5-2020

Exploring Mechanical Properties and Configurational Energetics of Toxbox Using Molecular Dynamics Simulation

Shams Mehdi
The University of Texas Rio Grande Valley

EXPLORING MECHANICAL PROPERTIES AND CONFIGURATIONAL ENERGETICS OF TOXBOX USING MOLECULAR DYNAMICS SIMULATION

A Thesis

by

SHAMS MEHDI

Submitted to the Graduate College of
The University of Texas Rio Grande Valley
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

MAY 2020

Major Subject: Physics

EXPLORING MECHANICAL PROPERTIES AND CONFIGURATIONAL ENERGETICS OF TOXBOX USING MOLECULAR DYNAMICS SIMULATION

A Thesis
by
SHAMS MEHDI

COMMITTEE MEMBERS

Dr. Ahmed Touhami
Chair of Committee

Dr. Andreas Hanke
Committee Member

Dr. Soumya Mohanty
Committee Member

May 2020

ABSTRACT

Mehdi, Shams, Exploring Mechanical Properties and Configurational Energetics of Toxbox Using Molecular Dynamics Simulation. Master of Science (MS), May 2020, 58 pp., 1 table, 31 figures, references, 49 titles.

All-atom Molecular Dynamics Simulations (MDS) have been performed to obtain 5 ns trajectory of the solvated, neutralized, and equilibrated toxbox system in NPT ensemble at 300 K. This trajectory data has been used to calculate the configurational entropy of toxbox by employing a quantum mechanical approach. The method is based on evaluating determinant of the covariance matrix, built from generalized coordinates of all atoms for each frame. The upper limit to the configurational entropy of toxbox has been calculated to be 30,030 J/mol-K. A preliminary investigation has been conducted to study the effects of sequence-dependent DNA conformation (DNA Crookedness) on the mechanical properties of toxbox by implementing constant force MDS. Results of this research may serve as the reference for studying ToxT – DNA interactions.

ACKNOWLEDGMENTS

I will always be grateful to Professor Dr. A. Touhami, chair of my thesis committee for his permanent support and encouraging comments. My thanks go to thesis committee members: Professor Dr. A. Hanke and Professor Dr. S. Mohanty for their guidance and input towards the completion of my MS Thesis.

I would also like to thank the Graduate College, UTRGV and College of Sciences, UTRGV for the financial support that I received during my MS studies.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

The *Vibrio cholerae* bacterium is the causative agent of the acute diarrheal disease cholera. Cholera continues to be a significant problem in the developing world, where morbidity and mortality levels remain high. Researchers have estimated that each year there are 1.3 million to 4.0 million cases of cholera, and 21,000 to 143,000 deaths occur worldwide due to cholera [1]. This diarrheal illness occurs when *V. cholerae* infects and colonizes the small intestines of a human body. The symptoms of Cholera include watery diarrhea/ rice water stool, vomiting, muscle cramps, tiredness, rapid heart rate, low blood pressure, and renal failure. It can take a few hours up to 5 days for symptoms to appear after infection. In most cases, the infection is either mild or does not show any symptoms. However, approximately 5-10 percent of the infected individuals will suffer from severe Cholera [27]. In such cases, a rapid loss of body fluids leads to dehydration and the loss of vital electrolytes (sodium, chloride, potassium, and bicarbonate). Without appropriate treatment the infected person can die within hours.

1.1 Cholera – A Major Epidemic Disease

Cholera is a water-borne disease because its causative agent *V. cholerae* spreads primarily through contaminated water. The watery diarrhea produced by infected individuals contain many *V. cholerae* bacterial cells which can infect others if ingested [27]. Thus, it is unlikely that developed countries where drinking water and sewage systems are properly separated will ever see a large outbreak of Cholera. However, in the developing areas of the world Cholera epidemics are still a great public health concern.

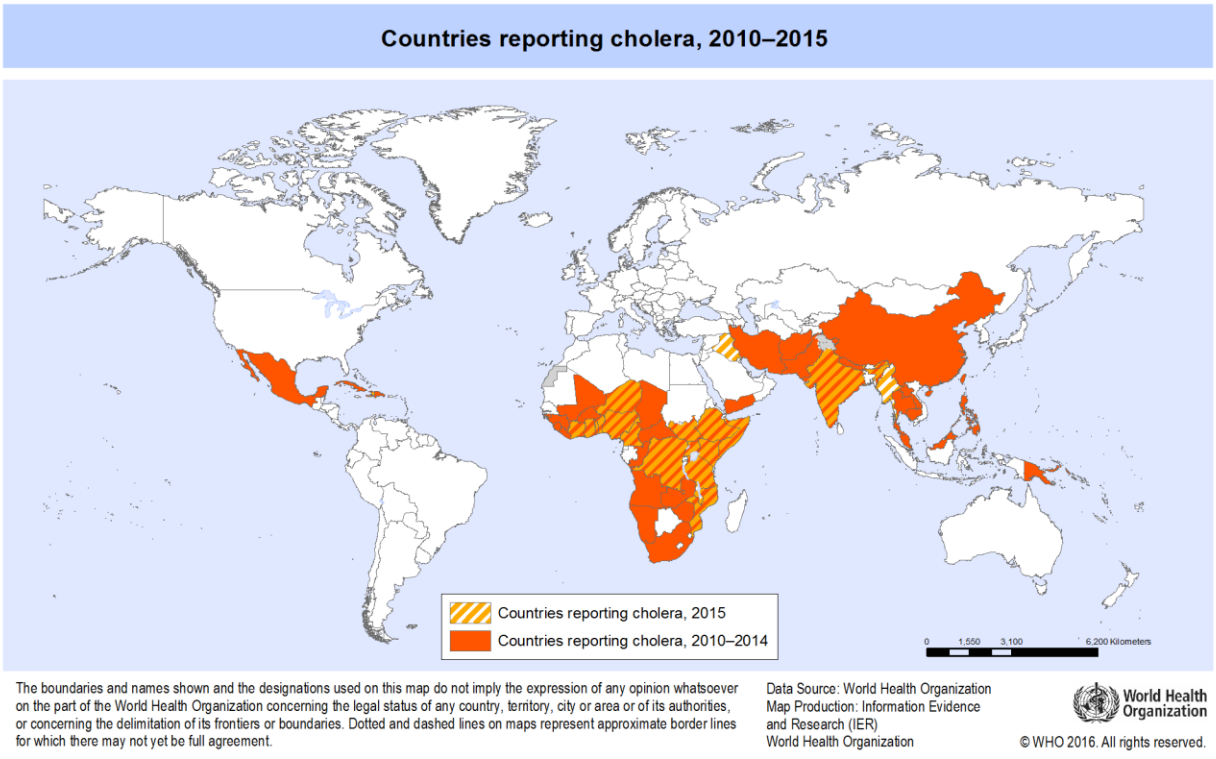


Figure 1. Countries reporting cholera, 2010-2015 (WHO) [48].

1.2 Lifecycle of *V. cholerae*

*V. cholera*e is a gram-negative, comma-shaped bacterium with a flagellum at one of the cell poles. The flagellum can rotate and provides propulsion to the bacterium.

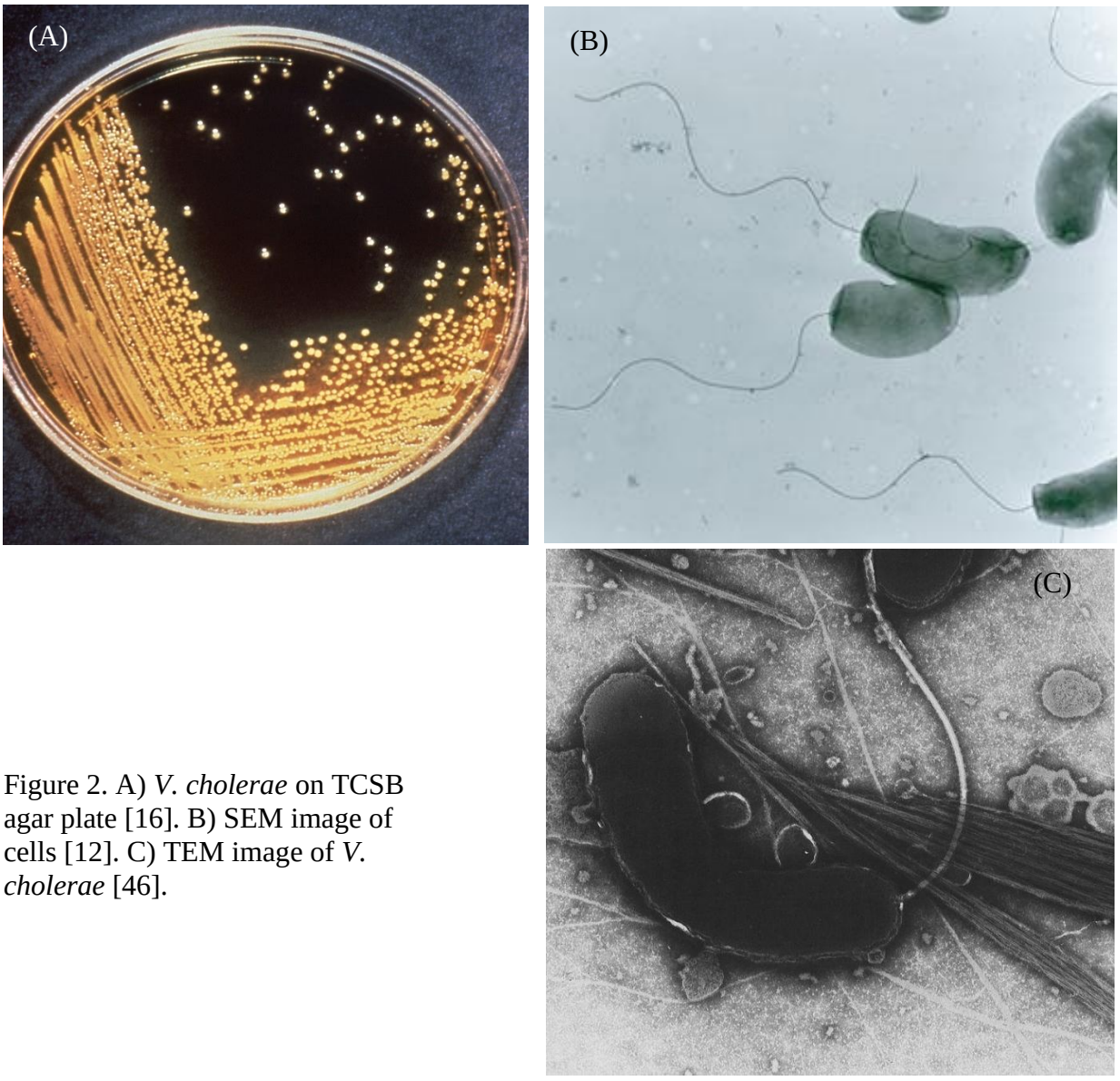


Figure 2. A) *V. cholerae* on TCSB agar plate [16]. B) SEM image of cells [12]. C) TEM image of *V. cholerae* [46].

V. cholerae thrives in saltwater reservoirs where they generally attach themselves to the shells of crabs, shrimps, and other shellfish. When a human ingests contaminated water/ food, the bacteria infects and subsequently colonizes the small intestines. The watery diarrhea produced by an infected person contains *V. cholerae* cells which make their way back to an aquatic reservoir and start the cycle again.

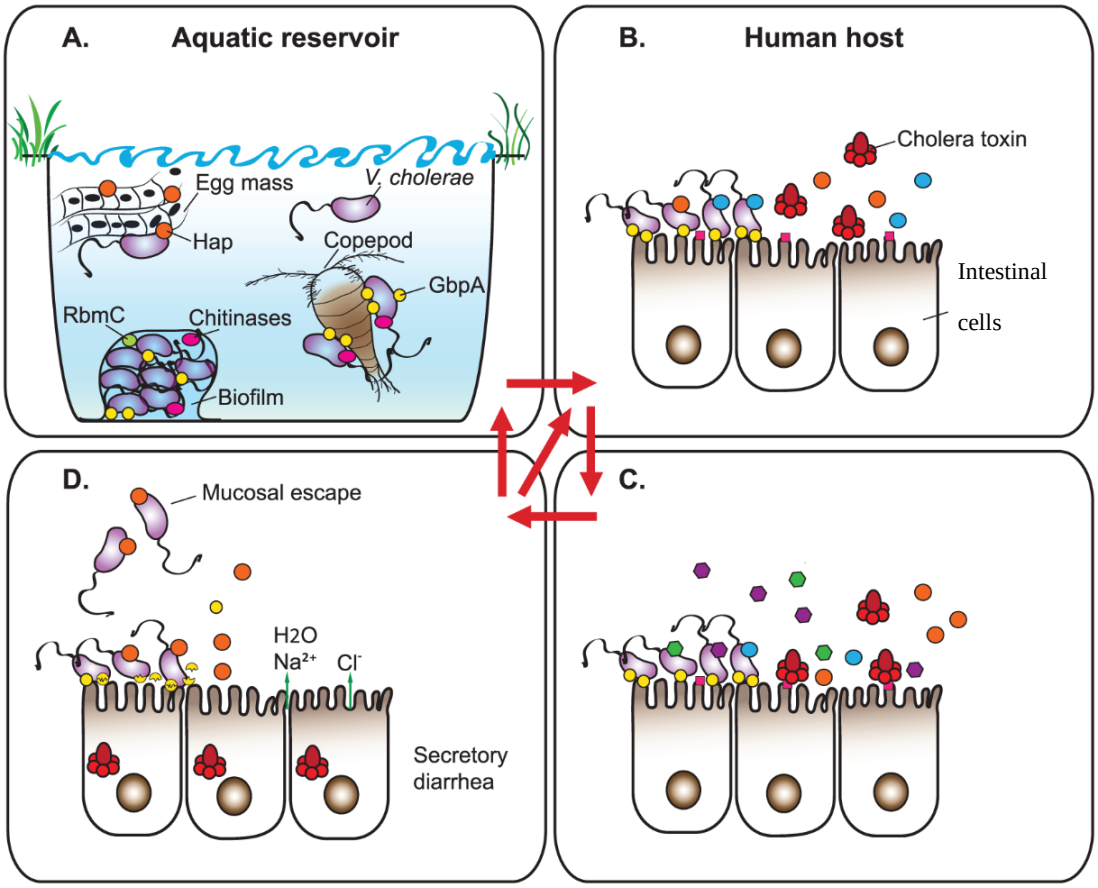


Figure 3. Lifecycle of *V. cholerae:* A) Planktonic bacterium thriving in an aquatic reservoir. B) Production of Cholera Toxin (CT). C) CT makes its way to intestinal cells. D) Release of bacteria through watery diarrhea [44].

1.3 Major Virulence Factors

The two major virulence factors produced by *V. cholerae* are the cholera toxin (CT) and the toxin-coregulated pilus (TCP). CT is a protein that causes diarrhea. TCP is a flexible appendage on the surface of bacterial cells required for aggregation of *V. cholerae* within the intestines.

