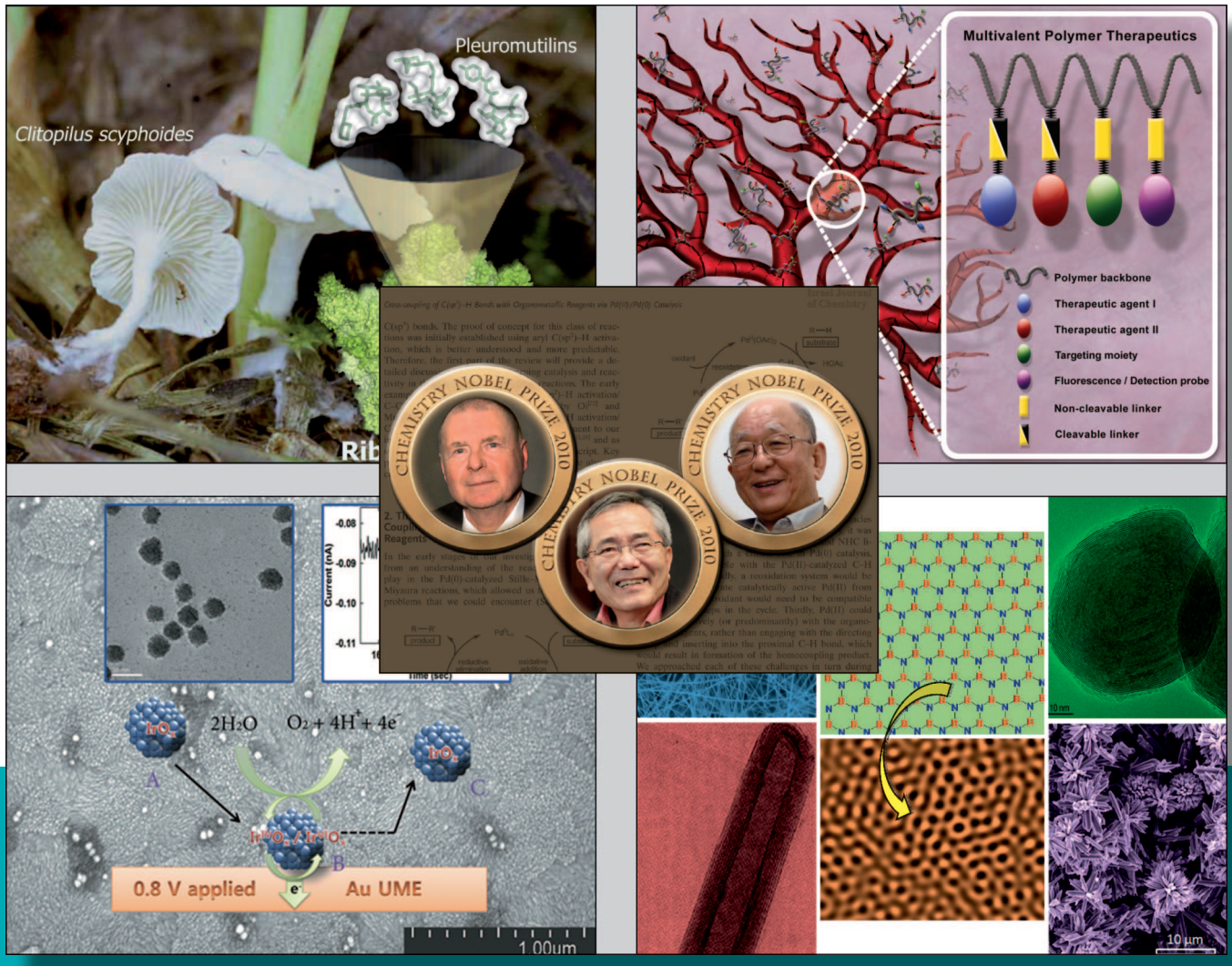
Israel Journal of Chemistry

IJC

Official Journal of the Israel Chemical Society

REPRINT

www.ijc.wiley-vch.de



WILEY-VCH

Israel Journal of Chemistry

IJC

Official Journal of the Israel Chemical Society

Table of Contents

WILEY-VCH

Review

Israel Journal of Chemistry

DOI: 10.1002/ijch.201400022

Coarse-Grained and Atomistic MD Simulations of RNA and DNA Folding

Jessica D. Leuchter,[a] Adam T. Green,[a] Julian Gilyard,[b] Cecilia G. Rambarat,[a] and Samuel S. Cho*[c]

Abstract: Although the main features of the protein folding problem are coming into clearer focus, the microscopic viewpoint of nucleic acid folding mechanisms is only just beginning to be addressed. Experiments, theory, and simulations are pointing to complex thermodynamic and kinetic mechanisms. As is the case for proteins, molecular dynamics (MD) simulations continue to be indispensable tools for providing a molecular basis for nucleic acid folding mechanisms. In this review, we provide an overview of biomolecular folding mechanisms focusing on nucleic acids. We outline the important interactions that are likely to be the main determinants of nucleic acid folding energy landscapes. We discuss recent MD simulation studies of empirical force field and Go-type MD simulations of RNA and DNA folding mechanisms to outline recent successes and the theoretical and computational challenges that lie ahead.

Keywords: DNA structures · Go-type models · molecular modeling · nucleic acids · RNA structures

1 Central Dogma of Biomolecular Folding and Assembly

There is no better "big picture" of genetic information flow to biological functions than the central dogma of molecular biology,[1] and we present it here in the context of biomolecular folding mechanisms (Figure 1). Of course, it is well known that DNA replication passes on information from one generation to the next, and it is transcribed into RNA, which in turn is translated into proteins. These proteins have been selected through evolution to fold into specific structures that correspond to a myriad of catalytic, transport, storage, structural, and regulatory functions in the cell.[2] Although notable exceptions to the original version of the central dogma exist, the general paradigms by which information flows from DNA to RNA to proteins and from sequences to structure to function still hold true. Misfolding events can result in aggregation that leads to diseases.[3,4]

In the decades since MD simulations first provided a molecular view of biomolecular dynamics,[5] a quantitative framework for how biomolecules assemble to form structures that correspond to functions is coming into clearer focus.[6] For proteins, Anfinsen's seminal experiments[7] moved the protein folding field, once a purely experimental problem, into one that could also be tackled through theory and computations because the information for finding a protein's 3D structure was largely self-contained in its sequence.[8] The still-complex problem of a conformational search for the folded state has been accepted to be simpler than a random search first put forth by Levinthal's Paradox,[9] and a large body of evidence from experiments, theory, and simulations have established that natural proteins have a funneled energy landscape for folding that is globally directed towards its native basin.[10–13] There are many excellent recent reviews on this fascinating subject,[10–12,14] and it continues to be an active area of research.[15–17]

However, we have also known for quite some time that there exist RNA, such as tRNA and rRNA, that do not act as passive intermediaries that code for protein sequences but instead fold into specific structures for other functions.[18] It was the discovery of ribozymes that are capable of enzymatic catalysis by Cech and co-workers that sparked a renewed interest in the functional roles of

[a] J. D. Leuchter, A. T. Green, C. G. Rambarat
Department of Physics
Wake Forest University
1834 Wake Forest Road
Winston-Salem, NC 27109 (USA)

[b] J. Gilyard
Department of Computer Science
Wake Forest University
1834 Wake Forest Road
Winston-Salem, NC 27109 (USA)

[c] S. S. Cho
Departments of Physics and Computer Science
Wake Forest University
1834 Wake Forest Road
Winston-Salem, NC 27109 (USA)
Phone: (+1) 336-758-3922
Fax: (+1) 336-758-6142
e-mail: choss@wfu.edu

RNA molecules.[19] We now know that RNA molecules are responsible for a host of biological functions that include not only catalysis, but also replication, transcriptional and translational regulation, and ligand binding.[2] DNA G-quadruplexes, which are a form of DNA found in G-rich regions of DNA, can inhibit telomerase activity, for example.[20] The determination of their folding mechanisms has been a focus of recent studies.[21–23] In addition, each of these biomolecules can also assemble to form quaternary structured proteins, RNA, and DNA oligomers, as well as protein–RNA and protein–DNA complexes. As such, DNA, RNA, and proteins all fold into specific structures that correspond to biological functions (Figure 1).

Although structural information of proteins still far outstrips that of RNA and DNA, as well as their structures in complex with proteins, many new and interesting structures have been solved by X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy.[24] However, the structures of folded nucleic acids by themselves do not provide crucial details about the energetics and driving interaction forces of their dynamics and folding mechanisms.

