realistic potential followed by the simulations of a realistic system
on liquid water in 1974 (Rahman, 1964; Stillinger and Rahman, 1974). When compared to the simulations of liquid argon (Ar), water simulations are a greater challenge since the inclusions of van der Waals interactions, Coulomb force and the hydrogen bonds. McCammon and Karplus have studied the dynamics of biological macromolecule, folded globular protein of bovine pancreatic trypsin inhibitor (BPTI) by solving the equations of motion for the atoms with an empirical potential energy function in 1977 (McCammon et al., 1977). Later Rasmussen and coworkers have introduced the concept of energy minimization (Melberg and Rasmussen, 1979). Then the Hard Sphere Energy calculations gained popularity to calculate the three dimensional structure of carbohydrates in the 1980s. The effect of solvent and temperature has been considered on the protein structure and dynamics (Brünger et al., 1985; Frauenfelder et al., 1987). Since then MD simulation studies are extensively used in the investigations of carbohydrate conformations and their complexes with other biomolecules (Paulson, 1989; Pérez, 1993; Rao et al., 1998; Woods, 1998; Karplus and McCammon, 2002; Veluraja and Margulis, 2005; Lütteke, 2009; Veluraja et al., 2010; Priyadarzini et al., 2012). The technological advancements of modern computers have extended the simulation time period to a range from 100 ns to µs to study the biological phenomena (Karplus, 2003; Kubitzki and De Groot, 2007; Koshy et al., 2010; Park et al., 2011). MD simulation techniques are widely used to refine and determine the three-dimensional structures of proteins and other macromolecules based on the experimental constraints from X-ray crystallography and NMR spectroscopy. Simulations can be used to investigate the individual particle motions as a function of time and in the calculation of free energy differences (Simonson et al., 2002; Karplus and McCammon, 2002).
2.6 Applications of molecular dynamics simulations

Molecular dynamics simulations have extensively used to investigate the dynamical behavior of biomolecules such as proteins, nucleic acids, carbohydrates and their complexes in full atomic details. To determine or refine the structures obtained from X-ray crystallography or NMR spectroscopy, simulated annealing protocols of MD simulations can be used. MD is also used to obtain a description of the system at equilibrium, including structural and dynamical behavior of the biological systems. Point defects like vacancies, dislocations, interstitials and grain faults in the crystal systems can be studied using MD. Simulations are also used to understand the mechanical, physical and chemical properties of cluster of atoms of any size. MD is also extensively involved in protein folding, protein-carbohydrate interactions, structure based drug design and rational drug design. Conformational analysis of carbohydrates and protein–carbohydrate complexes, ligand-binding specificity, binding free energy of the complexes and the water molecules involved in mediating hydrogen bonds have been examined using MD simulations.

2.7 Software and the tools used in the present study

Good numbers of molecular dynamics simulations softwares are available. A few simulation packages available for biophysical research and simulations are AMBER (Assisted Model Building with Energy Refinement), CHARMM (Chemistry at HARvard Macromolecular Mechanics), GROMACS (GROningen MAchine for Chemical Simulations), GROMOS (GROningen MOlecular Simulation), DESMOND (to perform high-performance molecular dynamics simulations for biomolecular systems) and NAMD (NAnoscale Molecular
Dynamics) (Brooks et al., 1983; Berendsen et al., 1995; Lindahl et al., 2001; Phillips et al., 2005; van der Spoel et al., 2005; Hess et al., 2008; Case et al., 2012). Different softwares have different features and their own merits.

2.7.1 AMBER

AMBER is acronym as Assisted Model Building with Energy Refinement, which is a software package of computer programs used to study the molecular mechanics, molecular dynamics, binding free energy calculations and the normal mode analysis. It can also be used to simulate the structural and energetic properties of biomolecules. AMBER package contains a set of molecular mechanical force fields for the simulation of biomolecules and a package of molecular simulation programs that include source code and the demos. AMBER14 is the current version which is released on April 15, 2014. The force fields available in AMBER suite are, a) non-polarizable (ff99) and polarizable (ff02, ff03) protein force fields with improved torsional parameters for peptides and proteins, (b) a new united-atom (no non-polar hydrogens) force fields derived with a philosophy similar to the ff03 all-atom force fields, (c) an extension of General Amber Force Field (GAFF) that expands the range of applicable molecules, particularly for conjugated systems, (d) support for the AMOEBA polarizable potentials and (e) an empirical valence bond model that can be used to construct approximate potentials for chemical reactions. AMBER is developed by P. Kollman's group at the University of California, San Francisco in collaboration with D. A. Case at Rutgers University and co-workers from various varsities across the overseas. AMBER with GAFF yields reasonable results in the case of protein-carbohydrate complex simulations (Case et al., 2012).
2.7.2 NAnoscale Molecular Dynamics (NAMD)