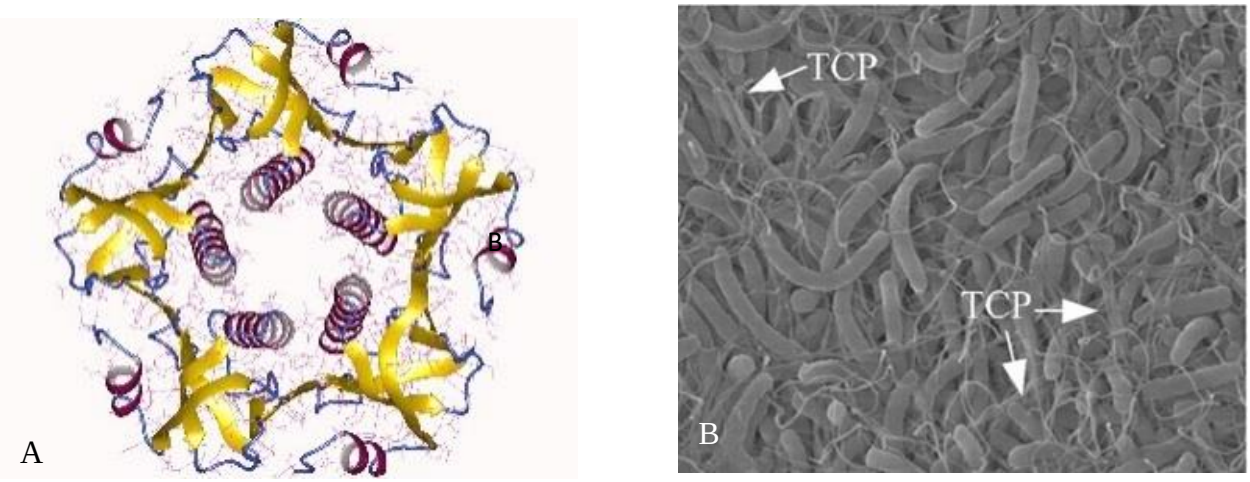
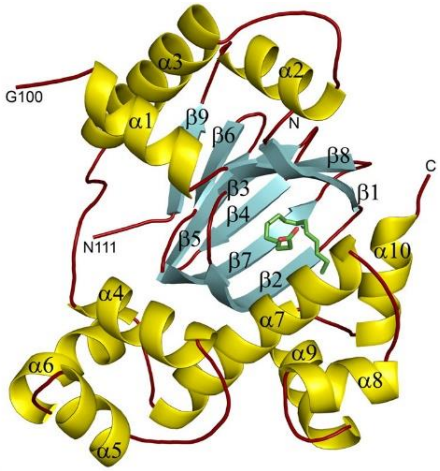


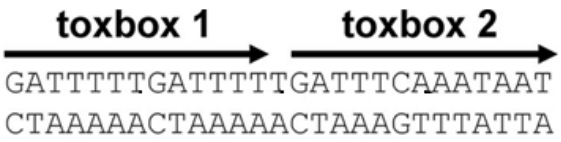
Figure 4. A) Cholera toxin (CT) [13]. B) Toxin-coregulated pilus (TCP) [29].

The direct activator of transcription of the majority of *V. cholerae* virulence genes is the ToxT protein [16], which is a member of the large AraC-family of transcriptional regulators. ToxT is a 32-kDa protein having a 100-amino-acid family domain in the C terminus that contains two helix-turn-helix domains for DNA binding [18,45]. The ToxT N-terminal domain (NTD) was proposed to be important for effector binding and dimerization [43]. There is significant evidence that ToxT binds DNA as a monomer [47]. However, bacterial two-hybrid studies revealed that the ToxT NTD is capable of dimerization when separated from the C-terminal domain (CTD), and ToxT dimerization after DNA binding may be important for the

transcription activation of some virulence genes [11]. The ToxT crystal structure also contains a buried unsaturated fatty acid, cis-palmitoleic acid, which was proposed previously to be a negative ToxT effector [32].



(A)



(B)

Figure 5. A) ToxT protein structure with 240 amino acids. B) Nucleotide sequences for toxbox 1 and toxbox 2.

ToxT binds to 13-bp sequences called toxboxes which are segments of 6 kbps genomic DNA. Toxboxes are located upstream of the genes whose transcription is activated by ToxT. Toxboxes are characterized by a well-conserved 5′ portion containing a poly(T) tract and a degenerate 3′ portion that is generally A/T rich. In addition to having somewhat degenerate sequences, toxboxes also vary in configuration and location relative to the transcriptional start site.

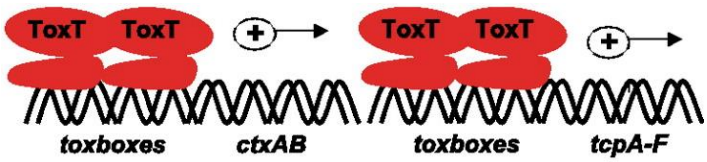


Figure 6. ToxT-toxbox interaction & ToxT dimerization.

1.4 Molecular Dynamics Simulations (MDS)

As the computing power of modern computers is growing spectacularly, computer simulations of complex biological systems are becoming viable. Such simulations have the potential to help us understand the molecular mechanisms that cannot be ascertained from observations or by conducting experiments. Currently, computer simulations are being performed to discover biological pathways [30,17,19], develop disease models [31], design drugs [38] etc.

There are two common simulation techniques that are generally employed by researchers when studying biological systems. They are: Molecular Dynamics Simulations (MDS) [40] and Monte Carlo Simulations (MCS) [8]. The choice of a technique depends on the specific aims of a research. The advantage of MCS is that it can be employed at atomic as well as the quantum scale. However, MDS is too computationally expensive to be used at the quantum scale. On the other hand, MDS can reveal dynamical properties including the molecular trajectory while MCS fails to do so [2]. As the algorithms used in this research utilize molecular trajectory to obtain results, MDS was chosen to conduct all computer simulations.

MDS can be classified into (i) quantum, (ii) all atom, (iii) coarse-grained, and (iv) multiscale approach. Although a quantum scale approach provides greater accuracy, it is computationally restricted to very small systems as mentioned previously. Coarse-grained approach [7] is primarily suited to study bulk properties or biological networks. A multiscale approach [9] is desirable when studying different length scales simultaneously. Thus, in this research, all atom MDS has been employed which evolves Newton's equations to produce a molecular trajectory in the spatial domain with time. In general, computational limitations restrict the timescale of all atom MDS to 1 μs. Although many biological processes such as

protein folding, take 1 μs – 1 ms to complete, all atom MDS can reveal useful information about the system. In all atom MDS, atoms are represented as classical "balls" that can inherit the properties of atoms as well as mimic the chemistry of the biomolecules [28].

$$U_{\text{total}} = \sum_{\text{Bonds}} K_r \left(r - r_0\right)^2 \quad \text{Bonds}$$

$$+ \sum_{\text{Angles}} K_\theta \left(\theta - \theta_0\right)^2 \quad \text{Polar Angles}$$

$$+ \sum_{\text{Dihedrals}} \frac{V}{2}\left[1 + \cos\left(n\phi - \gamma\right)\right] \quad \text{Dihedrals}$$

$$+ \sum_{\text{Atoms}} \varepsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12} - \left(\frac{R_{ij}}{r_{ij}}\right)^{6}\right] \quad \text{Van der Waals}$$

$$+ \sum_{\text{Charges}} \frac{q_i q_j}{\varepsilon r_{ij}} \quad \text{Electrostatics}$$

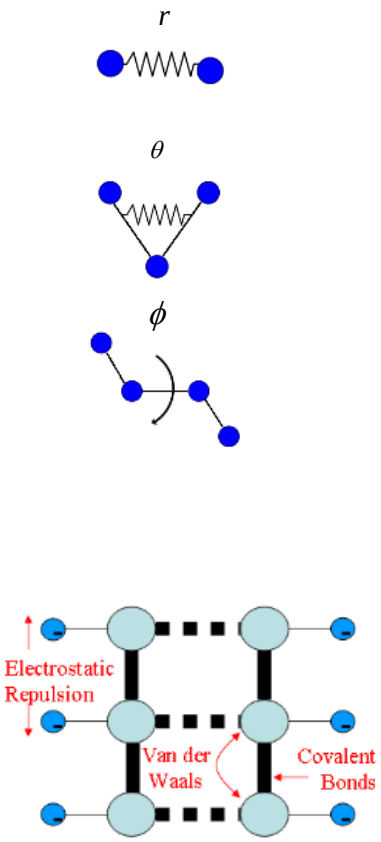


Figure 7. Form of the Hamiltonian employed in a typical MDS.

1.5 The Verlet Algorithm

In order to exploit the advantages of MDS, the choice of an efficient numerical algorithm is very important. Consider a system of N atoms with 3N position coordinates and 3N momentum coordinates. From these 6N coordinates the kinetic energy, K(**p**), the potential energy, U(**r**) and thus the Hamiltonian, H = K + U can be obtained. Using the Hamiltonian formalism, the equations of motions can be written in the form of a system of 6N coupled differential equations.

There are various algorithms that can be employed for solving a system of equations. However, an MDS algorithm needs to have certain features. For example, it needs to be able to perform an efficient numerical integration of the system of equations for long as well as short time scales. Moreover, the calculation of forces by performing integration is expensive and it is desired that the algorithm will do so as infrequently as possible. Additionally, the algorithm needs to ensure a constant energy at each step and an accurate measurement of the dynamical properties over long correlation times. As different forms of Verlet Algorithm satisfy these requirements, they are commonly used in MDS. A good overview of the algorithm can be found in the reference section of this literature [2].

$$\boldsymbol{p}_i(t+\tfrac{1}{2}\delta t) = \boldsymbol{p}_i(t) + \tfrac{1}{2}\delta t \boldsymbol{f}_i(t)$$
$$\boldsymbol{r}_i(t+\delta t) = \boldsymbol{r}_i(t) + \delta t \boldsymbol{p}_i(t+\tfrac{1}{2}\delta t)/m_i$$
$$\boldsymbol{p}_i(t+\delta t) = \boldsymbol{p}_i(t+\tfrac{1}{2}\delta t) + \tfrac{1}{2}\delta t \boldsymbol{f}_i(t+\delta t)$$

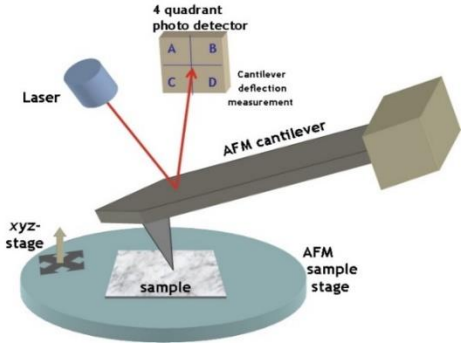
Figure 8. Velocity Verlet Algorithm [2].

1.6 Atomic Force Microscopy (AFM)

AFM is employed to conduct highly sensitive force measurements and record high resolution images [6]. A key component of AFM is the nanoscale tip, fixed at the end of AFM cantilever. The tip can be moved close to a sample (z axis movement) as well as along x-y plane in the nanoscale using a piezoelectric scanner. Interaction of the tip with the sample causes deflections in the cantilever. A photodiode measures these deflections by tracking the position of a laser which gets reflected from the backside of the cantilever (figure 9B). From these measurements nanoscale imaging of the sample can be obtained.

A key advantage of AFM compared to other microscopy techniques (such as SEM) is that it can be carried out in physiological conditions in the presence of liquid. This is a vital requirement for studying biological systems and is an effective technique for this research. Since, AFM has been used extensively in recent years to study DNA dynamics successfully [20,41,34], we have chosen to employ AFM in this research as well.

(A)



(B)

Figure 9. A) Atomic Force Microscope setup. B) AFM working principle.

1.7 Thesis Objective and Synopsis

The present work involves the application of all atom Molecular Dynamics Simulations (MDS) to conduct a comprehensive thermodynamic study on toxbox. Such a study may serve as the theoretical reference for studying ToxT-DNA interactions in future.

The DNA-ToxT complex is a paradigmatic disease model for transcriptional activators. Thus, an understanding of this interaction can be equally applicable to unravel the molecular mechanisms driving other host-pathogen interactions. A study of the affinity, conformational changes, and dynamics of this disease model were not known by the author to have been previously studied in this matter.

Chapter II of this literature is the materials & methods section. Background information involving hardware setup and software details is provided here. Preliminary experimental work by employing Atomic Force Microscopy (AFM) to study toxbox is discussed.

Chapter III presents the implementation details for obtaining molecular trajectory for toxbox. To accomplish this objective, all atom Molecular Dynamics Simulations (MDS) was employed. Afterwards, steered MDS was implemented to obtain toxbox conformational changes due to a constant pulling force.

Chapter IV is the post-processing & data analysis section. The configurational entropy of toxbox is calculated by following the Covariance Matrix approach. Preliminary studies have been conducted to examine the effects of sequence-dependent DNA conformation (DNA Crookedness) on mechanical properties (stretch modulus) of toxbox from the results of steered MDS.

Chapter V summarizes the results of the previous chapters. Suggestions are made for obtaining conclusive reference results using supercomputers.

CHAPTER II

MATERIALS & METHODS

2.1 Computational Hardware

The simulations were conducted using the “photon” and “gluon” workstations located at Cortez Hall, University of Texas Rio Grande Valley (UTRGV-Brownsville). Each workstation possesses 32 CPU cores (Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz) and two Nvidia GTX 780 ti (2880 cuda cores) connected through SLI. Each time step of the molecular trajectory simulations was conducted in parallel by utilizing the cuda cores while the analysis and certain other scripts were implemented using the CPU. Both workstations have CentOS 6.8 (64 bit) installed as the operating system.

2.2 Simulation Scripts

The molecular figures have been rendered using “Visual Molecular Dynamics” (VMD 1.9.3) [24]. Secure Shell (SSH) protocol [49] was used for transferring files and executing commands in the workstations.

The work in this study can be broadly categorized into system preparation, simulation, and analysis. Amber 14.0 is used as the primary simulation package which comes with hundreds of useful scripts for running simulations and analyzing the results. The scripts used in different stages of this research are specified in figure 10.