2 Nucleic Acid Folding Energy Landscapes

The main contours of nucleic acid folding energy landscapes for the folding are also being addressed using atomistic and coarse-grained MD simulations that can greatly enhance our experimental observations.[25–27] Since DNA and RNA adopt specific structures in a relatively short timescale, one can presume that their energy landscapes are also largely funneled towards a native basin. They consist of fewer possible subunits (nucleotides) than proteins, and their secondary structures consist of predictable Watson–Crick (WC) base-pairing rules. The folding

Jessica D. Leuchter is a 4th year undergraduate student at Wake Forest University. She is pursuing a B. S. degree in Chemistry with a concentration in Biochemistry. For the past year, her research has focused on all-atom MD simulations of protein–RNA complexes.

Adam Green is a 4th year undergraduate at Wake Forest University. He is pursuing a B. S. in Biology with a minor in Chemistry. He is the recipient of the Wake Forest Research Fellowship. For the past year, he has been conducting research on molecular modeling and all-atom MD simulations of DNA and RNA.

Julian Gilyard is a 2nd year undergraduate at Wake Forest University. He is pursuing B. S. degrees in Computer Science and Mathematical Economics. He is a Carswell Scholar and a recipient of the Wake Forest Research Fellowship. His research is focused on coarse-grained GPU-optimized biomolecular MD simulations.

Cecilia Rambarat is a 3rd year undergraduate at Wake Forest University. She is pursuing a B. A. in Economics with a minor in Chemistry. Her research is focused on all-atom MD simulations of protein–RNA complexes and intrinsically disordered proteins.

Samuel S. Cho is an assistant professor with a joint appointment in the Departments of Physics and Computer Science at Wake Forest University. He received B. S. degrees in Biochemistry and Computer Science from the University of Maryland, Baltimore County. He was also an undergraduate researcher with Alexander D. Mackerell, Jr. He went on to the University of California, San Diego, where he received a Ph.D. in Physical Chemistry with Peter G. Wolynes. He then performed postdoctoral research with Dave Thirumalai at the University of Maryland, College Park, where he was awarded the NIH (NRSA) Postdoctoral Fellowship. His broad interdisciplinary research interests in computational biophysics are focused on biomolecular folding and assembly mechanisms.

Israel Journal of Chemistry

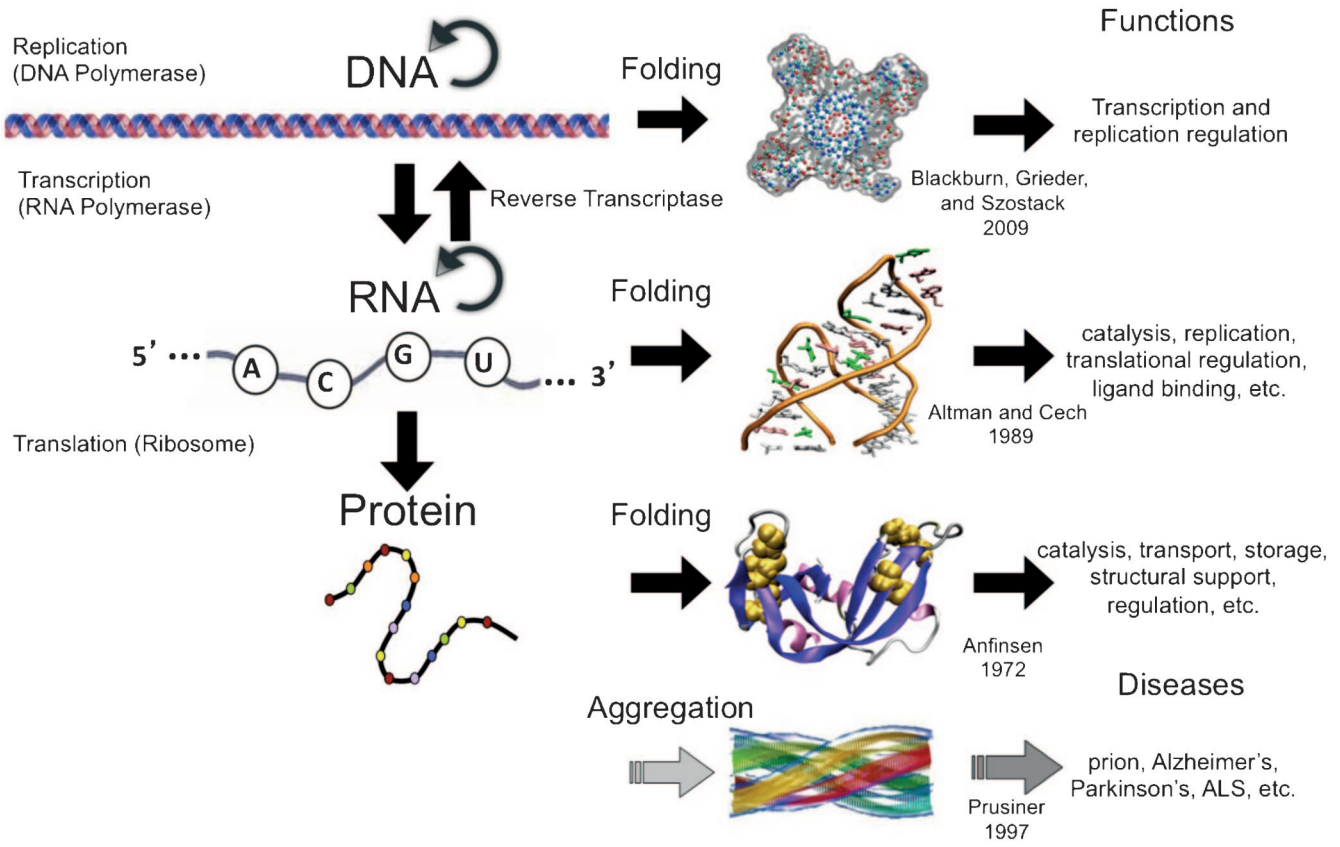


Figure 1. Central dogma of molecular biology in the context of biomolecular folding. In this extended version, we include DNA, RNA, and protein folding and some representative functions, as well as protein aggregation and associated diseases. Also listed are associated Nobel Prize laureate names with the years they received the award.

of RNA is known to be hierarchical in that the secondary structures generally form before tertiary structures,[28] although there are a few exceptions.[29–31] Based on these cursory observations, one might conclude that nucleic acid folding mechanisms and their corresponding energy landscapes are even simpler than proteins.

While DNA folding mechanisms are just starting to be explored, experimental and theoretical approaches are increasingly supporting the view that RNA folding mechanisms are actually more complex than those of proteins, often involving kinetically partitioned distinct parallel pathways and traps.[32] These seemingly counterintuitive observations can be rationalized by the structures of the possible nucleotides, which are largely similar with chemically identical ribose sugars and charged phosphates. The only difference between the nucleotides is in the four types of bases, which are all aromatic and roughly the same size, and the charged phosphate backbone, which has nonspecific electrostatic interactions. This is in striking contrast to the 20 natural amino acids of various sizes found in proteins, which can be not only aromatic but also polar, nonpolar, or charged. The level of specificity for the native basin inherent in protein folding mechanisms cannot exist for RNA molecules because their bases are deficient in variation, leading to more promiscuous folding mechanisms. In addition, while the bases in the four types of RNA nucleotides can hydrogen bond to one another in WC base pairings, only about half of RNA sequences are found to be WC base-paired, and many of the remaining RNA nucleotides form unpaired regions including bulges, loops, dangling ends, and other motifs.[33]

Taken together, the problem of nucleic acid folding mechanisms is likely to be more complex than protein folding. Indeed, even simple RNA hairpins and pseudoknots have complex folding kinetics.[30,34–37] However, the similar themes imply that the same formalism and concepts from protein folding can serve as a basis for the study of nucleic acid folding mechanisms.[32,38]

3 Bottom-Up and Top-Down Approaches for Biomolecular MD Simulations

MD simulations continue to be an indispensable tool for the study of biomolecular folding and assembly mechanisms. Although there are now many different flavors of MD simulations, we summarize two classes of approaches: empirical force fields and native structure–

Israel Journal of Chemistry

based MD simulations. An exhaustive description of these energy functions is beyond the scope of this review, and here we only discuss a basic description of the approaches. For empirical force field MD simulations, there exists a simple potential energy function that describes the biomolecule of interest that is defined by bonds, angles, dihedral, Urey–Bradley, out-of-plane improper, Lennard-Jones, and electrostatic terms:

$$
\begin{aligned}
E_{total} &= E_{backbone} + E_{nonbonded} \\
E_{backbone} &= \sum^{bonds} K_r(r - r_0)^2 + \sum^{angles} K_\theta(\theta - \theta_0)^2 \\
&+ \sum^{dihedrals} K_\phi^{(n)}[1 + \cos(n\phi - \delta))] \\
&+ \sum^{U-B} K_{UB}(S - S_0)^2 + \sum^{impropers} K_\omega(\omega - \omega_0)^2 \\
E_{nonbonded} &= \sum_{i<j-3}^{nonbondedpairs} \left\{ \varepsilon_1(i,j) \left[\left(\frac{\sigma_{i,j}^{nat}}{r_{i,j}} \right)^{12} - 2 \left(\frac{\sigma_{i,j}^{nat}}{r_{i,j}} \right)^{6} \right] + \frac{q_i q_j}{4\pi D r_{ij}} \right\}
\end{aligned}
\tag{1}
$$

In a "bottom-up" approach, each of these terms contains atomistic force field parameters that are typically obtained from quantum mechanical calculations of model compounds or empirical data when available. Based on the potential energy function for the molecule(s) of interest for which the parameters must be known, long MD simulations are performed for folding events to occur, and the energy landscape for folding would be an emergent phenomenon (Figure 2, left).