NAnoscale Molecular Dynamics (NAMD) is a simulation package designed and developed for high-performance computer simulations of large biomolecular systems such as proteins, carbohydrates and nucleic acids. NAMD is developed by the Theoretical and Computational Biophysics Group (TCBG), Beckman Institute, University of Illinois at Urbana-Champaign in collaboration with Parallel Programming Laboratory (PPL) based on the Charm++ parallel programming models. NAMD has the significant features that include CHARMM19 and CHARMM22 parameter support, NVE ensemble dynamics [moles (N), volume (V) and energy (E) ensemble], velocity rescaling, Langevin dynamics, harmonic atom restraints and energy minimization. NAMD is file-compatible with AMBER, CHARMM and X-PLOR and is distributed free of charge with the source code (Phillips et al., 2005). Preparation and analysis of the trajectories obtained from NAMD is integrated into the visualization package VMD–Visual Molecular Dynamics (Humphrey et al., 1996). Nelson and co-workers have introduced the parallel molecular dynamics code enabling interactive molecular simulations by linking to the visualization code VMD in 1995 (Nelson, 1995a). NAMD is having many features and scaling to thousands of processors and the latest stable version is 2.10 as of December 2014. Distributed Parallel Multipole Tree Algorithm (DPMTA) is the algorithm which is incorporated into NAMD (Rankin and Board, 1995). DPMTA provides an O(N) means to calculate full electrostatics interactions (N – number of atoms in the simulations). To perform the interactive MD simulations, MDScope is also incorporated into NAMD, which combines the computational power of NAMD with the molecular graphics software VMD using the communication package MDComm (Nelson et al, 1995a).
2.7.3 Visual molecular dynamics (VMD)

Visual molecular dynamics (VMD) is the molecular visualization software for displaying, modelling, animating and analyzing the larger assemblies of biomolecules such as proteins, carbohydrates, protein–carbohydrate complexes, nucleic acids, lipids and the membranes. VMD runs on UNIX workstations, Apple MacOS X and Microsoft Windows, and distributed free of charge with the source code. VMD can read standard Protein Data Bank (PDB) files and provides a wide variety of methods for rendering and coloring a molecules such as simple points and lines, CPK spheres and cylinders, licorice bonds, backbone tubes and ribbons and cartoon drawings. VMD can load the atomic coordinate trajectories obtained from the software packages that include AMBER, CHARMM, DL_POLY, GROMACS, MMTK, NAMD and X-PLOR and few other molecular simulation packages. VMD is widely used to animate and analyze the simulation trajectories. VMD is written in C++ and maintained by the Theoretical and Computational Biophysics group at the University of Illinois at Urbana-Champaign under a non-free license (Humphrey et al., 1996).

2.7.4 UCSF Chimera

UCSF Chimera (University of California San Francisco - Chimera or simply Chimera) is a highly extensible program for interactive visualization and analysis of molecular structures. It is developed by the Resource for Biocomputing, Visualization and Informatics (RBVI) at the University of California, San Francisco and can be downloaded free of charge. UCSF Chimera can also be used to generate the density maps, supramolecular assemblies, sequence alignments, docking results, trajectories from the molecular dynamics simulations and the
conformational ensembles. High-quality images, movies and animations can be generated using this chimera package. In this present study, this package is used to visualize the hydrogen bonding interactions of lectin-carbohydrate complexes in a typical frame of reference with visible manner. Then MolScript is used further to draw the atomic level interactions in the selected frame of reference either bond or ball and stick representations. UCSF Chimera is available for Windows, Linux, Apple Mac OS X and SGI IRIX (Pettersen et al., 2004).

2.7.5 MolScript and Raster3D

MolScript is a program, which is used to create the molecular images in a detailed and schematic representation from the molecular three dimensional coordinates especially from the structure of proteins. MolScript is developed by Per J. Kraulis (Kraulis, 1991). Input file for MolScript is in the form of script, which specifies the coordinate file, objects to render and the exact appearance of the objects through the graphics state parameters. Different types of representations can be generated using MolScript such as simple wire models, ball and stick models, CPK models and the schematic drawings. MolAuto is a supportive program, which produces a useful first-approximation input file generated from the PDB file. Raster3D is a set of tools or raster graphics program for generating high resolution images of proteins and to render the pictures composed in other programs such as MolScript, without the need of high powered graphics card (Merritt and Bacon, 1997).
2.8 CUPSAT: Web tool for mutant stability analysis

CUPSAT is an acronym for Cologne University Protein Stability Analysis Tool, which is the web tool used to predict and analyze the protein stability changes upon mutations. Mutant stability prediction is performed either from existing PDB structures or custom protein structures. PDB ID, amino acid residue number, wild-type amino acid (native), experimental method (thermal or denaturants) and the chain ID are required as input for CUPSAT run. If the PDB structures are not having chain specification and have only one chain, the information about chain ID is not required. If there are multiple chains and the supplied wild-type amino acid and residue ID match only one chain, the program assumes that the input belongs to a specific chain. Finally, the CUPSAT program gives comprehensive prediction results about the overall structural stability (stabilizing or destabilizing) and torsion values (favourable or unfavourable) along with the ΔΔG values (stability change upon mutation) in the next screens for the possible 19 amino acid residues. The atom potentials and torsional angle distribution are used to calculate the stability of proteins. The positive and negative ΔΔG represents the stabilizing and destabilizing effects respectively (Topham et al., 1997; Gromiha et al., 1999; Parthiban et al., 2006; Parthiban et al., 2007a; Parthiban et al., 2007b).