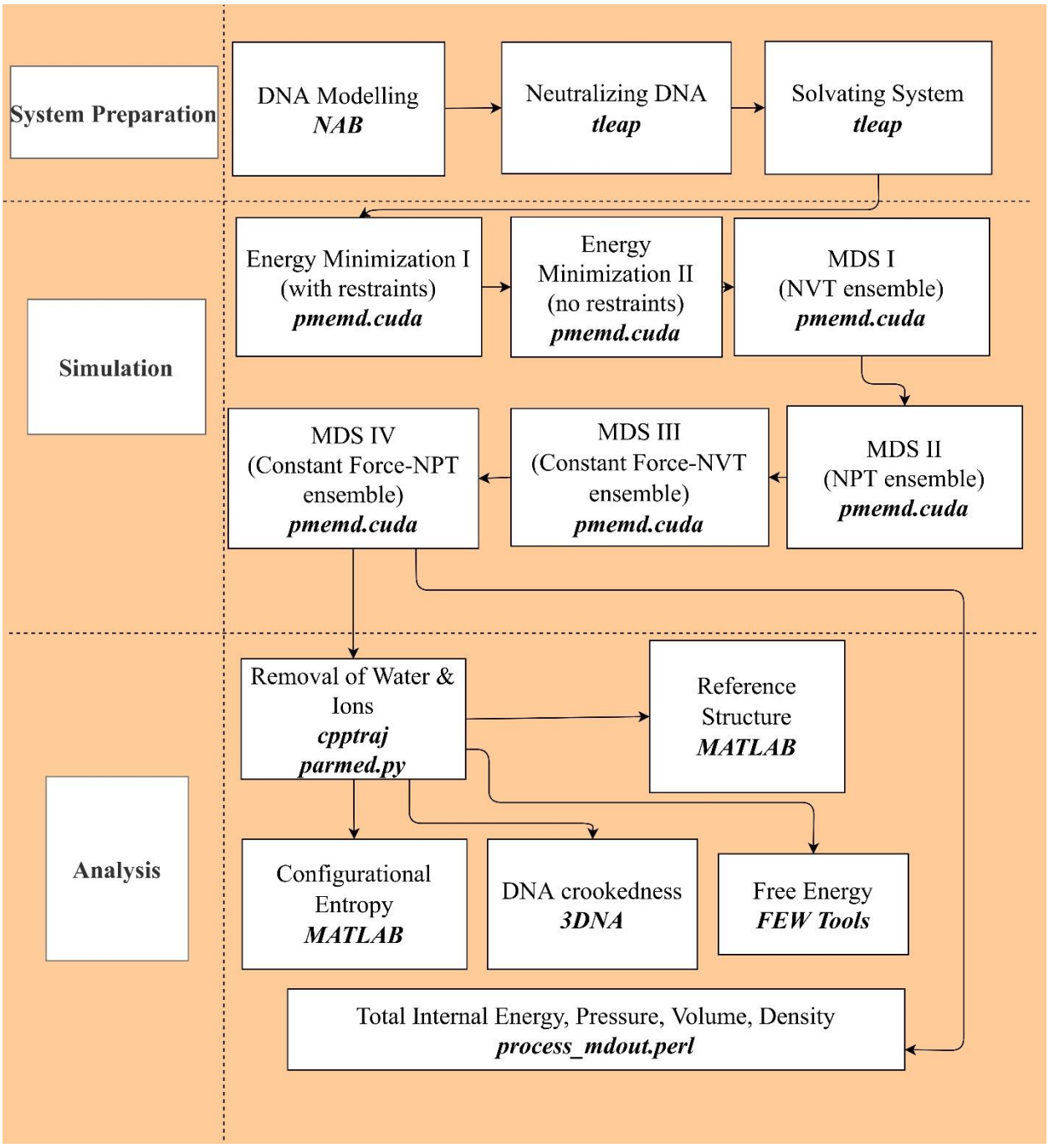


Figure 10. A diagram of the scripts used at different stages of this study.

2.3 AFM Imaging of DNA Molecule

Preliminary experimental work involving AFM has been conducted to obtain the height image of 6kbps stretch *V. cholerae* (strain O395) genomic DNA adsorbed on mica surface (figure 11). The DNA was synthesized by following optimized PCR protocols [25]. The Bruker AFM used in this study is located at Single Molecule Biophysics Lab, UTRGV-Brownsville. NanoScope 9.1 software was used to obtain the image and NanoScopeAnalysis 1.50 was used for post-processing.

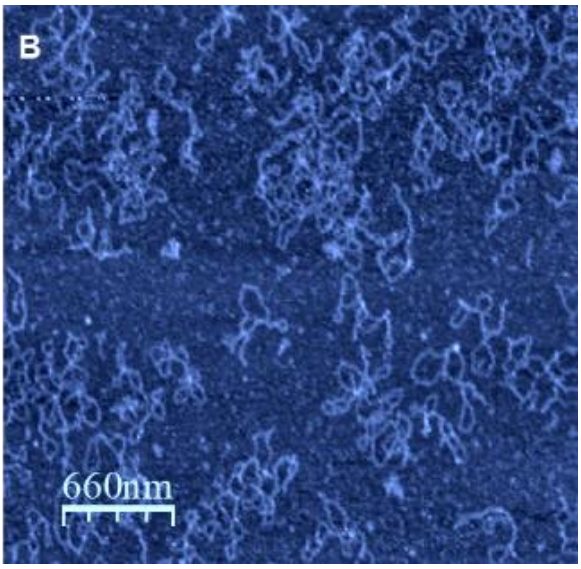


Figure 11. An AFM height image of *V. cholerae* genomic DNA imaged in liquid conditions.

2.4 Summary

The Photon and Gluon workstations were capable of performing MDS on the toxbox system upto a limited timescale of 5 ns due to computational and storage space limitations. A supercomputer/ computer cluster is necessary to perform MDS upto a desired timescale of 1 μs as well as to study the ToxT – DNA interactions. Thus, the available hardware placed a limitation on the scope of this study.

An important aspect of this research is that the theoretical/ simulation results can be verified through the use of AFM imaging and AFM force spectroscopy. However, soon after this study was started, the AFM began malfunctioning due to a possible circuit failure within the E-Box of the AFM. Due to this malfunction of the z piezo, the three legs would go out of synchronization and no vertical movement of the cantilever was possible. It was also observed that the error would randomly appear and go away which suggets a possible voltage fluctuation issue. This placed a limitation on the scope of the study and the investigation of the mechanical properties of toxbox was not possible by employing the AFM.

CHAPTER III

IMPLEMENTATION DETAILS

3.1 DNA Modelling

The following 35 bps (-CGCGGATTTTTGATTTTTGATTTCAAATAATCGCG-) DNA segment contains toxbox 1 and toxbox 2 which are required for the ToxT activation of PctxAB. This DNA segment has been modelled with Right-Handed B-DNA (Arnott) structure using the NAB (Nucleic Acid Builder) programming language. In order to reduce end effects, toxbox 1 and toxbox 2 have been placed in between (-CGCG-) sequences [3].

```
molecule m;
m = fd_helix( "abdna", "cgcggatttttgatttttgatttcaaataatcgcg", "dna" );
putpdb( "tb_d.pdb", m, "-wwpdb");
```

(A)

Helix type options for fd_helix()	
arna	Right Handed A-RNA (Arnott)
aprna	Right Handed A'-RNA (Arnott)
lbdna	Right Handed B-DNA (Langridge)
abdna	Right Handed B-DNA (Arnott)
sbdna	Left Handed B-DNA (Sasisekharan)
adna	Right Handed A-DNA (Arnott)

(B)

Figure 12. A) NAB code. B) Allowed helix type options.

- fd_helix is a function for creating Watson Crick duplexes. Its arguments are string helix type, string sequence and string acid type. Allowed values for string helix type are specified in figure 10 (c). String acid type can have values “dna” or “rna”. “-wwpdb” flag tells nab compiler to use Brookhaven PDB naming scheme for residue and atom.

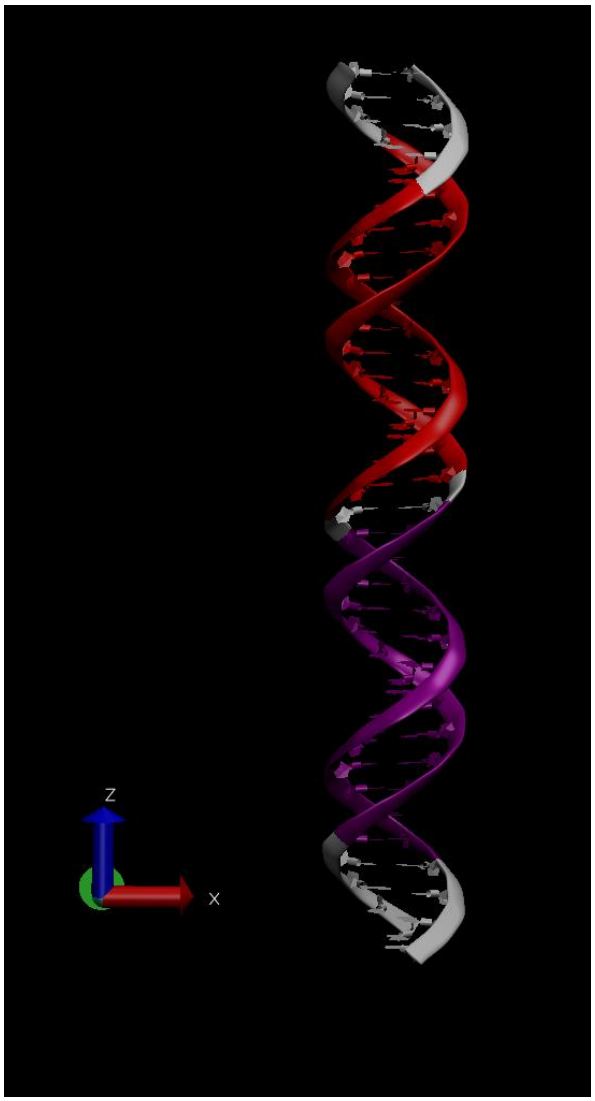


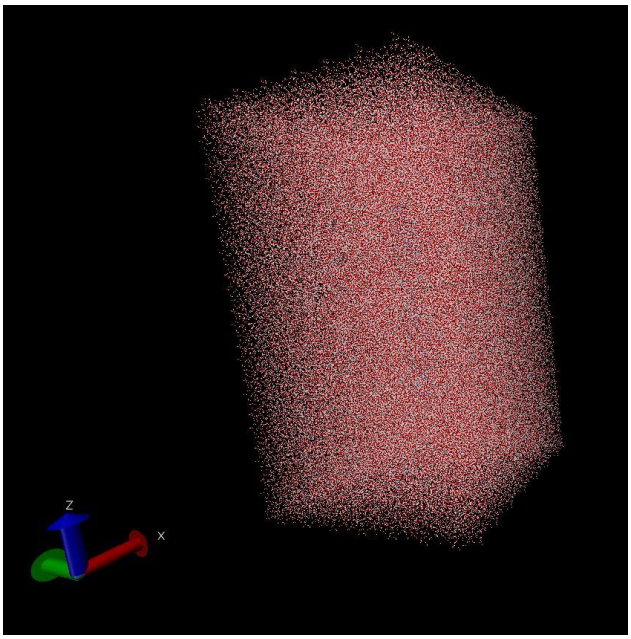
Figure 13. DNA segment obtained from NAB. Purple identifies toxbox 1 and red identifies toxbox 2.

3.2 System Generation

As the DNA molecule is negatively charged, Na+ counterions were added to neutralize the system using "tleap" module of Amber 14.0. Otherwise, the system would become unstable during MDS. Afterwards, the DNA segment was solvated in a rectangular box of water using the TIP3P water model. TIP3P was chosen because the angle between hydrogen atoms are kept fixed for computational efficiency. The rectangular box has dimensions 134 x134 x 224 Angstrom^3 and contains 133,922 water molecules. The dimension of the box was set up by adding 50 Angstroms to both sides of the DNA in all the three axes. The system was generated by using ff14SB force field [35]. "frcmod.ionsjc_tip3p" by Joung & Cheatham [26] was used as the monovalent ion parameters for Ewald and Tip3p water.

```
>source leaprc.ff14SB
>x = loadpdb "tb_d.pdb"
>loadamberparams frcmod.ionsjc_tip3p
>addIons x Na+ 0
>solvatebox x TIP3P 50.0
```

(A)



(B)

Figure 14. A) tleap code for system generation. B) Neutralized & solvated system

3.3 Energy Minimization

The default geometry obtained from NAB may not correspond to the actual minima in the force field. Because, NAB creates a cylindrical backbone and places the nucleotides sequentially without considering the inter-sequence interactions. If MDS is running using the NAB output, there is a great possibility that the DNA would become unstable. Thus before running MDS, a modified DNA structure corresponding to the closest local minima of the total internal energy was obtained by using “pmemd.cuda” simulation engine (GPU). The same could be done using the “sander” or “sander.MPI” simulation engines. However, those engines run the simulation on CPU as opposed to GPU. Thus, pmemd.cuda was chosen as it runs considerably faster on the hardware used in this research.

The minimization was performed in two steps. In the first step position restraints were put on the heavy atoms and a total of 5000 minimization steps were done (MDS I). Afterwards, the whole system is minimized for 5000 steps with no restraints (MDS II). The output files obtained in these steps were examined. It was observed that the energy of the system was significantly lower at the last step compared to the first step as shown in figure 16.

```
initial minimization with
constraints
 &cntrl
  imin   = 1,
  maxcyc = 5000,
  ncyc   = 2000,
  ntb    = 1,
  ntr    = 1,
  ig     = -1,
  cut    = 12.0
 /
Hold the DNA fixed
500.0
RES 1 70
END
END
```

(A)

```
minimization whole system
 &cntrl
  imin   = 1,
  maxcyc = 5000,
  ncyc   = 2000,
  ntb    = 1,
  ntr    = 0,
  ig     = -1,
  cut    = 12.0
 /
```

(B)

Figure 15. Code for: A) Energy minimization I. B) Energy minimization II.

```
================================================================================

   NSTEP       ENERGY          RMS            GMAX         NAME    NUMBER
      1      -1.7230E+06     7.4685E-01     4.1185E+01     C2'      2222

 BOND    =   136956.2407  ANGLE   =     1422.8419  DIHED      =     1610.1490
 VDWAALS =   366699.7253  EEL     = -2225418.1305  HBOND      =        0.0000
 1-4 VDW =      858.9581  1-4 EEL =    -5103.5652  RESTRAINT  =        0.0000

================================================================================

   NSTEP       ENERGY          RMS            GMAX         NAME    NUMBER
   5000      -1.7388E+06     1.3448E-01     2.3051E+01     O5'      1533

 BOND    =   140393.2365  ANGLE   =      322.5389  DIHED      =     1397.6264
 VDWAALS =   380673.6846  EEL     = -2256815.3387  HBOND      =        0.0000
 1-4 VDW =      518.4206  1-4 EEL =    -5305.0598  RESTRAINT  =        0.0000

================================================================================
```

Figure 16. Energy output comparison between the first step of MDS I and last step of MDS II.