Popular atomistic empirical force field–based MD simulation programs include CHARMM,[39] AMBER,[40] and NAMD.[41] In these approaches, the partial charges of atoms are static and located at their centers but new polarizable force fields are being developed that aim to account for the electronic polarization of its environment. While these calculations are computationally more expensive than the simpler fixed-point charge models, they improve the accuracy of the electronic description of the biomolecular environment. Since nucleic acids are highly charged systems, polarizability is likely to become an important direction for empirical force field improvements

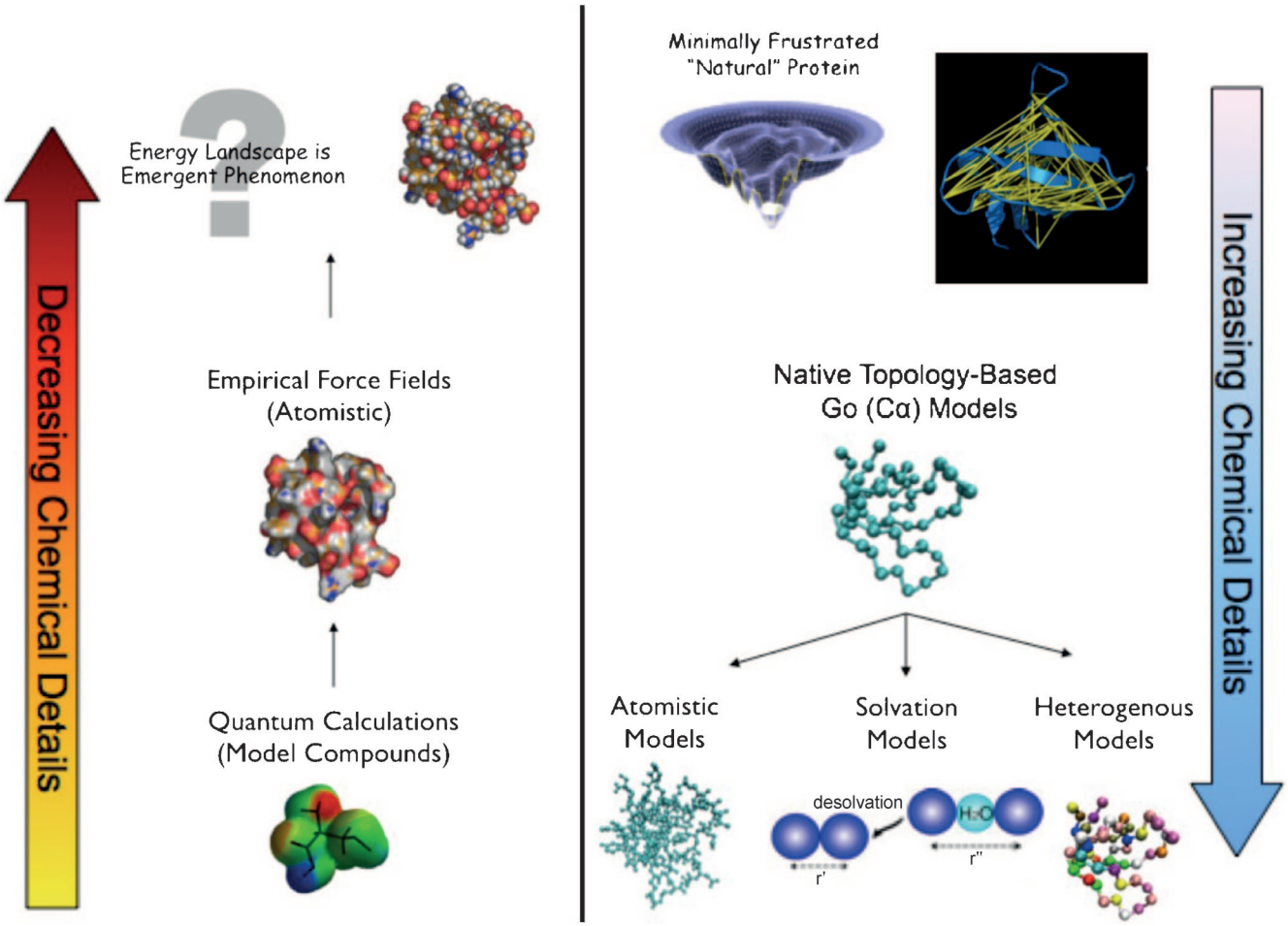


Figure 2. Comparison between the "bottom-up" empirical force field and the "top-down" native structure based MD simulation approaches for studying biomolecular folding mechanisms.

in DNA and RNA MD simulations, and they are now becoming available.[42]

The computational demands of the atomistic empirical force field MD simulations limit the size of biomolecular system that one can investigate. A nucleic acid is a highly solvated biomolecule in the presence of ions that play important roles in the stability and folding mechanisms. To perform MD simulations in a reasonable time while still capturing the folding events, advanced sampling methods such as replica exchange can reduce the time spent in trap states and increase conformational sampling at the expense of any kinetic information about the folding mechanism.[43] Other approaches include using advanced computing infrastructure such as the distributed computing Folding@Home[44] approach or the Anton supercomputer,[45] which are inaccessible to the average researcher. One can also coarse-grain biomolecules based on empirical force fields to increase timescales while reproducing basic structural order parameters.[46]

An independent, native structure–based approach for MD simulations assumes that the energy landscape is funneled or globally directed towards the native basin. The principle of minimal frustration states that natural biomolecules have been evolutionarily selected to have sequences whose interactions are favorable if they contribute to the native basin and all other conflicting interactions that would compete with non-native trap states are disfavored.[47] A minimally rugged funneled energy landscape results that is directed toward the native basin.[48,49]

A Go-type model uses an idealized and simplified representation where the energy landscape is perfectly funneled.[50] The potential energy function of an off-lattice Go-type model MD simulation is defined by bonds, angles, dihedral, and Lennard-Jones terms whose parameters correspond to the native structure such that its global minimum is the native basin:

$$\begin{aligned} E_{total} &= E_{backbone} + E_{nonbonded} \\ E_{backbone} &= \sum^{bonds} K_r(r - r_0)^2 + \sum^{angles} K_\theta(\theta - \theta_0)^2 \\ &+ \sum^{dihedrals} K_\phi^{(n)}[1 - \cos(n(\phi - \phi_0))] \\ E_{nonbonded} &= \sum_i \end{aligned} \quad (2)$$

The main difference in comparison to the empirical force field approach is that all of the parameters correspond to the native state as the energy minimum for each term in the Go-type model. Although the off-lattice Go-type model is typically represented as a coarse-grained one bead per residue model for proteins, other variants include all-atom representations,[51,52] ones with solvation models,[53,54] and ones with native interactions "flavored" by sequence identity[54,55] (Figure 2, right).

4 RNA Folding Mechanisms: Base-Stacking Interactions and Electrostatics

4.1 Overview

The major goal of any MD simulation approach is to accurately represent the physics of the biological system such that one can compare directly with experiments and inform them with higher-resolution predictions. As such, these MD simulation models must capture the important interactions that contribute significantly to the folding. Nucleic acids contain aromatic nucleotide bases that pair with other bases through hydrogen bonds and then stack on top of one another through π-stacking interactions. Also, phosphates line the backbone with charged electrostatic interactions with ions (Figure 3). For RNA molecules, the base-stacking and electrostatic interactions play pivotal roles in their folding mechanisms.

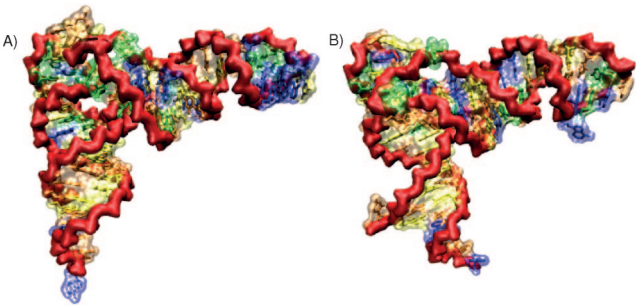


Figure 3. Structures of *Escherichia coli* A) $tRNA^{Phe}$ and B) $tRNA^{fMet}$. The phosphate backbone is shown in red and the rest of the RNA is shown by a transparent space-filled and opaque ball-and-stick representation and colored according to the nucleotide type.