2.9 Molecular Mechanics-Poisson Boltzmann Surface Area (MM-PBSA)

MM-PBSA is a computational technique and the post-processing end-state method used to calculate the binding free energies of molecules in solution state. MM-PBSA is abbreviated as Molecular Mechanics-Poisson Boltzmann Surface Area. MM-PBSA is a program written in Python language for streamlining end-state free energy calculations using ensembles derived from MD and MC
simulations. The available implicit solvation models are, i) the Poisson–Boltzmann Model, ii) the Generalized Born Model, and iii) the Reference Interaction Site Model. Free energy calculations are used in various applications such as structure based drug design, protein structure determination and protein-carbohydrate interactions. The other methods to calculate free energies are Free Energy Perturbation (FEP), Replica Exchange Free Energy Perturbation (REFEP) and Thermodynamic Integration (TI). Steps involved in the calculation of binding free energies: a) build the starting structure and run the simulations (MD or MC) to obtain an equilibrated system, b) run the production simulation and obtain an ensemble of snapshots or simply trajectories and c) calculate the binding free energy of the protein-carbohydrate complexes and analyze the results. For example, the binding free energy calculations using MM-PBSA is determined by subtracting the binding free energies of unbound protein and carbohydrate from the binding free energy of the bound complex (Srinivasan et al., 1998; Bryce et al., 2001; Miller et al., 2012) as shown in Equation 2.17.

\[
\Delta G_{\text{bind}} = \Delta G_{\text{complex}} - (\Delta G_{\text{protein}} + \Delta G_{\text{glycan}}) \tag{2.17}
\]

\(\Delta G_{\text{bind}}\) represent the final estimated binding free energy obtained from electrostatic energy, van der Waals contribution from Molecular Mechanics and non-polar contribution to the solvation free energy. From the results of negative total binding free energy in kcal/mol, we can get the favourable protein-ligand complex but keep in mind that the result does not equal the real binding free energy since the MM-PBSA calculations does not considered the unfavourable entropy contribution to binding. The PB (Poisson Boltzmann) approach gives a
favourable bound state energy when compared with the GB (Generalized Born) energy. The output file contains all the average energies, standard deviations and standard error of the mean for GB followed by PB along with the error values of ΔG binding. MMPBSA.py is released with AmberTools under the GNU General Public License (Miller et al., 2012).

2.10 Modelling of glycans at the binding site of proteins: Eulerian rotation

The molecular structures of sialylglycans are generated using standard geometry at an arbitrary frame of reference by keeping the centroid of the sialic acid residue at origin, C2 atom along X-axis and C3 atom in the XY plane which is shown in Figure 2.2. The terminal sialic acid residues are often found in the binding pocket of the crystal structure of proteins, however the structure of penultimate residues are unclear in many of the proteins. These complex structures as such cannot be taken for further modelling studies. Hence it becomes necessary to fit the sialylglycan structures at the binding pocket with the generated-sialic acid overlapping with the crystal structure of sialic acid.

\[ \text{Figure 2.2. Model building of carbohydrate structure from the origin.} \]
The principles of rigid body dynamics and Eulerian rotations are used to accomplish this model building. The sialylglycan is built at the binding pocket of proteins by keeping its centroid coinciding with the crystallographic centroid. Not all the orientations are allowed but the information about the allowed orientations are inferred by varying the Eulerian angles $\Phi$, $\Theta$ and $\Psi$ from 0° to 360°, 0° to 180° and 0° to 360° respectively. A region is said to be allowed if there are no stereochemical clashes between the glycan and the active site residues of proteins. The increments are generally at 10° and a total of 23328 (36 x 18 x 36) orientations are checked for compatibility in the binding pocket when the step size is 10 degrees. For each orientation, the stereochemical clashes between the amino acid residues within 20Å of sialic acid residue are calculated. The orientations without stereochemical clashes are separated out and analyzed for hydrogen bonding distance criteria and then the percentages of allowed regions are calculated. The orientations which satisfy the Ramachandran contact criteria without making any stereo-chemical clashes give valuable insights into the binding modes of sialylglycans inside the binding pocket of proteins.

In this present work, the softwares and web tools viz AMBER (Assisted Model Building with Energy Refinement), NAMD (NAanoscale Molecular Dynamics), VMD (Visual molecular dynamics), PDB (Protein Data Bank), UCSF Chimera (University of California San Francisco - Chimera or simply Chimera), MolScript, Raster3D, CUPSAT (Cologne University Protein Stability Analysis Tool) and MM-PBSA (Molecular Mechanics-Poisson Boltzmann Surface Area) are used for molecular modelling and molecular dynamics simulations to design the cyborg lectins with distinct and desired glycan-binding specificities.