Input variables used in the scripts for energy minimization are explained below:

- **IMIN = 1**: Minimization is turned on (no MDS)
- **MAXCYC = 5000**: Conduct a total of 5,000 steps of minimization.
- **NCYC = 2000**: Initially perform 2000 steps of steepest descent minimization followed by 3000 steps (MAXCYC - NCYC) of conjugate gradient minimization.
- **NTB = 1**: Use constant volume periodic boundaries (Particle Mesh Ewald (PME) is always "on" when NTB > 0).
- **CUT = 12.0**: Use a cutoff of 12 angstroms.
- **NTR = 1**: Use position restraints based on the GROUP input given in the input file. Use a force constant of 500 kcal mol^{-1} $angstrom^{-2}$ and restrain residues 1 through 70 so that the water and counterions are free to move. **NTR** = 0 for no constraint.
- **ig = -1**: Use MB distribution to generate random velocity between each restart**.**

3.4 Molecular Dynamics Simulation (MDS)

The energy minimized coordinates obtained from the last step was used as the initial coordinates for MDS. MDS was implemented in two steps that were both run in explicit solvent with periodic boundaries. In the first step (MDS I), the system was heated slowly at constant pressure with weak restraint on the DNA atoms using Langevin temperature equilibration scheme up to 300K for 1 ns (500,000 steps with step size 2 fs) in NVT ensemble.

```
1ns MD with res on DNA
 &cntrl
  imin   = 0,
  irest  = 0,
  ntx    = 1,
  ntb    = 1,
  cut    = 12.0,
  ntr    = 1,
  ntc    = 2,
  ntf    = 2,
  tempi  = 0.0,
  temp0  = 300.0,
  ntt    = 3,
  gamma_ln = 1.0,
  ig       = -1,
  nstlim = 500000, dt = 0.002
  ntpr = 100, ntwx = 500, ntwr =5000
 /
Keep DNA fixed with weak restraints
10.0
RES 1 70
END
END
```

Figure 17. Code for MDS I.

Input variables used in this step are explained below:

- **IMIN = 0**: Minimization is turned off (run molecular dynamics).
- **IREST = 0, NTX = 1**: Random initial velocities from a Boltzmann distribution are generated.
- **NTB = 1**: Use constant volume periodic boundaries (PME is always "on" when NTB>0).
- **CUT = 12.0**: Use a cutoff of 12 angstroms.
- **NTR = 1**: Use position restraints based on the GROUP input given in the input file. In this case we will restrain the DNA with a force constant of 10 kcal mol^{-1} $angstrom^{-2}$. Note that the restraint for energy minimization was much higher.
- **NTC = 2, NTF = 2**: SHAKE should be turned on and used to constrain bonds involving hydrogen.
- **TEMPI = 0.0, TEMP0 = 300.0**: We will start our simulation with a temperature, derived from the kinetic energy, of 0 K and we will allow it to heat up to 300 K. The system should be maintained, by adjusting the kinetic energy, as 300 K.
- **NTT = 3, GAMMA_LN = 1.0**: The Langevin dynamics should be used to control the temperature using a collision frequency of 1.0 ps^{-1}.
- **NSTLIM = 500000, DT = 0.002**: We are going to run a total of 500,000 molecular dynamics steps with a time step of 2 fs per step, possible since we are now using SHAKE, to give a total simulation time of 1 ns.
- **NTPR = 100, NTWX = 100, NTWR = 1000**: Write to the output file (**NTPR**) every 100 steps (200 fs), to the trajectory file (**NTWX**) every 100 steps and write a restart file (**NTWR**), in case our job crashes and we want to restart it, every 1,000 steps.

Afterwards, MDS was run in equilibrium without restraint at 300K for 5 ns (2,500,000 steps with step size 2 fs) in NPT ensemble. SHAKE algorithm was used to restraint hydrogen atom motion of water because (i) TIP3P water model requires such restraint, (ii) removal of such high frequency oscillations allow larger time steps. Results from this step was used for analysis. It should be noted that, NPT ensemble and not NVT ensemble resembles laboratory system as in a laboratory experiment pressure remains fixed and volume can vary.

```
5 ns MD
 &cntrl
  imin = 0, irest = 1, ntx = 7,
  ntb = 2, pres0 = 1.0, ntp = 1,
  taup = 2.0,
  cut = 12.0, ntr = 0,
  ntc = 2, ntf = 2,
  tempi = 300.0, temp0 = 300.0,
  ntt = 3, gamma_ln = 1.0,
  nstlim = 2500000, dt = 0.002,
  ig     = -1,
  ntpr = 100, ntwx = 500, ntwr = 5000
 /
```

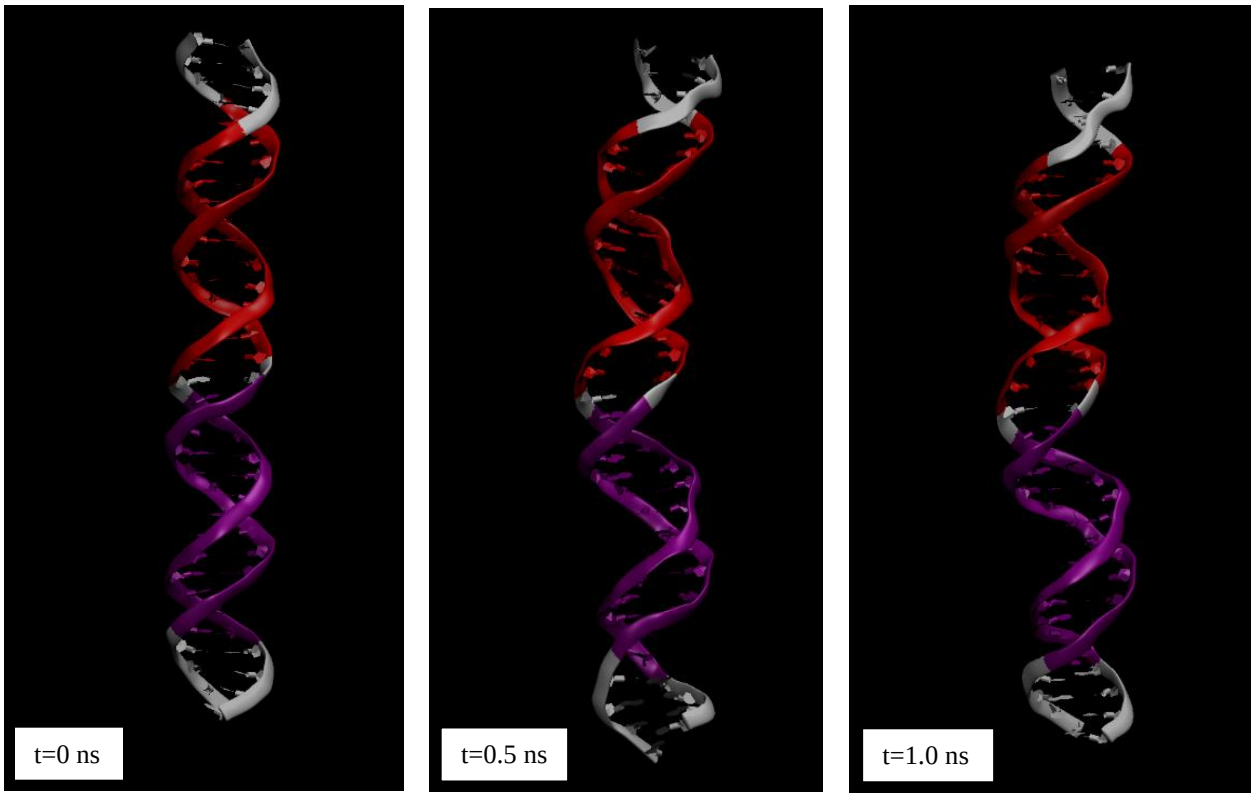
Figure 18. Code for MDS II.

Input variables used in this step are given below:

- **IREST = 1, NTX = 7**: MDS needs to continue from the last step instead of initializing. **IREST** tells pmemd.cuda that we want to restart a simulation, **NTX = 7** specifies that the coordinates, velocities and box information will be read from a formatted (ASCII) restart file.
- **NTB = 2, PRES0 = 1.0, NTP = 1, TAUP = 2.0**: Use constant pressure periodic boundary with an average pressure of 1 atm (**PRES0**). Isotropic position scaling should be used to

maintain the pressure (**NTP=1**) and a relaxation time of 2 ps should be used (**TAUP=2.0**).

- **NTR = 0**: We are no longer using positional restraints.
- **TEMPI = 300.0, TEMP0 = 300.0**: Our system was already heated to 300 K during the first stage of MD so here it will start at 300 K and should be maintained at 300 K.
- **NSTLIM = 2500000, DT = 0.002**: We are going to run a total of 2,500,000 molecular dynamics steps with a time step of 2 fs per step, it is possible since we are now using SHAKE, to give a total simulation time of 5 ns.



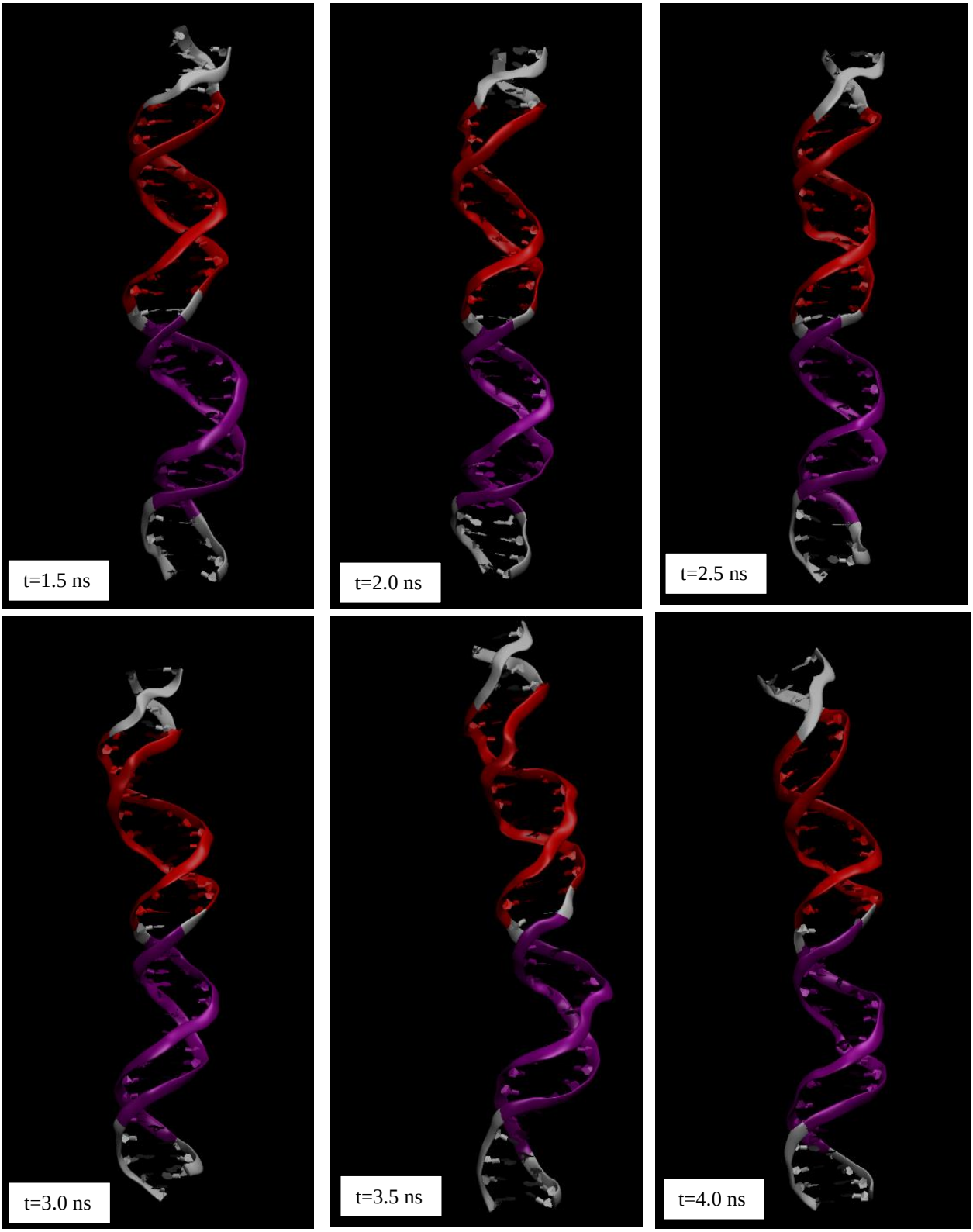


Figure 19. Molecular trajectory with time.

3.5 Steered MDS

An effective technique for measuring the mechanical properties of DNA such as the stretch modulus is Atomic Force Microscopy (AFM) as discussed in the next section. Such experiments are conducted by pulling one end of the DNA at a constant force by keeping the other end fixed. Such conditions can be computationally implemented by running steered MDS [36].

In order to facilitate a theoretical calculation of the stretch modulus, the results of MDS II were used to run steered MDS at constant forces of 1,5,10,20 pN. Constant force pulling on the DNA segment was implemented at first in NVT ensemble (MDS III) and afterwards in NPT ensemble (MDS IV). As the DNA explores the configurational space/ different microstates in the last step, results from this step was used for analysis. Both steps were implemented by exploiting a piecewise defined potential energy function available in Amber 14.0 [3]. As shown in figure 20, by imposing $R<r_1$ a constant force can be applied between two Centre of Masses (COM). In this research, the two COM were obtained by averaging the positions of C1' atoms belonging to 2^{nd}, 69^{th} and 34^{th}, 37^{th} residues respectively. An outward force was applied along a line joining these two COM to simulate constant force pulling.

$$E_{res}(R) = \begin{cases} A_1R + B_1 & \text{if } R \le r_1 \\ rk_2(R-r_2)^2 & \text{if } r_1 < R \le r_2 \\ 0 & \text{if } r_2 < R \le r_3 \\ rk_3(R-r_3)^2 & \text{if } r_3 < R \le r_4 \\ A_2R + B_2 & \text{if } R > r_4 \end{cases}$$

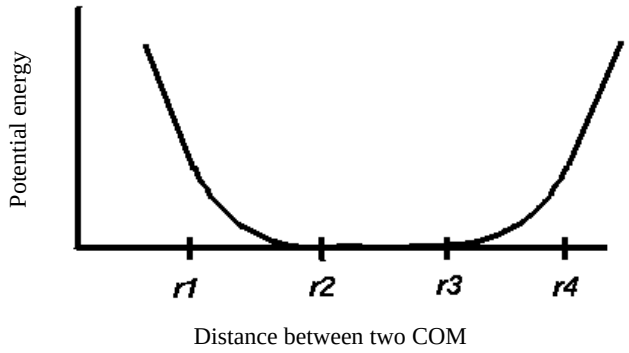


Figure 20. Form of the piece-wise defined function [37].

```
force input 2 ns
 &cntrl
 imin=0,irest=1,ntx=7, ntb=1, ntr=0, ntc=2, ntf=2, ntt=3, gamma_ln=1.0,
 ig=-1, nstlim=1000000, dt=0.002, pencut=-0.001, nmropt=1, ntpr=200,
 ntwx=1000, ntwr=1000, cut=9.0,
 /
 &wt type='REST', istep1=0,istep2=300000,value1=0.1, value2=1.0,        /
 &wt type='REST', istep1=300001,istep2=1000000,value1=1.0, value2=1.0,  /
 &wt
  TYPE="END"
 &end
LISTOUT=POUT
DISANG=rst
```

(A)

```
&rst iat=-1,-1,0
iresid=0,irstyp=0,ifvari=0, ninc=0, imult=0, ir6=0, ifntyp=0,
r1=900, r2=905, r3=910, r4=990, rk2=0.00144, rk3=0.00144,
igr1=39,2173,igr2=1063,1155,
/
```

(B)

Figure 21. A) Code for constant force MDS in NVT ensemble (MDS III). B) restraint file specifying COM for force = 1pN.