4.2 Atomistic Empirical Force Field MD Simulations of RNA Folding

The nucleotide bases consist of aromatic moieties that have delocalized electrons in π orbitals. The stacking of bases on top of one another provides additional stability through quadrupole moments that stabilize the interaction between bases. It has been known for some time that the stacking interactions in CHARMM and AMBER are lower and higher than expected, respectively.[56,57] As a result, the conformational sampling of nucleic acid bases in CHARMM is greater than expected, while conformational sampling in AMBER is too stable. Both scenarios could lead to unphysical trap states in MD simulations. In a very recent study, Chen and Garcia optimized the van der Waals parameters of the AMBER force field and performed replica-exchange MD simulations to fold three RNA tetraloop hairpins with non-canonical loop-stabilizing interactions. The resulting structures were within about 1–3 Å of their experimentally determined X-ray crystallographic or NMR structures.[58] The im-

provements in the force field parameters could open the possibility of more accurate MD simulations of larger RNAs.

For nucleic acids, there is also a profound dependence of structure on salt conditions, due to the ionic interactions with the phosphate backbone. For DNA, a high salt concentration results in the A-DNA conformation and then a B-DNA conformation in physiological salt concentrations. The G-quadruplex structure contains a negatively charged channel that requires the presence of cations for stability.[59] For RNA folding, the presence of ions determines the overall tertiary structure.[60] In 2000, Cheatham and Kollman stated, "Study of the ionic strength and identity effects on biomolecular structure in MD simulations is the next frontier for MD simulation",[61] and the accuracy of ion interactions remains a significant challenge.[62]

4.3 Coarse-Grained Go-Type MD Simulations of RNA Folding

Hyeon and Thirumalai developed the three interaction site (TIS) model, a three bead per nucleotide model of RNA folding.[63] The three beads correspond to the base, sugar, and phosphate moieties of a nucleotide. In addition to the traditional Go-type long-range interactions for the bases, they included sequence-dependent base-stacking interactions using empirical Turner's rules[64] and nonspecific electrostatic interactions between the phosphate beads using the ion concentration–dependent Debye–Hückel potential. The TIS model MD simulations were applied to RNA hairpins, pseudoknots, and larger RNA, and directly compared to experimental observables. The intermediates observed in experiments and their melting temperatures are accurately reproduced in TIS model MD simulations.[30,31,36,63,65,66] In addition, the TIS model predicts: 1) for the telomerase RNA pseudoknot that the tertiary structure formation occurs before secondary structure is complete,[30] and 2) for p5abc that an intermediate exists in the folding mechanism where the secondary and tertiary structure formation are coupled.[31] The stabilities of the individual hairpins in TIS models of RNA are generally predicted on the basis of the base-stacking stabilities.[30]

The TIS model can also accurately predict the kinetics of RNA pseudoknot folding. Cho et al. performed TIS model temperature quench folding MD simulations and monitored the folding times and pathways for several H-type RNA pseudoknots. They observed parallel folding pathways that were kinetically partitioned between the folding of the individual substituent stems.[30] The folding times predicted by the TIS model were remarkably similar to the relaxation times measured by Ansari and co-workers in a subsequent laser temperature-jump perturbation study.[37] Inspired by interrupted multiple ion-jump single-molecule experiments,[67] Biyun et al. developed novel ion-concentration induced jump TIS model MD simulations by abruptly changing the ion concentration in the Debye–Hückel potential.[36]

Very recently, Li et al. performed TIS model simulations of four different tRNA molecules of similar length and tertiary structure but different sequence.[66] The similar length meant that the electrostatic interactions from the phosphates and the Go-type interactions, as well as the tertiary structure, were nearly identical. The main difference between the tRNA molecules in the TIS model was the sequence-dependent base-stacking interactions (Figure 3). In classical absorbance spectra experiments, Crothers and co-workers observed different melting profiles for different tRNA molecules, and the TIS model accurately predicted them for several tRNA.

For certain tRNA molecules, Li et al. also predicted the existence of premature unproductive folding mechanisms. These "backtracking" events were first observed in proteins by Clementi et al. in the original off-lattice Go-type MD simulations[50] and in other subsequent studies,[68–70] and they have been ascribed to topological frustration. In these cases, premature partial folding of a protein requires complete unfolding before proceeding to the folded state. For tRNA, the Ψ hairpin loop is the least stable hairpin in the tRNAs they studied and must unfold to reach the folded state.[66]

As with any theoretical representation of biomolecules, the TIS model has limits. In particular, it performs poorly in the limit of low ion concentration. When the ion concentration was tuned in the Debye–Hückel potential to less than 0.01 M $[Na^+]$ for tRNAs, they remained melted even at temperatures close to the freezing temperature of water, even though it is expected to fold at a significantly higher temperature.[66] Denesyuk and Thirumalai also observed ambiguity between experiments and simulations for concentrations greater than 0.2 M $[Na^+]$.[65]

A simpler Go-type coarse-grained MD simulation approach is the self-organized polymer (SOP) model.[71] This one bead per nucleotide/residue model consists of only two sets of terms: a FENE potential for chain connectivity and a Lennard-Jones potential for the long-range Go-type interactions. Recently, Lin et al. applied the SOP model to riboswitches, which are noncoding RNA that regulate gene expression by controlling transcription or translation through the specific binding of a metabolite. The metabolites themselves are comparable in size to one or more nucleotides. Representing a metabolite as one or more beads with a SOP model representation for RNA is sufficient to differentiate between force extension curves of the unliganded and liganded states.[72,73]

4.4 Atomistic Go-Type MD Simulations of RNA Folding

The Go-type energy function can be readily extended into an all-atom representation, such that all of the heavy atom interactions are globally directed to the native

basin. Whitford et al. performed all-atom Go-type MD simulations of the *S*-adenosylmethionine-1 (SAM-1) riboswitch and showed that the binding of SAM-1 stabilizes the P1 helix and reduces the free energy barrier for folding.[74] Feng et al. studied the queuosine anabolic intermediate $preQ_1$ riboswitch using an all-atom Go-type energy function. The $preQ_1$ riboswitch is largely unstructured in the absence of $preQ_1$. Interestingly, they observed premature folding of P2 that has to unfold or "backtrack" so that P1 can form and folding can proceed to completion.[75]

5 DNA Folding Mechanisms

5.1 Overview

The primary function of DNA is to contain hereditary information and pass it along to successive generations or to transcribe the information to RNA molecules. There have been numerous studies of MD simulations involving DNA due to its obvious importance in the cell, but many of the MD simulations to date have focused on the proteins that complex with DNA.

Many transcription regulation proteins bind to duplex DNA by neatly fitting into the major or minor grooves, and their binding mechanisms have been extensively studied using coarse-grained MD simulations. Generally, the electrostatic interactions are incorporated between the negatively charged DNA phosphate backbone and charged amino acids (i.e., positively charged Arg and Lys and negatively charged Asp and Glu).[76] Coarse-grained MD simulations have also been used to identify the three steps in which RNA polymerase partially melts and unzips DNA so that it can be transcribed by RNA polymerase into RNA.[77] Atomistic MD simulations of protein–DNA complexes have been largely restricted to the study of the local dynamics of the folded state.[78] However, partial unfolding of DNA through protein-mediated base flipping has been extensively studied using umbrella sampling to map out the energetic barriers using multiple empirical force fields.[56]

Generally, unlike RNA molecules that fold into one of many possible specific structures, the DNA molecules are often seen in MD simulations as static structures that are already folded into a duplex to carry out its function. Therefore, it is natural to presume that DNA folding mechanisms might have little biological relevance. However, there are numerous biological examples where already folded DNA duplexes assemble into complex structures including chromatin and viral genome assembly, and we direct readers to a very recent review of the development of Go-type coarse-grained MD simulations of duplex DNA.[79] Here, we focus on MD simulations of DNA G-quadruplex folding mechanisms.

5.2 DNA G-Quadruplex Folding Mechanisms

Guanine-rich sequences of DNA and RNA can fold into quaternary structures called G-quadruplexes, but we will focus on the DNA form here. DNA G-quadruplexes are of particular biological relevance because they are found at the telomeric regions at the end of DNA.[20] DNA G-quadruplexes have been known to fold in vitro in human telomeres with the repeats TTAGGG. This sequence is not recognized by the telomerase enzyme, which is upregulated in cancerous cells. G-quadruplexes have been an important anticancer target for promoting G-quadruplex formation and stabilization through the introduction of small molecules that recognize G-quadruplex structures to inhibit telomerase activity and induce apoptosis.[80,81]

G-quadruplexes consist of four guanines interacting through Hoogsteen hydrogen bonding to form a square planar tetrad, and two or three of these tetrads interact through base stacking to form a quadruplex. The G-quadruplex structure can involve one, two, or four strands of DNA. The G-quadruplex structure can consist of many different topologies that can be largely classified into parallel (all strands go in the same direction), antiparallel (half of the strands go in the opposite direction), or a mixed hybrid topology (three of the strands go in the same direction while one goes in the opposite), as shown in Figure 4. The formation of the G-quadruplex structure also requires dehydrated K^+ or Na^+ ions in a negatively charged channel formed by the centers of each tetrad.

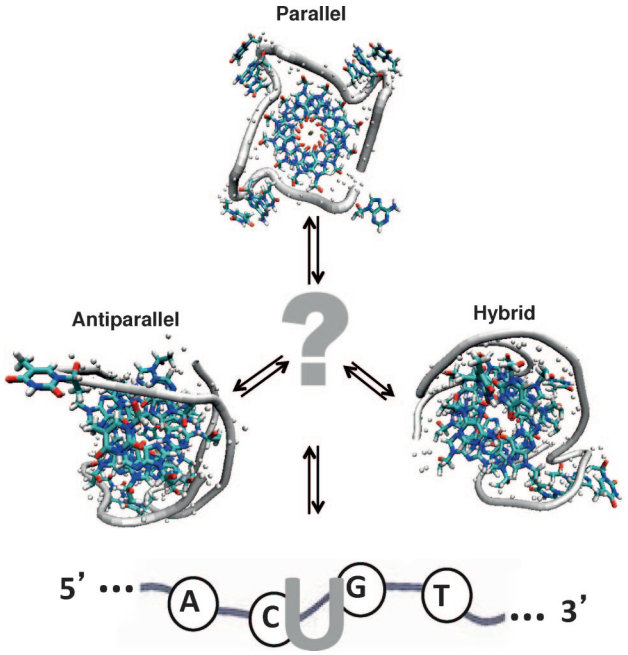


Figure 4. Schematic of DNA G-quadruplex folding from an unfolded state (U) to several different possible topologies through some intermediate state(s). The folding mechanism is thought to be complex and involve at least one intermediate state.

Israel Journal of Chemistry

The correct representation of the ions in G-quadruplex MD simulations is a challenge.[82]

Tremendous progress has been made in the experimental characterization of G-quadruplexes, but their folding mechanisms remain unsettled (Figure 4). Despite the simplicity of the G-quadruplex structures, their folding mechanisms involve intermediate states and complex kinetics. There are currently two proposed models of G-quadruplex folding. The first involves the stepwise addition of the strands to form a duplex, then a triplex, and finally a quadruplex to complete the folding process.[83] In the second model, FRET and magnetic tweezers experiments show that a partial disruption of a few base pair contacts is sufficient to completely unfold the quadruplex structure, which is inconsistent with a long-lived DNA triplex intermediate.[84,85] We note, however, that both mechanisms are possible and could simply be kinetically partitioned.

To date, MD simulations of G-quadruplexes have been restricted to atomistic empirical force field MD simulations. Due to their relatively small size (~20 nucleotides per three-tetrad G-quadruplex), G-quadruplex MD simulations have reached the microsecond timescale using fairly standard computational techniques. While this is an insufficient timescale for a complete energy-landscape characterization of G-quadruplexes, putative intermediates were identified.

Recently, Sponer and co-workers performed unfolding MD simulations of G-quadruplexes by starting from the folded structure and removing the ion.[105] They observed individual strand slippage that supports a triplex intermediate mechanism. They then performed subsequent MD simulations in excess ions, similar in spirit to stopped-flow experiments, and they observed structures moving closer to the folded state. While their study involved very few trajectories over a microsecond timescale, which is inadequate to characterize thermodynamically or kinetically, they observe "misfolded" arrangements that indicate a complex folding mechanism.

6 MD Simulations on the GPU Architecture

6.1 Overview

Over the past few years, graphics processing units (GPUs) have become a commonly used approach for performing MD simulations of biomolecular systems (Figure 5). For parallelizable algorithms such as those used in MD simulations, GPUs can in some cases result in a substantial speedup of their performances. GPUs are specialized hardware devices that are optimized for parallel execution of floating-point operations.[86,87]

In a heterogeneous CPU–GPU environment, a GPU device is connected to the CPU through a PCI express bus and consists of multiple cores that can be executed independently with a limited set memory. The single instruction, multiple data (SIMD) parallel programming paradigm is implemented by assigning multiple threads to execute the same single kernel instruction on multiple, distinct data. Parallel algorithms using the SIMD programming paradigm can be implemented using GPU-specific programming languages such as NVIDIA CUDA or OpenCL. CUDA is specific for NVIDIA GPUs while OpenCL is designed for cross-platform compatibility. As a result, CUDA has less overhead for GPU programming and is easier to implement, while the OpenCL programming language allows greater specific control that can result in greater performance even when OpenCL is implemented and optimized for NVIDIA GPUs.

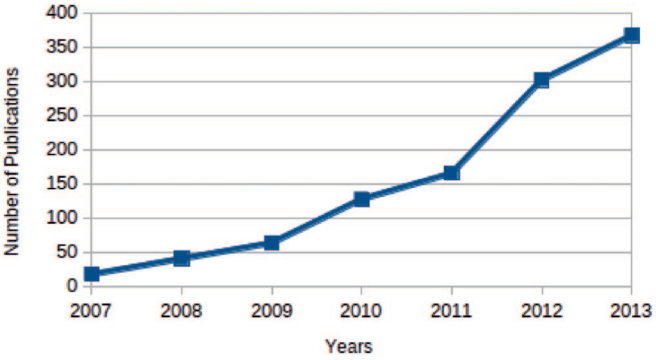


Figure 5. Number of publications with the words "MD simulations" and "GPU" in the title according to Google Scholar. General-purpose GPU programming became available in 2007.

The main goal of developing any algorithm for the GPU architecture is to isolate the parallelizable portion of the code that can be computed independently and recast the algorithm so that these calculations are performed in parallel on the GPU instead of serially on the CPU. Broadly, the basic MD simulation algorithm has two main portions: 1) the calculation of the forces between interacting particles, and 2) the update of the positions and velocities in the next timestep based on the forces in the current timestep. Since the update of the positions and velocities are dependent on the previous step, it is inherently serial and would not benefit from a parallel execution. However, the calculations of the forces between the particles, the main computational bottleneck of MD simulations, are independent of one another and therefore amenable to parallel execution. Kernel instructions for calculating the forces are executed for the different data points corresponding to each particle's position and velocity relative to every other interacting particle in the current timestep. For reasons we will state below, we must note that the porting of MD simulation codes to the GPU architecture is much more involved than just reprogramming the code in a different language. It requires the development of new algorithms that are optimized specifically for the GPU architecture.

Recently, several GPU-optimized atomistic and coarse-grained MD simulation codes have been developed and

implemented. For atomistic MD simulations, several well-known MD simulation codes that had been previously available for traditional CPUs are now available for GPUs, including NAMD,[88] AMBER,[89] and GROMACS.[90] Coarse-grained or general particle dynamics MD simulation codes have also been developed, and they include HOOMD-Blue,[91] LAMMPS,[92] OpenMM,[93] and SOP-GPU.[94] Here, we will largely focus on our own in-house coarse-grained MD simulation software for the GPU,[95–98] but many of the strategies and issues we discuss are general and applicable to all MD simulation codes on GPUs.

6.2 Performance Issues of MD Simulations on GPUs

While there are clear benefits to using a GPU for MD simulations, there also exist a number of performance and accuracy issues because the primary purpose of the GPU architecture is to render images. New advances, largely driven by the gaming market that drives down the prices of GPUs to make them economically viable for scientific applications, continue to remove or lower these barriers for the average researcher. As with all computer algorithms, there are temporal and spatial performance and accuracy issues related to implementing the MD simulation algorithm on the GPU architecture.

Since the GPU device is separate from the CPU, data involved in MD simulations must be transferred to the GPU for calculations to be performed. Currently, GPUs have a hardware memory limit of 1.5–12 GB, depending on the model, and most of the commercially available models have memory limits that are unfortunately closer to the bottom of that range. Furthermore, the most significant bottleneck of programming on GPUs is the transfer of information to the CPU and back across the PCI express bus. It is possible to optimize a program to hide the latency of the information transfer such that other calculations are performed while the transfer is occurring, but the optimizations are usually hardware specific and can vary between GPU models. While these latency-hiding approaches should be taken whenever possible, the performance differences between GPU models limit the generality of the optimization techniques. GPU programs must be optimized so that as little information as possible is transferred and to keep the information on the GPU for as long as possible to limit the number of information transfers. Many tried-and-true CPU strategies[99] such as keeping precomputed values in memory should be avoided because it may be faster to simply recompute the values over and over rather than accepting the cost of transferring the information. Also, the precomputed values would take up limited memory space.