The input variables used in these scripts are explained below:

- **nmropt = 1:** Specifier for steered MDS.
- **&wt** tags gradually increase the force instead of applying full force from the beginning.
- **r1, r2, r3, r4** input variables in the restraint file determine the shape of the piece-wise potential energy function (figure 18). The strength of the force can be controlled by tuning **rk2** and **rk3** parameters (in the unit of kcal/mol*Angstrom^2). To find rk2 and rk3 the following equation can be used [23]: $F = 2\ rk2\ (r2 - r1)$. By setting, F = 1 pN, (r2 – r1) = 5 Angstroms, we get rk2 = 0.00144 kcal/mol*Angstrom^2.

- **iat = -1, -1**: Implements Center of Mass (COM) restraints defined by igr1 & igr2 atom groups.
- **igr1**: Find COM of the C1' atoms belonging to the 2nd and 69th residues.
- **Igr2**: Find COM of the C1' atoms belonging to the 34th and 37th residues.

Afterwards, constant force simulations in NPT ensemble was conducted.

```
5 ns NPT
 &cntrl
  imin=0, irest=1, ntx=7, ntb=2, pres0=1.0, ntp=1, taup=2.0,
  cut=9.0, ntr=0, ntc=2,ntf=2, tempi=300, temp0=300, ntt=3, gamma_ln=1.0,
  nstlim=2500000, dt=0.002, ig=-1, ntpr=200, ntwx=1000, ntwr=1000,
 /
 &wt
  TYPE="END"
 &end
LISTOUT=P2OUT
DISANG=rst
```

Figure 22. Code for constant force simulations in NPT ensemble.

3.6 Discussions

An observation of time evolution of the molecular trajectory shows clearly that the DNA segment undergoes local conformational changes. However, the segment retains an overall cylindrical/ rod-like structure even after MDS is complete. This can be explained by the fact that this DNA segment consists of 35 bps. Therefore, the length of this molecule is in the order of ~11 nm. As the persistence length of DNA is known to be in the range of ~50 nm [5], the result is consistent.

Performing steered MDS was the most difficult step in this chapter as the implementation of constant force pulling is not a standard protocol. Moreover, the GPU implementation of COM constraints for steered MDS require an updated version of Amber 14.0 (Amber 14.13 & AmberTools 14.27). As the workstations were running an older version of Amber (Amber 14.08 & AmberTools 14.22), earlier attempts at imposing COM restraints failed.

Input & output files for each of the six simulation steps were read and a summary for a single run is given in the next page. Although the computations were performed on GPU, it should be noted that CPU performed the task of writing data (atomic coordinates) on the hard drive. Thus, data must be transferred from GPU to CPU after each timestep which increases computation time. As a result, a high value of NTWX in the input (frequency of writing trajectory) not only increase storage space requirements but computation time significantly.

Simulation Step	No. of Timesteps	Simulated Time (ns)	Computation Time (hours)	Simulated Time/ Computation Time (ns/day)	NTWX Value	Trajectory Filesize (GB)	Trajectory Filesize With No Water (MB)
Energy Minimization I	5000	-	0.10	-	-	-	-
Energy Minimization II	5000	-	0.10	-	-	-	-
MDS I	500,000	1.0	6.24	3.85	500	9.81	-
MDS II	2,500,000	5.0	41.64	2.88	500	49.05	133.8
MDS III	500,000	1.0	8.73	5.50	1000	9.81	-
MDS IV	2,500,000	5.0	12.44	4.33	1000	24.52	66.91

Table 1. Input & output summary for a single run.

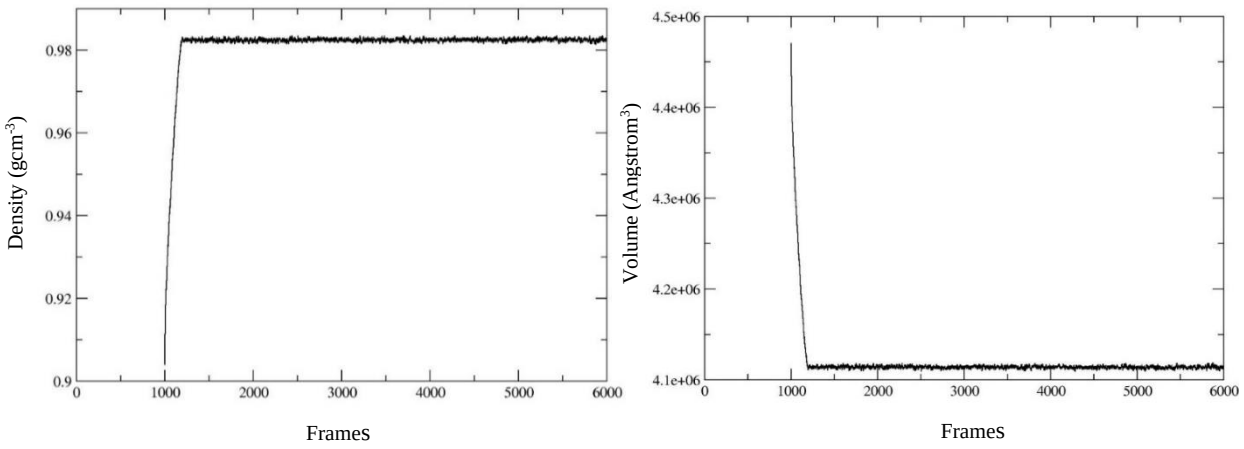
CHAPTER IV

ANALYSIS & RESULTS

The molecular trajectory obtained by performing MDS can be analyzed to explore the features of a biological system and evaluate dynamical properties of interest.

4.1 System Properties

After MDS was complete, “process_mdout.perl” script was run to obtain the system properties. The output files can be checked to examine the stability of the system.



(A) (B)

Figure 23. A) System volume. B) System density.

It can be observed (figure 23) that the volume initially decreases and density increases until they hit a plateu at 150 picoseconds (frame 150). They remain almost constant throughout the rest of the NPT simulation. The smooth transition from NVT to NPT ensemble followd by oscillations about a mean value suggests equilibriation has been successful. From the graph it can be seen that the system volume averages at 4,118,000 angstrom^3 and density averages at 0.981 gcm^{-3}.

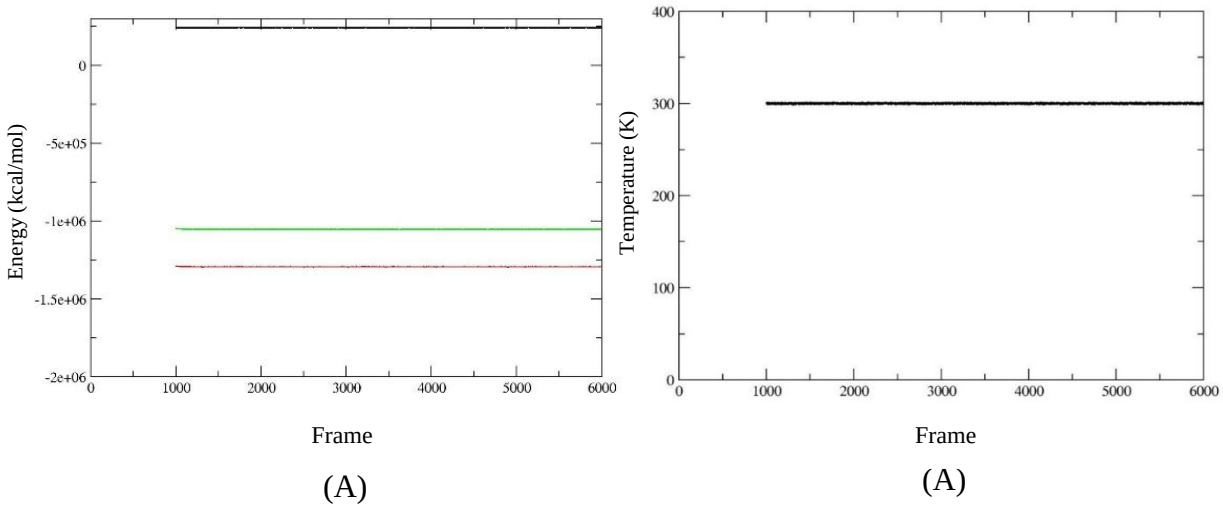


Figure 24. A) System Total energy. B) System Temperature.

A key requirement of the Verlet algorithm is that, the total energy of the system must be kept constant at the end of each step. It can be observed from figure 24A that the kinetic energy (black line) remains fixed indicating a constant temperature (figure 24B). A closer look at the potential energy (red line) and total energy (green), reveals that the values decreasd from initial values upto 150 ps after which both of them hit a plateu representing a relaxation of the system.

4.2 Configurational Entropy

Biological interactions such as molecular recognition, protein folding, DNA structure etc. seek to minimize the Gibbs Free Energy following the equation: $\Delta F = \Delta U - T\Delta S$, where ΔU is the Enthalpy change and ΔS is the Entropy change of the system. Thus, these thermodynamic parameters should to be explored to understand how the local conformational changes facilitate the biological function of toxbox.

The configurational entropy of the toxbox system was calculated by diagonalizing the covariance matrix as proposed by Jürgen Schlitter [42]. This quantum mechanical approach is useful because the form of the entropy equation is simple to evaluate. Moreover, it can be shown that the equation holds under classical limits. The approach defines covariance matrix for a molecular trajectory in terms of the position coordinates (3N) of the N atoms (N=1724) with respect to the time average structure (see figure 25A). If **M** is the mass matrix with atomic masses along the diagonal and zero elsewhere, then upper limit to the configurational entropy, S' is given by figure 25B. Entropy changes between two stable structures can be calculated from figure 25C. Eigenvalues obtained from diagonalization of the **Mσ** matrix can be interpreted as mass-weighted variances of generalized position coordinates corresponding to associated eigenmodes (figure 25B). Because of sampling limitations, entropy for a particular run will depend on temporal length [21,22] of the trajectory. However, S_∞ tends to an upper limit as the length of the simulation is increased.

$$\sigma_{ij} = \langle (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \rangle$$

(A)

$$S < S' = 0.5\, k \ln \det[\mathbf{1} + (kTe^2/\hbar^2)\mathbf{M}^{1/2}\boldsymbol{\sigma}\mathbf{M}^{1/2}] = 0.5\, k \ln \det[\mathbf{1} + (kTe^2/\hbar^2)\mathbf{M}\boldsymbol{\sigma}]$$

(B)

$$\Delta S \sim \Delta S' = 0.5\, k \ln \frac{\det[\boldsymbol{\sigma}_B + \mathbf{M}^{-1}\hbar^2/(kTe^2)]}{\det[\boldsymbol{\sigma}_A + \mathbf{M}^{-1}\hbar^2/(kTe^2)]}$$

(C)

Figure 25. A) Covariance Matrix. B) Configurational Entropy, S'. C) Change in entropy.

In this research, the calculated entropies between 0.8 – 5.0 ns runs can be fitted well using the power law relationship: $S(t) = S_\infty - at^b$ (see figure 26). The values of the fitting parameters are, S_∞ = 30,030 J/mol-K, a = 5,048,000 J/mol-K, b = - 0.85. Thus, at 300 K, TS = 2153.2027 kcal/mol. The fitting was performed using bisquare fitting scheme of MATLAB curve fiting tool. A table containing the calculated data points can be found in the Appendix section of this literature.

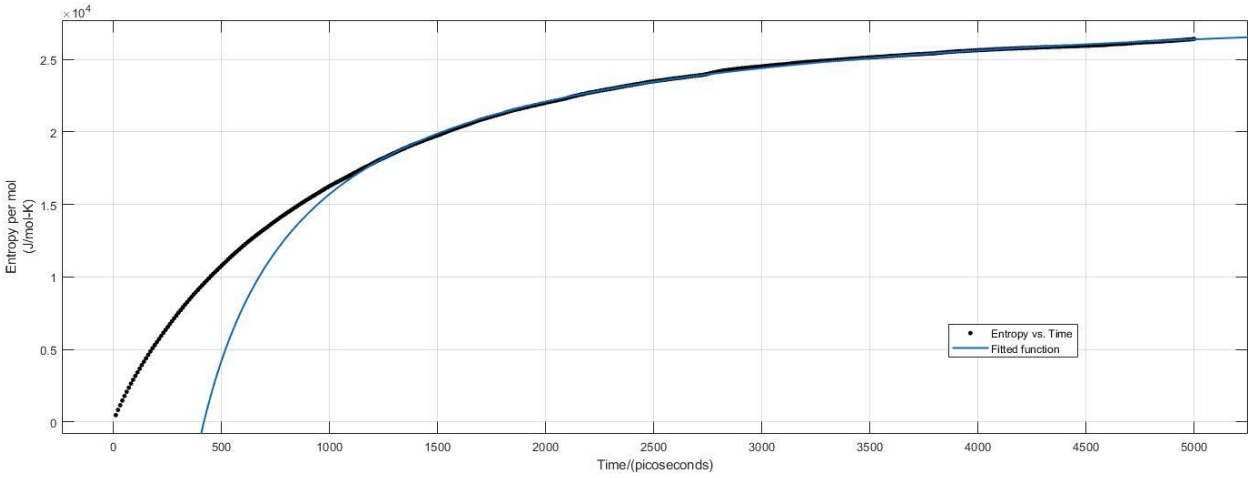


Figure 26. Configurational entropy of toxbox vs. time period of simulation.

Entropy calculations are performed by using the results obtained from MDS II. Here, two output files are of interest. The trajectory file and the original parameter/ topology file that were created during system generation. At first, the water molecules, Na+ ions, and 16 buffer bases (8 bps) at the two ends need to be removed as they are unnecessary for the calculations and will

increase the processing time tremendously. To remove the corresponding atoms from the trajectory file, cpptraj [14] was used and a new trajectory file was obtained. Similarly, the atoms were removed from the parameter/ topology file using parmed.py [39] and a new parameter/ topology file was obtained. Using cpptraj, atomic cordinates for each frame of the simulation were extracted in the form of .pdb files. Entropy calculations are performed using the MATLAB scripts given in the next page. The first script analyzes each time frame and produces new coordinates by eliminating translational and rotational motion of the DNA molecule. The second script implements calculation of entropy by diagonalization of the covariance matrices for the entire trajectory.

```
>parm tb_d.prmtop
>trajin f_npt_01.nc
>strip :WAT
>strip :Na+
>strip :67
>strip :67
>strip :67
>strip :67
>strip :36
>strip :36
>strip :36
>strip :36
>strip :32
>strip :32
>strip :32
>strip :32
>strip :1
>strip :1
>strip :1
>strip :1
>trajout f_npt_01_nw.nc
>run
```

(A)

```
>parm tb_d.prmtop
>strip :WAT
>strip :Na+
>strip :67
>strip :67
>strip :67
>strip :67
>strip :36
>strip :36
>strip :36
>strip :36
>strip :32
>strip :32
>strip :32
>strip :32
>strip :1
>strip :1
>strip :1
>strip :1
>parmout f_npt_01_nw.prmtop
>go
```

(B)

Figure 27. Code for: A) Reduced trajectory file. B) Reduced parameter/ topology file.

```
%First script for calculating new coordinates of the atoms

clear clc
%Input variables
first_frame = 1;
last_frame = 5000;

%Initialization
last_dna_atom = 1724;
reference_matrix = zeros([last_dna_atom 3]);
mass_matrix = zeros ([last_dna_atom*3 last_dna_atom*3]);

%Iteration for all frames
for z = first_frame : last_frame
   %Read file
   file_number = int2str(z);
   filename_syntax = 'frames.pdb.';
   filename=strcat(filename_syntax,file_number);

   %Create modified files
   fin = fopen(filename, 'r');
   ter=866;
   modified_filename = strcat('mod.',filename);
   fout=fopen(modified_filename, 'w');
   fgets(fin);
   for k = 1 : ter-1
        inline = fgets(fin);
        fwrite(fout,inline);
   end
   fgets(fin);
   for k = ter : last_dna_atom
         inline = fgets(fin);
         fwrite(fout,inline);
   end
   fclose(fout);
   fclose(fin);

   %Creating Coordinate Matrix
    c = fopen(modified_filename); d = textscan(c,'%*s %*s %*s %*s %*s %f %f %f %*[^\n]');fclose(c);
    current_matrix=cell2mat(d([1 2 3]));

    %Creating Mass Matrix
    c = fopen(modified_filename);
    f = textscan(c,'%*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %s');
    atom_names=[f{:}];
    writecell(atom_names,'atom_names');
    fclose(c);
    for o = 1:last_dna_atom
         om = (o-1)*3;
         if strcmp(atom_names{o},'H')
            mass_matrix(om+1,om+1) = 1.00794;
            mass_matrix(om+2,om+2) = 1.00794;
            mass_matrix(om+3,om+3) = 1.00794;
         elseif strcmp(atom_names{o},'C')
            mass_matrix(om+1,om+1) = 12.0107;
            mass_matrix(om+2,om+2) = 12.0107;
            mass_matrix(om+3,om+3) = 12.0107;
         elseif strcmp(atom_names{o},'N')
            mass_matrix(om+1,om+1) = 14.0067;
            mass_matrix(om+2,om+2) = 14.0067;
            mass_matrix(om+3,om+3) = 14.0067;
         elseif strcmp(atom_names{o},'O')
```

```
            mass_matrix(om+1,om+1) = 15.9994;
            mass_matrix(om+2,om+2) = 15.9994;
            mass_matrix(om+3,om+3) = 15.9994;
        elseif strcmp(atom_names{o},'P')
            mass_matrix(om+1,om+1) = 30.973762;
            mass_matrix(om+2,om+2) = 30.973762;
            mass_matrix(om+3,om+3) = 30.973762;
        else
            disp('Atom not found. Check pdb!')
        end
    end

    %Separation of COM
    COM = zeros ([1 3]);
    total_mass=0;
    for p = 1:last_dna_atom
        current_atom_mass=mass_matrix(p*3,p*3);
        for q = 1:3
            d=current_matrix(p,q);
            COM(1,q) = COM(1,q) + current_atom_mass*current_matrix(p,q);
        end
        total_mass=total_mass + current_atom_mass;
    end

    COM = COM/total_mass;
    for p = 1:last_dna_atom
        for q = 1:3
            current_matrix(p,q)= current_matrix(p,q)-COM(1,q);
        end
    end

    %Principal axis rotation
    inertia_tensor = zeros ([3 3]);
    for a = 1:3
        for b = 1:3
            all_atom_total=0;
            for i = 1:last_dna_atom
                first_component = 0;
                second_component = 0;
                if a == b
                    first_component = current_matrix(i,1)* current_matrix(i,1) + current_matrix(i,2)*
current_matrix(i,2)+current_matrix(i,3)* current_matrix(i,3);
                end
                second_component = current_matrix(i,a)*current_matrix(i,b);
                total = mass_matrix(i*3,i*3)*(first_component - second_component);
                all_atom_total = all_atom_total+total;
            end
            inertia_tensor(a,b)=all_atom_total;
        end
    end
    [eigen_vector,eigen_value] = eig(inertia_tensor);

    %Check orthogonality
    rotation=transpose(eigen_vector);
    check=rotation*eigen_vector;
    inertia_2=rotation*inertia_tensor*eigen_vector;

    %Ordering eigenvalues & eigenvectors
    [eigen_vector,eigen_value]=sortem(eigen_vector,eigen_value);

    %Define rotation matrix
    rotation_matrix=transpose(eigen_vector);
```

```
rotation_check=rotation_matrix*eigen_vector;

%Final inertia tensor (not needed for coordinate transform)
final_inertia = rotation_matrix*inertia_tensor*eigen_vector;

%New position coordinates after separation of principal axis rotation

for i=1:last_dna_atom
   old_pos_vector=zeros([3 1]);
   new_pos_vector=zeros([3 1]);
   for j=1:3
      old_pos_vector(j,1)=current_matrix(i,j);
  end
   new_pos_vector=rotation_matrix*old_pos_vector;
   for j=1:3
      current_matrix(i,j)=new_pos_vector(j,1);
      reference_matrix(i,j)=reference_matrix(i,j)+current_matrix(i,j);
   end
end

%Create new pdb files
filename_traj_maker=strcat(num2str(z),'.pdb');
fout =fopen(filename_traj_maker,'w');
fin=fopen(filename,'r');
fwrite(fout,fgets(fin));
fclose(fin);
fin2=fopen(modified_filename, 'r');
scan=textscan(fin2,'%s %s %s %s %s %s %s %s %s %s %s');
for k=1:ter-1
   tst1=scan{1,1};
   tst2=scan{1,2};
   tst3=scan{1,3};
   tst4=scan{1,4};
   tst5=scan{1,5};
   tst9=scan{1,9};
   tst10=scan{1,10};
   tst11=scan{1,11};
   line=tst1(k,1)+"  "+sprintf('%5s',char(tst2(k,1)))+"  "+sprintf('%-4s',char(tst3(k,1)))+" "+sprintf('%-
2s',char(tst4(k,1)))+"  "+sprintf('%-4s',char(tst5(k,1)))+"
"+sprintf('%8.3f',current_matrix(k,1))+sprintf('%8.3f',current_matrix(k,2))+sprintf('%8.3f',current_matrix(k,3))+"
"+sprintf('%-6s',char(tst9(k,1)))+"0.00          "+tst11(k,1)+"  "+newline;
   fwrite(fout, line);
end
fwrite(fout, 'TER    866      DT    27 ');
fwrite(fout, newline);
for k=ter:last_dna_atom
   tst1=scan{1,1};
   tst2=scan{1,2};
   tst3=scan{1,3};
   tst4=scan{1,4};
   tst5=scan{1,5};
   tst9=scan{1,9};
   tst10=scan{1,10};
   tst11=scan{1,11};
   line=tst1(k,1)+"  "+sprintf('%5s',char(tst2(k,1)))+"  "+sprintf('%-4s',char(tst3(k,1)))+" "+sprintf('%-
2s',char(tst4(k,1)))+"  "+sprintf('%-4s',char(tst5(k,1)))+"
"+sprintf('%8.3f',current_matrix(k,1))+sprintf('%8.3f',current_matrix(k,2))+sprintf('%8.3f',current_matrix(k,3))+"
"+sprintf('%-6s',char(tst9(k,1)))+"0.00          "+tst11(k,1)+"  "+newline;
   fwrite(fout, line);
end
fwrite(fout, 'TER   1725      DC    54 ');
fwrite(fout, newline);
```

```
        fwrite(fout, 'END   ');
        fwrite(fout, newline);
        fclose(fin2);
        fclose(fout);

        %Create new coordinate file for each frame
        new_coordinates=strcat('new_coordinates.',filename_syntax,file_number,'.txt');
        writematrix(current_matrix, new_coordinates,'Delimiter','tab')

        %Delete modified files
        delete(modified_filename);
        disp(z)
end

%reference matrix calculation and writing .txt file
reference_matrix=reference_matrix/(last_frame-first_frame+1);
reference_matrix_filename = 'reference structure';
writematrix(reference_matrix, reference_matrix_filename,'Delimiter','tab')

function [P2,D2]=sortem(P,D)
% this function takes in two matrices P and D, presumably the output
% from Matlab's eig function, and then sorts the columns of P to
% match the sorted columns of D (going from largest to smallest)
%
% EXAMPLE:
%
% D =
%    -90     0     0
%      0   -30     0
%      0     0   -60
% P =
%      1     2     3
%      1     2     3
%      1     2     3
%
% [P,D]=sortem(P,D)
% P =
%      2     3     1
%      2     3     1
%      2     3     1
% D =
%    -30     0     0
%      0   -60     0
%      0     0   -90
D2=diag(sort(diag(D),'descend')); % make diagonal matrix out of sorted diagonal values of input D
[c, ind]=sort(diag(D),'descend'); % store the indices of which columns the sorted eigenvalues come from
P2=P(:,ind); % arrange the columns in this order
end
```

```
%Second script for calculating Entropy

clear clc
%Input variables
first_frame = 1;
last_frame = 5000;
last_dna_atom = 1724;
entropy_cal_every=10;
holder = zeros(last_dna_atom*3, last_dna_atom*3);
data_points=zeros([last_frame/entropy_cal_every 2]);
reference_matrix_filename = 'reference structure.txt';
```

```
u=fopen(reference_matrix_filename);
v= textscan(u,'%f %f %f');fclose(u);
reference_matrix=cell2mat(v([1 2 3]));
reference_matrix_transpose=transpose(reference_matrix);
reference_matrix=reshape(reference_matrix_transpose, [1 last_dna_atom*3]);
atom_names='atom_names.txt';
g=fopen(atom_names);
h= textscan(g,'%s');fclose(g);
atom_names=[h{:}];
mass_matrix = zeros ([last_dna_atom*3 last_dna_atom*3]);
for o = 1:last_dna_atom
        om = (o-1)*3;
        if strcmp(atom_names{o},'H')
           mass_matrix(om+1,om+1) = 1.00794;
           mass_matrix(om+2,om+2) = 1.00794;
           mass_matrix(om+3,om+3) = 1.00794;
        elseif strcmp(atom_names{o},'C')
           mass_matrix(om+1,om+1) = 12.0107;
           mass_matrix(om+2,om+2) = 12.0107;
           mass_matrix(om+3,om+3) = 12.0107;
        elseif strcmp(atom_names{o},'N')
           mass_matrix(om+1,om+1) = 14.0067;
           mass_matrix(om+2,om+2) = 14.0067;
           mass_matrix(om+3,om+3) = 14.0067;
        elseif strcmp(atom_names{o},'O')
           mass_matrix(om+1,om+1) = 15.9994;
           mass_matrix(om+2,om+2) = 15.9994;
           mass_matrix(om+3,om+3) = 15.9994;
        elseif strcmp(atom_names{o},'P')
           mass_matrix(om+1,om+1) = 30.973762;
           mass_matrix(om+2,om+2) = 30.973762;
           mass_matrix(om+3,om+3) = 30.973762;
        else
           disp('Atom not found. Check pdb!')
        end
end
for z=first_frame:last_frame

   %Read file

   file_number = int2str(z);
   filename_syntax = 'frames.pdb.';
   filename=strcat('new_coordinates.',filename_syntax,file_number,'.txt');
    c = fopen(filename);
    f = textscan(c,'%f %f %f');fclose(c);
    current_matrix=cell2mat(f([1 2 3]));

   %Covariance matrix calculation

   sigma_matrix = zeros(last_dna_atom*3, last_dna_atom*3);
   current_matrix_transpose=transpose(current_matrix);
   current_matrix=reshape(current_matrix_transpose, [1 last_dna_atom*3]);
  for i=1:last_dna_atom *3
     for j=1:last_dna_atom*3
      sigma_matrix(i,j)=(current_matrix(1,i)-reference_matrix(1,i))*(current_matrix(1,j)-reference_matrix(1,j));
     end
  end
    holder = holder + sigma_matrix;
    disp (filename)

   %Entropy calculation every 10 frame and save data in .txt file
```

```
    if mod(z,entropy_cal_every)==0
      k = 1.38064852*10^(-23);
      h_bar = 1.054571628*10^(-34);
      temp = 300;
      euler = 2.7183;
      avogadro_number = 6.023*10^23;
      time_average_divisor = (z - first_frame+1);
      holder_c=holder/time_average_divisor;
      holder_c=mass_matrix*holder_c;
      constants =((1.660539066*10^(-27)*10^(-20)*k*temp*euler*euler)/(h_bar*h_bar));
      holder_term=(holder_c*constants);
      final_matrix=(eye(last_dna_atom*3)+holder_term);
      e=eig(final_matrix);
      log_microstates=0;
      for i=1:last_dna_atom*3
        if imag(e(i,1))==0
          log_microstates=log_microstates+log(e(i,1));
        end
      end
      entropy = 0.5*k*log_microstates;
      entropy_per_mole = entropy * avogadro_number;
      disp(entropy_per_mole)
      data_points(z/entropy_cal_every,1)=z;
      data_points(z/entropy_cal_every,2)=entropy_per_mole;
      data_points_filename = 'data points';
      writematrix(data_points, data_points_filename,'Delimiter','tab')
    end
end
```

Figure 28. MATLAB scripts for calculating configurational entropy of toxbox.

4.3 DNA Crookedness

In order to understand the molecular mechanisms of ToxT – toxbox interaction, it would be illuminating to conduct an investigation of the parameter(s) that regulate mechanical properties of toxbox. A model proposed by Gonzalez et al. [36] shows that sequence dependent local conformation called crookedness can regulate mechanical properties of a DNA. In the mentioned work, crookedness is defined by a parameter β (in radians) so that $\cos \beta = x / \Sigma l_i$, where x is the end-end distance of the DNA segment and Σl_i is the sum of distances between subsequent base pair centers. Thus, for a completely straight DNA, crookedness is zero.