Recently, we introduced two noteworthy novel GPU-optimized MD simulation algorithms based on well-known CPU versions, which we will highlight in the following sections. We stress that these algorithms are not just a trivial recasting of code from one programming language to another, but instead involve the development of new algorithms that are specifically optimized for the GPU architecture.

6.2.1 Parallel Verlet Neighbor List Algorithm

Since the computation of the forces between long-range interacting particles is the main performance bottleneck of MD simulations, many cutoff algorithms have been developed to reduce the computational demands of the force calculations. One of the most widely used for Lennard-Jones interactions is the Verlet neighbor list algorithm (Figure 6A).[100] In this approach, a subset list of interactions within a "skin" distance cutoff is generated and updated every n timesteps. Out of all of the members of the neighbor list, the forces are computed for only a further subset of interactions within a cutoff radius.

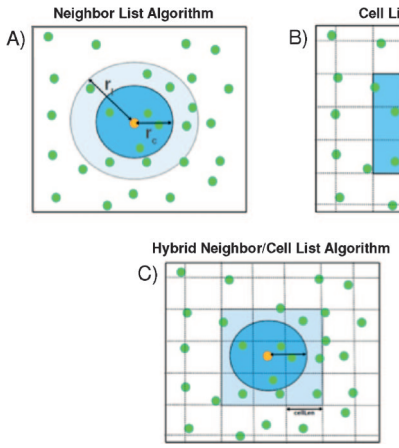


Figure 6. Truncation algorithms for MD simulations. A) Verlet neighbor list algorithm, B) cell list algorithm, and C) a hybrid algorithm where the outer layer is determined by the cell list algorithm and the inner pair list uses a distance cutoff like the Verlet neighbor list algorithm.

While the original Verlet neighbor list algorithm works well on a CPU, it is inherently serial because the generation of a subset list requires looping over all of the interactions and placing only the subset into a list, one after another. Lipscomb et al. developed a novel parallel Verlet neighbor list algorithm[95] where the distances between interacting particles are computed and sorted using a parallel algorithm such that those within the distance cutoff are placed at the top of the list. Only those interactions within the distance cutoff are copied, in parallel, to the subset list. A similar approach is used to generate a pair list of interactions as well. The parallel Verlet neighbor list algorithm involves more steps but can be performed using only parallel operations. As such, the entire algorithm can be performed on the GPU without any transfer of information between the CPU and GPU.

6.2.2 Parallel Hybrid Neighbor/Cell List Algorithm

Another commonly used MD simulation algorithm for interaction cutoffs is the cell list algorithm (Figure 6B).[99] In this algorithm, the simulation box is divided evenly into cubic "cells". The benefit of this approach over the Verlet neighbor list algorithm is that the relatively computationally expensive distance calculations between pairs of interacting particles need not be computed because only their positions need to be known to determine whether they are in neighboring cells. The forces are computed only for pairs of particles that are located in neighboring cells.

The cell list algorithm also works well on the CPU, and it is typically implemented using a linked list data structure. The implementation of this algorithm on the GPU would greatly limit the size of the simulation one could perform, because of the memory required to maintain the linked list. We instead implemented the algorithm on the GPU without the linked list data structure and using only parallel operations, but we observed performance that was significantly poorer than our implementation of the parallel Verlet neighbor list algorithm even though it could be performed entirely on the GPU.[95]

Previous studies have combined the neighbor and cell list algorithms into a hybrid version, and these demonstrated performance increases on the CPU beyond the individual algorithms alone.[101] Proctor et al. implemented a hybrid neighbor/cell list algorithm where a cell list was updated every n timesteps and the forces were computed for only those particles within a distance cutoff (Figure 6C).[97] Again, the algorithm was performed entirely on the GPU because it involves parallel operations only.

6.2.3 Performance Analysis: N-Dependent Speedup

In our GPU-optimized coarse-grained MD simulation code, we implemented the parallel Verlet neighbor list[95] and the parallel hybrid neighbor/cell list[97] algorithms. We compared the performance to an equivalent CPU-optimized code with the Verlet neighbor list algorithm and compared execution times of MD simulations of systems that range in size from a tRNA (76 beads) to the 70S ribosome (10,219 beads). We note that the CPU code was developed such that it was optimal for the CPU architecture. For example, precomputed values were used extensively in the code, as one would expect in code that was appropriate for the CPU-only architecture.

For the Verlet neighbor list algorithm, we observed that for the smallest system it required more time to perform the MD simulation on a GPU than on a CPU. The cost of transferring information to the GPU was not worth the gain of performing the force calculations on the GPU in parallel. On the other hand, for the largest system, we observed approximately 30× speedup over the CPU code.[95]

We then compared our code to HOOMD, one of the leading GPU-optimized general particle dynamics software available. This is not a particularly fair comparison because the HOOMD software is designed to be flexible and our code was developed specifically for Go-type MD simulations. However, the comparison is valuable to quantify the performance of our MD simulation code relative to one of the leading GPU-optimized MD simulation codes. We implemented the SOP model energy function into HOOMD, and we observed that HOOMD has a faster execution time for systems that are less than around 500 beads. For systems that contain about 500–1000 beads, our code has a faster execution time. After that, HOOMD, due to the memory overhead associated with its flexibility, cannot perform MD simulations for larger systems, but our code can accommodate a maximum of ~80,000 beads for GPUs with 1.5 GB of memory.[96]

When we compared our parallel Verlet neighbor list algorithm MD simulation code to the same code using instead the parallel hybrid neighbor/cell list algorithm, we observed about 10% speedup for the systems we studied, if we excluded the smallest system with only 76 beads.[97]

6.3 Accuracy Issues: Single vs. Double Precision

Another significant issue has to do with the accuracies of the MD simulation calculations themselves. Since the original purpose of GPUs was to accelerate the floating-point operations involved in rendering graphical images that did not need to be accurate, the floating-point operations in older GPU models were not IEEE compliant to speed up calculations. As a result, MD simulations implemented on some of the early NVIDIA Tesla model GPUs exhibited an energy drift,[102] indicating that detailed balance was not preserved due to errors in the calculations. More recent NVIDIA GPU models are now IEEE compliant so this is no longer an issue.[95] However, unlike traditional CPUs, the GPU architecture has a significant performance difference between single-precision and double-precision calculations.[86,95,103] It is usually standard to implement MD simulations on CPUs using double-precision calculations, because the performance difference is negligible.

Since the use of double-precision calculations to perform MD simulations is an arbitrary goalpost for accuracy that is convenient for computer architecture, we generated single- and double-precision versions of our code to quantify the difference. Using the recent Tesla models (and more recent Kepler models; data not published), we observed no significant energy drift that was observed in previous studies. To quantify the significance of the difference between the single- and double-precision calculations, we ran independent MD simulations using single- and double-precision calculations starting from the same structure and the same velocities. As with any calcula-

tions, round-off errors do occur and for parallel programs the order in which the calculations are performed can result in different numbers. We then computed order parameters that could be directly compared to experiments, namely the radius of gyration that could be compared to SAXS experiments, and end-to-end distances that could be directly compared to single-molecule FRET experiments. We observed minimal differences between the single- and double-precision coarse-grained MD simulations, at least at the resolution of these experimentally relevant biophysical metrics.[95]

In a recent study, Walker and co-workers developed a mixed single/double precision approach for performing MD simulations in AMBER, and they applied it to generalized Born implicit solvent and explicit solvent particle mesh Ewald MD simulations. They observed minimal energy drift and minimal differences in RMSD and RMSF while still retaining similar performance to the single-precision-only approach.[103] If it turns out that MD simulations are indeed necessarily sensitive to the precision of the calculations, such precision optimization techniques may be necessary to obtain the accurate MD simulation trajectories while retaining performance benefits as well. However, even when performing MD simulations on the GPU using double-precision calculations only, a significant performance gain can be observed.[95,103]

7 Summary and Outlook

MD simulations continue to be a critical tool to advance experiments and theory. While many of the main features and even the finer details of protein folding theory are becoming settled, our understanding of nucleic acid folding mechanisms continue to be developed and MD simulations are a key tool.