Assume that an external force, F causes the end-end distanc of the DNA to change from $x(0)$ to $x(F)$. Two factors contribute to this extension. As shown in figure 30, they are: (i) an increase in the distance between subsequent base pairs (unstacking), and (ii) alignment of the base pair centers to helical axis (aligning) [36]. According to this model, the DNA can be represented by N springs in series. The first N-1 springs represent unstacking of the N-1 basepairs and the corresponding stiffness are $k_{l,i}$. The last spring represents the alignment of the DNA along the direction of the force and the corresponding stiffness is k_β. Thus, stretch modulus, S is defined using the equations below. The model also showed that, the $k_{l,i}$ contributions to S is minor. In this research, stretch modulus was calculated from k_β as the dominant contribution comes from it.

$$S^{-1} = \sum_{i=1}^{N-1} k_{l,i}^{-1} + k_\beta^{-1} \qquad k_\beta \equiv \frac{F}{\Delta \cos\beta(F) / \cos\beta(0)} \qquad k_{l,i} \equiv \frac{x(0)F}{\Delta x_{l,i}(F)}$$

Figure 29. Definitions of stretch modulus and stiffness [36].

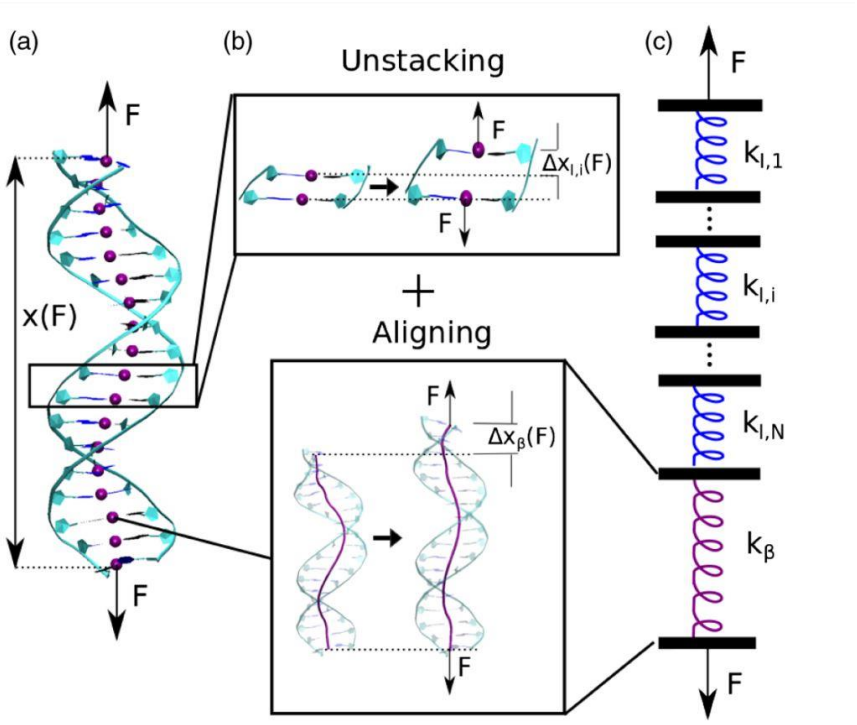


Figure 30 a) A double stranded DNA under external force, F. b) Unstacking and aligning due to F. c) Spring representation [36].

In this research preliminary investigation was conducted to theoretically calculate stretch modulus of toxbox based on the DNA crookedness model. Calculation of DNA crookedness involved the use of molecular trajectory obtained from MDS IV. Additionally, cpptraj was used to calculate an average structure from all of the trajectory snapshots and to find the DNA end-end distances, *x(0)* and *x(1pN)*. All of these steps were performed for each of the force values. Figure 21, 22 show example codes for force = 1pN.

Subsequently, 3DNA software [33] was used to calculate distance between consecutive basepairs, $l = \sqrt{Slide^2 + Shif^2t + Rise^2}$. 3DNA is an open-source software capable of calculating the features of a DNA as shown in figure 31. Finally, stretch modulus was calculated using the definitions in figure 29.

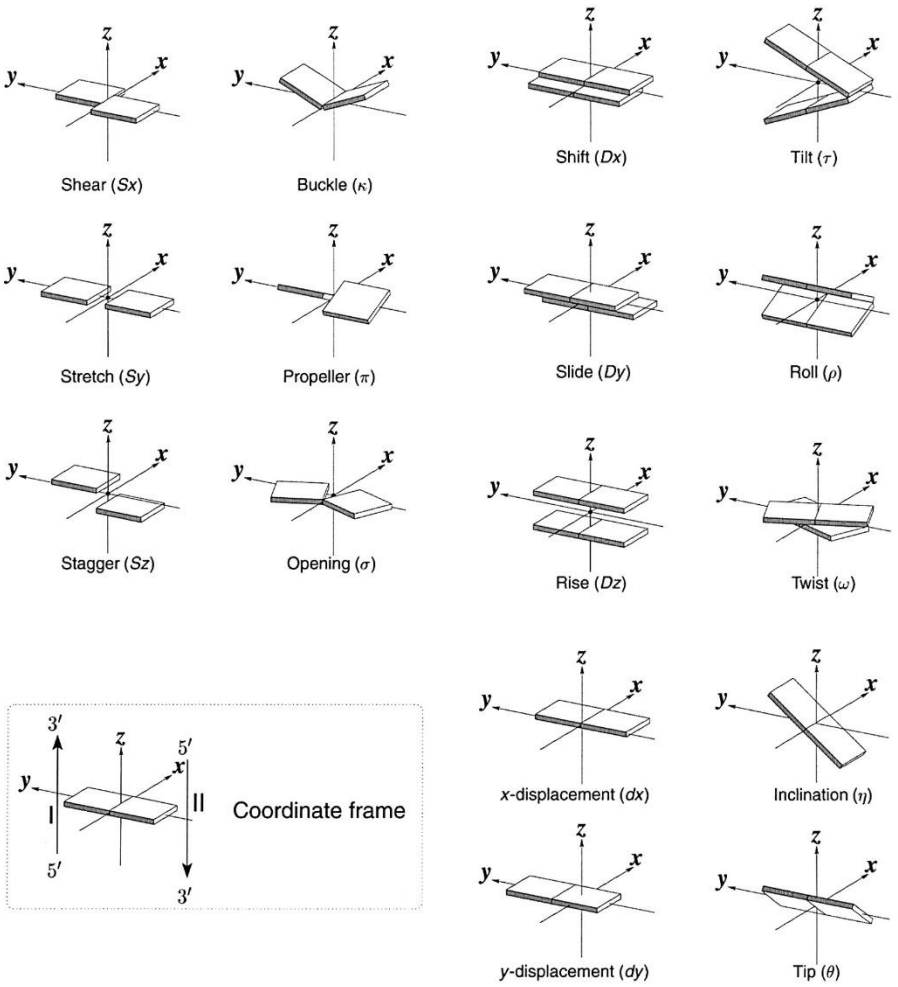


Figure 31. DNA base-pair properties measurable by 3DNA.

CHAPTER V

CONCLUSION

In the previous chapters we have presented a detailed scheme for performing MDS and analyzing the molecular trajectory of the toxbox system. An extensive overview of the configurational entropy calculation has been presented as well. The calculated value is in great agreement with the extensive research performed by Harris et al. (2001) [21,22]. However, in addition to the configurational entropy, it is important to explore free energy of toxbox to obtain a comprehenive understanding of the configurational energetics of the system. Such a study can be performed by employing the Free Energy Workflow Tool (FEW) of Amber 14.0 and is planned for future research.

A scheme for calculating the stretch modulus of the DNA segment based on the trajectory and conformational data has been presented in the previous chapter. Experimentally the stretch modulus can be obtained and compared to the calculated result to check the validity of the model. However, to obtain a reliable result for DNA crookedness, the production simulation should be run in the neighbourhood of 1 μs so that a large configurational space can be sampled. From that ~1 μs simulation, the data for atleast the first 200 ns should be discarded to ensure equilibriation. The scheme presented in this work can be implemented in supercomputers to run production simulations and to obtain final results. An allocation has been received for Lonestar5, a supercomputer located at Texas Advanced Computing Center (TACC), Texas where the algorithms are planned to be implemented and perform production simulations.

The work in this research is the first step in studying and understanding the molecular mechanisms of ToxT-DNA interactions. The next step is to incorporate ToxT protein in the system and conduct MDS which will facilitate a study of: (i) Conformational changes and thermodynamic parameters associated with binding of fatty acid to ToxT, (ii) Conformational changes and thermodynamic parameters associated with binding of ToxT to DNA with and without the presence of fatty acid, (iii) Insight into entropy and energetics of ToxT-DNA binding. Because of the promising outcome of the current research, it is expected that a completion of the future objectives would help to understand the molecular mechanisms behind Cholera.

REFERENCES

Ali, M., Nelson, A. R., Lopez, A. L. & Sack, D. A. (2015). Updated global burden of cholera in endemic countries. *PLoS Negl. Trop. Dis.* **9,** e0003832.

Allen, M. P. (2004). Introduction to molecular dynamics simulation. *Computational soft matter: from synthetic polymers to proteins*, *23*, 1-28.

Amber14 Reference Manual. Retrieved from https://ambermd.org/doc12/Amber14.pdf.

Bakan, A., Meireles, L. M., & Bahar, I. (2011). ProDy: protein dynamics inferred from theory and experiments. *Bioinformatics*, *27*(11), 1575-1577.

Bednar, J., Furrer, P., Katritch, V., Stasiak, A., Dubochet, J., & Stasiak, A. (1995). Determination of DNA persistence length by cryo-electron microscopy. Separation of the static and dynamic contributions to the apparent persistence length of DNA. *Journal of molecular biology*, *254*(4), 579-594.

Binnig, G., Quate, C. F., & Gerber, C. (1986). Atomic force microscope. *Physical review letters*, *56*(9), 930.

Bond, P. J., Holyoake, J., Ivetac, A., Khalid, S., & Sansom, M. S. (2007). Coarse-grained molecular dynamics simulations of membrane proteins and peptides. *Journal of structural biology*, *157*(3), 593-605.

Bouzida, D., Kumar, S., & Swendsen, R. H. (1992). Efficient Monte Carlo methods for the computer simulation of biological molecules. *Physical Review A*, *45*(12), 8894.

Brocos, P., Mendoza-Espinosa, P., Castillo, R., Mas-Oliva, J., & Pineiro, Á. (2012). Multiscale molecular dynamics simulations of micelles: coarse-grain for self-assembly and atomic resolution for finer details. *Soft Matter*, *8*(34), 9005-9014.

Centers for Disease Control and Prevention (CDC). (1971). Retrieved from https://www.cdc.gov/cholera/diagnosis.html.

Childers, B. M. *et al.* (2011). N-terminal residues of the Vibrio cholerae virulence regulatory protein ToxT involved in dimerization and modulation by fatty acids. *J. Biol. Chem.* 286, 28644–28655.

Cholera cells. Retrieved from https://news.ucsc.edu/2009/12/images/cholera_cells_300.jpg.

Cholera Toxin (CT). Merritt, E.A., Hol, W.G.J. (1996). Retrieved from Astrojan - http://www.rcsb.org/pdb/explore/explore.do?structureId=1chq.

CPPTRAJ manual. Retrieved from https://amber-md.github.io/cpptraj/CPPTRAJ.xhtml#magicparlabel-194.

Cui, Q., & Bahar, I. (Eds.). (2005). *Normal mode analysis: theory and applications to biological and chemical systems*. CRC press.

DiRita, V. J., Parsot, C., Jander, G. & Mekalanos, J. J. (1991). Regulatory cascade controls virulence in Vibrio cholerae. *Proc. Natl. Acad. Sci.* 88, 5403–5407.

Fuxreiter, M., Magyar, C., Juhász, T., Szeltner, Z., Polgár, L., & Simon, I. (2005). Flexibility of prolyl oligopeptidase: molecular dynamics and molecular framework analysis of the potential substrate pathways. *Proteins: Structure, Function, and Bioinformatics*, *60*(3), 504-512.

Gallegos, M. T., Schleif, R., Bairoch, A., Hofmann, K. & Ramos, J. L. (1997), Arac/XylS family of transcriptional regulators. *Microbiol. Mol. Biol. Rev.* 61, 393–410.

Ghosh, A., & Vishveshwara, S. (2007). A study of communication pathways in methionyl-tRNA synthetase by molecular dynamics simulations and structure network analysis. *Proceedings of the National Academy of Sciences*, *104*(40), 15711-15716.

Hamon, L., Pastre, D., Dupaigne, P., Breton, C. L., Cam, E. L., & Pietrement, O. (2007). High-resolution AFM imaging of single-stranded DNA-binding (SSB) protein—DNA complexes. *Nucleic acids research*, *35*(8), e58.

Harris, S. A., Gavathiotis, E., Searle, M. S., Orozco, M., & Laughton, C. A. (2001). Cooperativity in drug− DNA recognition: a molecular dynamics study. *Journal of the American Chemical Society*, *123*(50), 12658-12663.

Harris, S. A., Sands, Z. A., & Laughton, C. A. (2005). Molecular dynamics simulations of duplex stretching reveal the importance of entropy in determining the biomechanical properties of DNA. *Biophysical journal*, *88*(3), 1684-1691.

Hensen, U., Lange, O. F., & Grubmüller, H. (2010). Estimating absolute configurational entropies of macromolecules: The minimally coupled subspace approach. *PloS one*, *5*(2).

Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD: visual molecular dynamics. *Journal of molecular graphics*, *14*(1), 33-38.

Innis, M. A., & Gelfand, D. H. (1990). Optimization of pcrs. *PCR protocols: A guide to methods and applications*, *3*, 12.

Joung, I. S., & Cheatham III, T. E. (2008). Determination of alkali and halide monovalent ion parameters for use in explicitly solvated biomolecular simulations. *The journal of physical chemistry B*, *112*(30), 9020-9041.

Kaper, J. B., Morris, J. G., & Levine, M. M. (1995). Cholera. *Clinical microbiology reviews*, *8*(1), 48-86.

Karplus, M., & McCammon, J. A. (2002). Molecular dynamics simulations of biomolecules. *Nature structural biology*, *9*(9), 646-652.

Kirn, T. J., Lafferty, M. J., Sandoe, C. M., & Taylor, R. K. (2000). Delineation of pilin domains required for bacterial association into microcolonies and intestinal colonization by Vibrio cholerae. *Molecular microbiology*, *35*(4), 896-910.

Kong, Y., & Karplus, M. (2009). Signaling pathways of PDZ2 domain: a molecular dynamics interaction correlation analysis. *Proteins: Structure, Function, and Bioinformatics*, *74*(1), 145-154.

Kumar, A., & Purohit, R. (2014). Use of long term molecular dynamics simulation in predicting cancer associated SNPs. *PLoS computational biology*, *10*(4).