In our review, we have focused on empirical force field and Go-type MD simulations, and we have highlighted their successes in reproducing experiments and outlined their limitations. Specifically, we highlighted recent advances in atomistic empirical force field MD simulations for refining parameters to accurately characterize base-stacking interactions. For TIS model MD simulations that use empirical Turner's rules for base-stacking interactions, RNA folding can reproduce thermodynamic and kinetic features of the folding mechanism. The accurate electrostatic representation in nucleic acids, however, continues to be a challenge for both approaches, which may be improved with the development of polarizable force fields. In both empirical force field and Go-type MD simulations, a complex folding mechanism for folding is observed. The chemical simplicity of RNA leads to promiscuous folding mechanisms arising from non-specific interactions such as electrostatics. We discussed above cases where this was true for both RNA and DNA.

In addition to physical advances for MD simulations, computational advances continue to play a fundamental role in describing folding and assembly mechanisms. Currently, the state-of-the-art atomistic empirical force field MD simulations of folding and assembly mechanisms are typically performed for systems that are about 100 residues/nucleotides in length for timescales of about a few microseconds. While these advances in computation are orders of magnitude better with respect to the time and length scales than the first MD simulations of a biomolecule, we feel that it is important to take a step back and also appreciate that we still have significant computational challenges ahead. Indeed, in a standard biology textbook, we found that the proteins and nucleic acids discussed are generally far larger in length than 100 residues/nucleotides (Figure 7). The development of coarse-grained MD simulation approaches, such as the Go-type model, will likely continue to play a key role in MD simulations of biologically relevant systems.

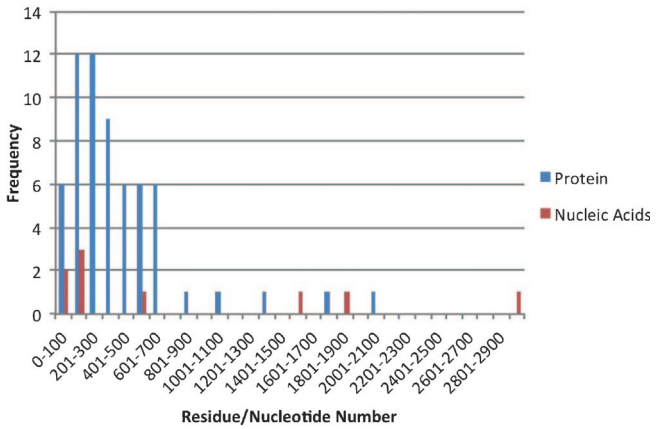


Figure 7. A histogram of the number of residues or nucleotides present in proteins and nucleic acids mentioned in an undergraduate-level biology textbook.[104] The approximate lengths were obtained from the PDB when the full-length structure was available or from other sources.

Recent work with performing MD simulations on the GPU architecture has already demonstrated that the parallel architecture lends itself well to MD simulations. The implementation of MD simulations on the GPU architecture will require new algorithms and we have discussed two novel ones that our group has developed. There remain issues with GPU performance and accuracy that are largely being addressed, and the field is becoming mature and accessible enough for the average researcher. The development of MD simulation algorithms continues to be a promising direction.

In light of the great advances we have seen over the past few decades, we are optimistic that the continued rapid expansion of computational methods will continue to advance our knowledge of biomolecular folding mech-

anisms. The development and application of MD simulations have always been interdisciplinary in nature, and recent developments indicate that significant contributions from biophysics and computer science will be required to advance the field of biomolecular folding mechanisms.

Acknowledgements

This work was supported by the National Science Foundation (CBET-1232724). S.S.C. acknowledges financial support from the Wake Forest University Center for Molecular Communication and Signaling. A.T.G. acknowledges financial support from the Wake Forest University URECA Research Fellowship. J.G. is a recipient of the Guy T. Carswell Scholarship. S.S.C. is thankful for fruitful discussions with H. J. Simpson.

References

[1] F. H. Crick, *Symp. Soc. Exp. Biol.* **1958**, *12*, 138.
[2] G. Storz, *Science* **2002**, *296*, 1260.
[3] S. B. Prusiner, *Proc. Natl. Acad. Sci. U.S.A.* **1998**, *95*, 13363.
[4] F. Chiti, N. Taddei, P. M. White, M. Bucciantini, F. Magherini, M. Stefani, C. M. Dobson, *Nat. Struct. Biol.* **1999**, *6*, 1005.
[5] J. A. McCammon, B. R. Gelin, M. Karplus, *Nature* **1977**, *267*, 585.
[6] P. G. Wolynes, J. N. Onuchic, D. Thirumalai, *Science* **1995**, *267*, 1619.
[7] C. B. Anfinsen, *Science* **1973**, *181*, 223.
[8] C. Anfinsen, H. Scheraga, *Adv. Protein Chem.* **1974**, *29*, 205.
[9] C. Levinthal, in *Mossbauer Spectroscopy in Biological Systems* (Eds.: P. Debrunner, J. Tsibris, E. Munck), University of Illinois Press, Urbana, **1969**, p. 22.
[10] J. N. Onuchic, P. G. Wolynes, *Curr. Opin. Struct. Biol.* **2004**, *14*, 70.
[11] M. Oliveberg, P. G. Wolynes, *Q. Rev. Biophys.* **2005**, *38*, 245.
[12] J. E. Shea, C. L. Brooks, *Annu. Rev. Phys. Chem.* **2001**, *52*, 499.
[13] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, H. S. Chan, *Protein Sci.* **1995**, *4*, 561.
[14] K. A. Dill, S. B. Ozkan, M. S. Shell, T. R. Weikl, *Annu. Rev. Biophys.* **2008**, *37*, 289.
[15] K. A. Dill, J. L. MacCallum, *Science* **2012**, *338*, 1042.
[16] P. G. Wolynes, W. A. Eaton, A. R. Fersht, *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, 17770.
[17] M. Gruebele, D. Thirumalai, *J. Chem. Phys.* **2013**, *139*, 121701.
[18] D. H. Turner, *Biopolymers* **2013**, *99*, 1097.
[19] J. A. Doudna, T. R. Cech, *Nature* **2002**, *418*, 222.
[20] C. W. Greider, E. H. Blackburn, *Cell* **1985**, *43*, 405.
[21] P. Hazel, J. Huppert, S. Balasubramanian, S. Neidle, *J. Am. Chem. Soc.* **2004**, *126*, 16405.
[22] R. D. Gray, J. B. Chaires, *Nucleic Acids Res.* **2008**, *36*, 4191.
[23] X. Cang, J. Šponer, I. Cheatham, *J. Am. Chem. Soc.* **2011**, *133*, 14270.
[24] J. A. Cruz, E. Westhof, *Cell* **2009**, *136*, 604.
[25] J. Pan, D. Thirumalai, S. A. Woodson, *J. Mol. Biol.* **1997**, *273*, 7.
[26] D. L. Pincus, S. S. Cho, C. Hyeon, D. Thirumalai, *Prog. Mol. Biol. Transl. Sci.* **2008**, *84*, 203.
[27] M. H. Bailor, A. M. Mustoe, C. L. Brooks III, H. M. Al-Hashimi, *Curr. Opin. Struct. Biol.* **2011**, *21*, 296.
[28] I. Tinoco, C. Bustamante, *J. Mol. Biol.* **1999**, *293*, 271.
[29] M. Wu, I. Tinoco, *Proc. Natl. Acad. Sci. U.S.A.* **1998**, *95*, 11555.
[30] S. S. Cho, D. L. Pincus, D. Thirumalai, *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106*, 17349.
[31] E. Koculi, S. S. Cho, R. Desai, D. Thirumalai, S. A. Woodson, *Nucleic Acids Res.* **2012**, *40*, 1–10.
[32] D. Thirumalai, D. K. Klimov, S. A. Woodson, *Theor. Chem. Acc.* **1997**, *96*, 14.
[33] R. I. Dima, C. Hyeon, D. Thirumalai, *J. Mol. Biol.* **2005**, *347*, 53.
[34] S. Cao, S. J. Chen, *J. Mol. Biol.* **2007**, *367*, 909.
[35] C. Hyeon, D. Thirumalai, *J. Am. Chem. Soc.* **2008**, *130*, 1538.
[36] S. Biyun, S. S. Cho, D. Thirumalai, *J. Am. Chem. Soc.* **2011**, *133*, 20634.
[37] R. Narayanan, Y. Velmurugu, S. V. Kuznetsov, A. Ansari, *J. Am. Chem. Soc.* **2011**, *133*, 18767.
[38] D. Thirumalai, C. Hyeon, *Biochemistry* **2005**, *44*, 4957.
[39] A. D. MacKerell, M. Feig, C. L. Brooks, *J. Am. Chem. Soc.* **2004**, *126*, 698.
[40] P. K. Weiner, P. A. Kollman, *J. Comput. Chem.* **1981**, *2*, 287.
[41] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, K. Schulten, *J. Comput. Chem.* **2005**, *26*, 1781.
[42] C. M. Baker, V. M. Anisimov, A. D. MacKerell, *J. Phys. Chem. B* **2011**, *115*, 580.
[43] Y. Sugita, Y. Okamoto, *Chem. Phys. Lett.* **1999**, *314*, 141.
[44] V. A. Voelz, G. R. Bowman, K. Beauchamp, V. S. Pande, *J. Am. Chem. Soc.* **2010**, *132*, 1526.
[45] R. O. Dror, R. M. Dirks, J. P. Grossman, H. Xu, D. E. Shaw, *Annu. Rev. Biophys.* **2012**, *41*, 429.
[46] A. Morriss-Andrews, J. Rottler, S. S. Plotkin, *J. Chem. Phys.*, **2010**, *132*, 035105.
[47] J. D. Bryngelson, P. G. Wolynes, *Proc. Natl. Acad. Sci. U.S.A.* **1987**, *84*, 7524.
[48] P. E. Leopold, M. Montal, J. N. Onuchic, *Proc. Natl. Acad. Sci. U.S.A.* **1992**, *89*, 8721.
[49] J. D. Bryngelson, J. N. Onuchic, N. D. Socci, P. G. Wolynes, *Proteins* **1995**, *21*, 167.
[50] C. Clementi, H. Nymeyer, J. N. Onuchic, *J. Mol. Biol.* **2000**, *298*, 937.
[51] L. Li, E. I. Shakhnovich, *Proc. Natl. Acad. Sci. U.S.A.* **2001**, *98*, 13014.
[52] P. C. Whitford, J. K. Noel, S. Gosavi, A. Schug, K. Y. Sanbonmatsu, J. N. Onuchic, *Proteins: Struct., Funct., Bioinf.* **2009**, *75*, 430.
[53] M. S. Cheung, A. E. Garcia, J. N. Onuchic, *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 685.
[54] J. Karanicolas, C. L. Brooks, *Protein Sci.* **2002**, *11*, 2351.
[55] S. S. Cho, Y. Levy, P. G. Wolynes, *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106*, 434.
[56] U. D. Priyakumar, A. D. MacKerell, *J. Chem. Theory Comput.* **2006**, *2*, 187.
[57] K. B. Hall, *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110*, 16706.

[58] A. A. Chen, A. E. García, *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110*, 16820.
[59] C. C. Hardin, M. J. Corregan, D. V. Lieberman, B. A. Brown, *Biochemistry* **1997**, *36*, 15428.
[60] D. E. Draper, *RNA* **2004**, *10*, 335.
[61] T. E. Cheatham III, P. A. Kollman, *Annu. Rev. Phys. Chem.* **2000**, *51*, 435.
[62] E. Fadrná, N. Špačková, R. Štefl, J. Koča, T. E. Cheatham, J. Šponer, *Biophys. J.* **2004**, *87*, 227.
[63] C. Hyeon, D. Thirumalai, *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6789.
[64] D. H. Mathews, J. Sabina, M. Zuker, D. H. Turner, *J. Mol. Biol.* **1999**, *288*, 911.
[65] N. A. Denesyuk, D. Thirumalai, *J. Phys. Chem. B* **2013**, *117*, 4901.
[66] R. Li, H. W. Ge, S. S. Cho, *J. Phys. Chem. B* **2013**, *117*, 12943.
[67] X. Qu, G. J. Smith, K. T. Lee, T. R. Sosnick, T. Pan, N. F. Scherer, *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105*, 6602.
[68] S. Gosavi, L. L. Chavez, P. A. Jennings, J. N. Onuchic, *J. Mol. Biol.* **2006**, *357*, 986.
[69] R. D. Hills Jr, C. L. Brooks III, *J. Mol. Biol.* **2008**, *382*, 485.
[70] R. D. Hills Jr, S. V. Kathuria, L. A. Wallace, I. J. Day, C. L. Brooks III, C. R. Matthews, *J. Mol. Biol.* **2010**, *398*, 332.
[71] C. Hyeon, R. I. Dima, D. Thirumalai, *Structure* **2006**, *14*, 1633.
[72] J.-C. Lin, D. Thirumalai, *J. Am. Chem. Soc.* **2008**, *130*, 14080.
[73] J.-C. Lin, D. Thirumalai, *J. Am. Chem. Soc.* **2013**, *135*, 16641.
[74] P. C. Whitford, A. Schug, J. Saunders, S. P. Hennelly, J. N. Onuchic, K. Y. Sanbonmatsu, *Biophys. J.* **2009**, *96*, L7.
[75] J. Feng, N. G. Walter, C. L. Brooks, *J. Am. Chem. Soc.* **2011**, *133*, 4196.
[76] A. Marcovitz, Y. Levy, *Proc. Natl. Acad. Sci. U.S.A.* **2011**, *108*, 17957.
[77] J. Chen, S. A. Darst, D. Thirumalai, *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107*, 12523.
[78] A. D. MacKerell Jr., L. Nilsson, *Curr. Opin. Struct. Biol.* **2008**, *18*, 194.
[79] D. A. Potoyan, A. Savelyev, G. A. Papoian, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2013**, *3*, 69.
[80] A. M. Burger, F. Dai, C. M. Schultes, A. P. Reszka, M. J. Moore, J. A. Double, S. Neidle, *Cancer Res.* **2005**, *65*, 1489.
[81] L. Rao, J. D. Dworkin, W. E. Nell, U. Bierbach, *J. Phys. Chem. B* **2011**, *115*, 13701.
[82] N. Spacková, I. Berger, J. Sponer, *J. Am. Chem. Soc.* **2001**, *123*, 3295.
[83] A. Y. Q. Zhang, S. Balasubramanian, *J. Am. Chem. Soc.* **2012**, *134*, 19297.
[84] M. De Messieres, J.-C. Chang, B. Brawn-Cinani, A. La Porta, *Phys. Rev. Lett.* **2012**, *109*, 058101.
[85] X. Long, J. W. Parks, C. R. Bagshaw, M. D. Stone, *Nucleic Acids Res.* **2013**, *41*, 2746.
[86] D. B. Kirk, W. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, 1st edition, Morgan Kaufmann, Burlington, MA, **2010**.
[87] J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, 1st edition, Addison-Wesley Professional, Upper Saddle River, NJ, **2010**.
[88] J. E. Stone, D. J. Hardy, I. S. Ufimtsev, K. Schulten, *J. Mol. Graphics Modell.* **2010**, *29*, 116.
[89] A. W. Götz, M. J. Williamson, D. Xu, D. Poole, S. Le Grand, R. C. Walker, *J. Chem. Theory Comput.* **2012**, *8*, 1542.
[90] S. Páll, B. Hess, *Comput. Phys. Commun.* **2013**, *184*, 2641.
[91] J. A. Anderson, C. D. Lorenz, A. Travesset, *J. Comput. Phys.* **2008**, *227*, 5342.
[92] S. Plimpton, B. Hendrickson, *J. Comput. Chem.* **1996**, *17*, 326.
[93] P. Eastman, V. S. Pande, *J. Comput. Chem.* **2010**, *31*, 1268.
[94] A. Zhmurov, R. I. Dima, Y. Kholodov, V. Barsegov, *Proteins: Struct., Funct., Bioinf.* **2010**, *78*, 2984.
[95] T. J. Lipscomb, A. Zou, S. S. Cho, in *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine (BCB '12)*, Orlando, FL, October 8–10, **2012**, ACM, New York, pp. 321–328.
[96] A. J. Proctor, T. J. Lipscomb, A. Zou, J. A. Anderson, S. S. Cho, in *2012 ASE/IEEE International Conference on BioMedical Computing (BioMedCom)*, Alexandria, VA, December 14–16, **2012**, pp. 14–19.
[97] A. J. Proctor, C. A. Stevens, S. S. Cho, in *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (BCB '13)*, Washington, DC, September 22–25, **2013**, ACM, New York, pp. 633–640.
[98] R. Li, R. Chen, P. Chen, Y. Wen, P. C. Ke, S. S. Cho, *J. Phys. Chem. B* **2013**, *117*, 13451.
[99] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press Inc., New York, **1989**.
[100] L. Verlet, *Phys. Rev.* **1967**, *159*, 98.
[101] W. Mattson, B. M. Rice, *Comput. Phys. Commun.* **1999**, *119*, 135.
[102] B. A. Bauer, J. E. Davis, M. Taufer, S. Patel, *J. Comput. Chem.* **2011**, *32*, 375.
[103] S. Le Grand, A. W. Götz, R. C. Walker, *Comput. Phys. Commun.* **2013**, *184*, 374.
[104] D. E. Sadava, D. M. Hillis, H. C. Heller, M. Berenbaum, *Life: The Science of Biology*, Sinauer Associates, Inc., Gordonsville, VA, **2011**.
[105] P. Stadlbauer, M. Krepl, T. E. Cheatham 3rd, J. Koča, J. Šponer, *Nucleic Acids Res.* **2013**, *41*, 7128–7143.