Lowden, M. J. *et al.* (2010) Structure of Vibrio cholerae ToxT reveals a mechanism for fatty acid regulation of virulence genes. *Proc. Natl. Acad. Sci.* 107, 2860–2865.

Lu, X. J., & Olson, W. K. (2003). 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic acids research*, *31*(17), 5108-5121.

Lyubchenko, Y. L., & Shlyakhtenko, L. S. (2009). AFM for analysis of structure and dynamics of DNA and protein–DNA complexes. *Methods*, *47*(3), 206-213.

Maier, J. A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K. E., & Simmerling, C. (2015). ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. *Journal of chemical theory and computation*, *11*(8), 3696-3713.

Marin-Gonzalez, A., Vilhena, J. G., Moreno-Herrero, F., & Perez, R. (2019). DNA crookedness regulates DNA mechanical properties at short length scales. *Physical review letters*, *122*(4), 048102.

Marin-Gonzalez, A., Vilhena, J. G., Perez, R., & Moreno-Herrero, F. (2017). Understanding the mechanical response of double-stranded DNA and RNA under constant stretching forces using all-atom molecular dynamics. *Proceedings of the National Academy of Sciences*, *114*(27), 7049-7054.

Mortier, J., Rakers, C., Bermudez, M., Murgueitio, M. S., Riniker, S., & Wolber, G. (2015). The impact of molecular dynamics on drug design: applications for the characterization of ligand–macromolecule complexes. *Drug Discovery Today*, *20*(6), 686-702.

Parmed.py Manual. Retrieved from https://parmed.github.io/ParmEd/html/parmed.html.

Rapaport, D. C., & Rapaport, D. C. R. (2004). *The art of molecular dynamics simulation*. Cambridge university press.

Sanchez, H., Suzuki, Y., Yokokawa, M., Takeyasu, K., & Wyman, C. (2011). Protein–DNA interactions in high speed AFM: single molecule diffusion analysis of human RAD54. *Integrative Biology*, *3*(11), 1127-1134.

Schlitter, J. (1993). Estimation of absolute and relative entropies of macromolecules using the covariance matrix. *Chemical Physics Letters*, *215*(6), 617-621.

Shakhnovich, E. A., Hung, D. T., Pierson, E., Lee, K. & Mekalanos, J. J. (2007). Virstatin inhibits dimerization of the transcriptional activator ToxT. *Proc. Natl. Acad. Sci.* 104, 2372–2377.

Sikora, A. E. (2013). Proteins secreted via the type II secretion system: smart strategies of Vibrio cholerae to maintain fitness in different ecological niches. *PLoS pathogens*, *9*(2).

Tobes, R. & Ramos, J. L. (2002). AraC-XylS database: a family of positive transcriptional regulators in bacteria. *Nucleic Acids Res.* **30,** 318–21.

Uchida, Y., Hasegawa, J., Chinnapen, D., Inoue, T., Okazaki, S., Kato, R., ... & Iemura, S. I. (2011). Intracellular phosphatidylserine is essential for retrograde membrane traffic through endosomes. *Proceedings of the National Academy of Sciences*, *108*(38), 15846-15851.

Withey, J. H. & DiRita, V. J. (2005). Activation of both acfA and acfD transcription by Vibrio cholerae ToxT requires binding to two centrally located DNA sites in an inverted repeat conformation. *Mol. Microbiol.* 56, 1062–1077.

World Health Organization (WHO). Retrieved from http://gamapserver.who.int/mapLibrary/app/searchResults.aspx.

Ylonen, T., & Lonvick, C. (2006). The secure shell (SSH) protocol architecture.

APPENDIX

APPENDIX

CONFIGURATIONAL ENTROPY DATA

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
10	473.1132914
20	827.9011693
30	1154.872979
40	1475.640108
50	1783.46695
60	2078.97222
70	2364.659114
80	2642.697529
90	2907.162769
100	3173.524993
110	3424.300756
120	3673.74351
130	3920.266929
140	4157.021929
150	4390.328365
160	4630.922242
170	4861.033637
180	5084.708481
190	5302.028752
200	5514.409376
210	5723.501414
220	5933.677764
230	6144.7323
240	6345.610983
250	6548.99427
260	6751.661159
270	6953.823673
280	7148.074792
290	7337.65474
300	7530.133664

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
310	7717.414752
320	7903.064505
330	8084.842014
340	8257.942137
350	8439.217922
360	8613.052591
370	8785.735093
380	8952.61298
390	9114.761106
400	9272.208692
410	9431.642076
420	9588.753357
430	9742.73197
440	9894.274435
450	10054.6749
460	10203.99131
470	10350.26729
480	10497.60409
490	10641.88091
500	10784.40225
510	10930.95044
520	11071.84264
530	11219.41145
540	11358.59967
550	11493.75848
560	11627.94795
570	11758.35135
580	11885.98937
590	12019.70881
600	12153.04394

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
610	12277.50301
620	12399.05405
630	12517.75567
640	12639.09523
650	12758.01909
660	12873.984
670	12988.26658
680	13100.92963
690	13213.6022
700	13322.86755
710	13433.86901
720	13543.57772
730	13655.06356
740	13766.00284
750	13875.30884
760	13983.45372
770	14091.91293
780	14195.49143
790	14297.42189
800	14396.42152
810	14498.13729
820	14598.29893
830	14694.95991
840	14795.93465
850	14893.63841
860	14993.99385
870	15090.76271
880	15185.48256
890	15276.53068
900	15368.53187

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
910	15459.69371
920	15551.08758
930	15639.54051
940	15733.37353
950	15822.57775
960	15914.77545
970	16009.92743
980	16104.02484
990	16186.44507
1000	16269.20449
1010	16353.50461
1020	16437.25252
1030	16517.07607
1040	16592.35184
1050	16665.71604
1060	16743.45552
1070	16821.52124
1080	16895.9056
1090	16983.99783
1100	17064.5227
1110	17140.70779
1120	17218.99703
1130	17300.28238
1140	17384.76172
1150	17464.27838
1160	17543.34536
1170	17617.48248
1180	17688.2992
1190	17762.17536
1200	17839.86095
1210	17922.58383
1220	17999.03916
1230	18074.27261
1240	18144.7644
1250	18210.15274
1260	18276.01822
1270	18348.59233
1280	18417.05071
1290	18489.55435
1300	18558.11471

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
1310	18628.32356
1320	18695.08246
1330	18766.15585
1340	18833.69371
1350	18896.03797
1360	18961.71104
1370	19026.68855
1380	19094.08132
1390	19151.30312
1400	19208.78343
1410	19264.7625
1420	19321.40997
1430	19380.01604
1440	19443.52928
1450	19498.87638
1460	19554.07654
1470	19606.83467
1480	19663.76552
1490	19714.90365
1500	19766.69531
1510	19817.85171
1520	19872.88162
1530	19935.6527
1540	19997.10004
1550	20055.82298
1560	20114.99263
1570	20169.84458
1580	20222.41101
1590	20276.6013
1600	20329.04581
1610	20378.5641
1620	20426.15851
1630	20474.73887
1640	20521.50161
1650	20579.76318
1660	20632.92228
1670	20686.86156
1680	20743.86304
1690	20800.22314
1700	20849.74182

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
1710	20891.65111
1720	20935.66971
1730	20979.37725
1740	21021.0975
1750	21062.25114
1760	21105.06031
1770	21146.45799
1780	21186.08848
1790	21227.16071
1800	21266.32218
1810	21317.3471
1820	21368.36092
1830	21411.24941
1840	21451.06857
1850	21488.91309
1860	21526.1043
1870	21561.00003
1880	21600.27761
1890	21635.27646
1900	21670.87503
1910	21707.33837
1920	21740.76517
1930	21775.02858
1940	21808.31749
1950	21843.30707
1960	21878.67245
1970	21913.91953
1980	21950.09942
1990	21981.28361
2000	22013.76032
2010	22046.8584
2020	22078.03802
2030	22109.91725
2040	22142.83742
2050	22175.15141
2060	22206.66823
2070	22237.78695
2080	22269.73071
2090	22300.90005
2100	22343.582

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
2110	22383.30052
2120	22425.50016
2130	22462.02256
2140	22497.4241
2150	22533.55819
2160	22571.82155
2170	22604.14125
2180	22634.22574
2190	22670.99414
2200	22705.23282
2210	22731.88864
2220	22757.95584
2230	22784.91422
2240	22813.77319
2250	22840.20365
2260	22867.8645
2270	22895.03311
2280	22920.65674
2290	22945.48578
2300	22971.78249
2310	22996.7794
2320	23025.68612
2330	23054.76077
2340	23083.55845
2350	23110.24092
2360	23137.85677
2370	23164.82901
2380	23189.60195
2390	23215.53188
2400	23242.27038
2410	23267.04611
2420	23292.95193
2430	23318.59287
2440	23345.79563
2450	23370.87901
2460	23398.79867
2470	23425.28086
2480	23450.96361
2490	23472.43815
2500	23492.86353

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
2510	23512.51225
2520	23535.6374
2530	23554.99505
2540	23574.54165
2550	23596.25621
2560	23617.79506
2570	23638.3804
2580	23658.65865
2590	23678.12194
2600	23696.11456
2610	23713.98995
2620	23732.00915
2630	23750.79006
2640	23769.57579
2650	23792.52602
2660	23812.05615
2670	23830.8496
2680	23849.02395
2690	23866.95909
2700	23885.39374
2710	23904.34801
2720	23921.99987
2730	23940.32023
2740	23974.71634
2750	24008.23835
2760	24042.10354
2770	24075.31617
2780	24107.81534
2790	24139.24254
2800	24171.05103
2810	24202.52562
2820	24232.10479
2830	24250.20719
2840	24270.44643
2850	24288.72249
2860	24305.22423
2870	24322.63348
2880	24339.98065
2890	24358.60443
2900	24377.16251

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
2910	24394.65572
2920	24410.02241
2930	24425.25866
2940	24439.72588
2950	24453.97054
2960	24468.3925
2970	24482.201
2980	24496.58072
2990	24510.77668
3000	24523.51368
3010	24538.03655
3020	24553.5797
3030	24568.66367
3040	24584.25679
3050	24598.70931
3060	24611.28178
3070	24623.8596
3080	24637.55603
3090	24651.17672
3100	24664.44186
3110	24677.849
3120	24692.09308
3130	24706.29191
3140	24727.94511
3150	24746.49284
3160	24759.97216
3170	24772.67503
3180	24785.14718
3190	24796.77012
3200	24808.18834
3210	24819.92224
3220	24831.45006
3230	24842.54419
3240	24853.51233
3250	24865.05295
3260	24877.52568
3270	24887.53957
3280	24897.89143
3290	24908.9398
3300	24919.98934

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
3310	24931.60293
3320	24941.4535
3330	24952.17526
3340	24963.07765
3350	24974.20127
3360	24985.15474
3370	24998.16467
3380	25010.0362
3390	25021.14587
3400	25032.13674
3410	25042.83964
3420	25054.18885
3430	25065.01384
3440	25075.67797
3450	25086.85472
3460	25097.87982
3470	25108.26515
3480	25117.69435
3490	25127.34932
3500	25137.26602
3510	25147.86741
3520	25157.71246
3530	25167.0153
3540	25176.2727
3550	25187.16836
3560	25197.0682
3570	25208.40255
3580	25219.4488
3590	25230.56678
3600	25240.70835
3610	25250.84695
3620	25260.00364
3630	25269.25286
3640	25278.56383
3650	25287.13808
3660	25297.29479
3670	25307.37347
3680	25317.34764
3690	25328.08963
3700	25337.57669

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
3710	25346.71913
3720	25355.17456
3730	25363.74535
3740	25371.89575
3750	25380.10363
3760	25388.01543
3770	25396.40212
3780	25405.30156
3790	25413.14801
3800	25425.42937
3810	25437.99387
3820	25452.47492
3830	25465.94041
3840	25482.29552
3850	25495.4671
3860	25507.70885
3870	25517.6694
3880	25528.6582
3890	25538.9727
3900	25549.35365
3910	25559.90037
3920	25567.17476
3930	25574.85059
3940	25583.95653
3950	25592.2851
3960	25600.04309
3970	25609.55611
3980	25619.18399
3990	25626.85455
4000	25634.63077
4010	25642.36659
4020	25648.95912
4030	25656.58746
4040	25664.41331
4050	25672.02417
4060	25680.33757
4070	25688.85252
4080	25697.50098
4090	25704.44644
4100	25712.06433

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
4110	25719.09034
4120	25727.13805
4130	25734.19016
4140	25740.05034
4150	25747.25681
4160	25754.72325
4170	25761.32449
4180	25767.62926
4190	25773.12015
4200	25777.98756
4210	25783.43084
4220	25789.44029
4230	25795.52845
4240	25800.9195
4250	25805.62741
4260	25811.68159
4270	25817.71263
4280	25825.21089
4290	25832.04387
4300	25838.31428
4310	25843.76894
4320	25848.75895
4330	25853.52157
4340	25858.26863
4350	25864.07017
4360	25869.51403
4370	25875.30779
4380	25880.07534
4390	25885.28707
4400	25891.04659
4410	25895.96745
4420	25900.59548
4430	25904.56071
4440	25909.47034
4450	25915.07662
4460	25921.78108
4470	25927.41877
4480	25932.46188
4490	25936.32806
4500	25940.6209

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
4510	25945.26745
4520	25952.42169
4530	25958.71091
4540	25966.6668
4550	25974.62921
4560	25982.57168
4570	25987.98966
4580	25992.09608
4590	25997.78776
4600	26013.51105
4610	26027.33568
4620	26038.06044
4630	26047.06422
4640	26052.98841
4650	26060.68091
4660	26067.52
4670	26072.8171
4680	26082.15775
4690	26090.75233
4700	26104.61302
4710	26116.46102
4720	26129.55973
4730	26140.46496
4740	26149.08945
4750	26157.64146
4760	26167.42783
4770	26176.36036
4780	26184.43229
4790	26192.08236
4800	26199.67153
4810	26206.03113
4820	26212.78823
4830	26218.81023
4840	26226.3694
4850	26237.63399
4860	26248.09674
4870	26259.98008
4880	26269.77915
4890	26277.14703
4900	26288.12771

Simulation Time Period (picoseconds)	Entropy (J/mol-K)
4910	26300.22544
4920	26311.84622
4930	26322.90701
4940	26332.84303
4950	26344.06625
4960	26355.11882
4970	26367.41777
4980	26379.29917
4990	26393.92022
5000	26407.99676

BIOGRAPHICAL SKETCH

Shams Mehdi pursued and completed Bachelor of Science (Physics) from University of Dhaka, Bangladesh in 2017. Afterwards, he obtained his Master of Science (Physics) degree from University of Texas Rio Grande Valley in Spring, 2020. He is expected to begin his PhD studies in Biophysics at University of Maryland – College Park from fall 2020. He enjoys discussing science and welcomes questions. He can be contacted through his personal email: shamsmehdi222@gmail.com.