STARS

University of Central Florida
STARS

Electronic Theses and Dissertations, 2004-2019

2015

The Relationship Between DNA's Physical Properties and the DNA Molecule's Harmonic Signature, and Related Motion in Water--A Computational Investigation

Victor Boyer
University of Central Florida

 Part of the Industrial Engineering Commons
Find similar works at: https://stars.library.ucf.edu/etd
University of Central Florida Libraries http://library.ucf.edu

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
Boyer, Victor, "The Relationship Between DNA's Physical Properties and the DNA Molecule's Harmonic Signature, and Related Motion in Water--A Computational Investigation" (2015). *Electronic Theses and Dissertations, 2004-2019*. 1447.
https://stars.library.ucf.edu/etd/1447

THE RELATIONSHIP BETWEEN DNA'S PHYSICAL PROPERTIES AND THE DNA MOLECULE'S HARMONIC SIGNATURE, AND RELATED MOTION IN WATER— A COMPUTATIONAL INVESTIGATION

by

VICTOR M. BOYER
B.S. University of Central Florida, 2007
M.S.I.E. University of Central Florida, 2009

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2015

Major Professor: Michael D. Proctor

ABSTRACT

This research investigates through computational methods whether the physical properties of DNA contribute to its harmonic signature, the uniqueness of that signature if present, and motion of the DNA molecule in water. When DNA is solvated in water at normal 'room temperature', it experiences a natural vibration due to the Brownian motion of the particles in the water colliding with the DNA. The null hypothesis is that there is no evidence to suggest a relationship between DNA's motion and strand length, while the alternative hypothesis is that there is evidence to suggest a relationship between DNA's vibrational motion and strand length. In a similar vein to the first hypothesis, a second hypothesis posits that DNA's vibrational motion may be dependent on strand content. The nature of this relationship, whether linear, exponential, logarithmic or non-continuous is not hypothesized by this research but will be discovered by testing if there is evidence to suggest a relationship between DNA's motion and strand length. The research also aims to discover whether the motion of DNA, when it varies by strand length and/or content, is sufficiently unique to allow that DNA to be identified in the absence of foreknowledge of the type of DNA that is present in a manner similar to a signature. If there is evidence to suggest that there is a uniqueness in DNA's vibrational motion under varying DNA strand content or length, then additional experimentation will be needed to determine whether these variances are unique across small changes as well as large changes, or large changes only. Finally, the question of whether it might be possible to identify a strand of unique DNA by base pair configuration solely from its vibrational signature, or if not, whether it might be possible to identify changes existing inside of a known DNA strand (such as a corruption, transposition or mutational error) is explored. Given the computational approach to

this research, the NAMD simulation package (released by the Theoretical and Computational Biophysics Group at the University of Illinois at Urbana-Champaign) with the CHARMM force field would be the most appropriate set of tools for this investigation (Phillips et al., 2005), and will therefore be the toolset used in this research. For visualization and manipulation of model data, the VMD (Visual Molecular Dynamics) package will be employed. Further, these tools may be optimized and/or be aware of nucleic acid structures, and are free. These tools appear to be sufficient for this task, with validated fidelity of the simulation to provide vibrational and pressure profile data that could be analyzed; sufficient capabilities to do what is being asked of it; speed, so that runs can be done in a reasonable period of time (weeks versus months); and parallelizability, so that the tool could be run over a clustered network of computers dedicated to the task to increase the speed and capacity of the simulations. The computer cluster enabled analysis of 30,000 to 40,000 atom systems spending more than 410,000 CPU computational hours of hundreds of nano second duration, experimental runs each sampled 500,000 times with two-femtosecond "frames."

Using Fourier transforms of run pressure readings into frequencies, the simulation investigation could not reject the null hypotheses that the frequencies observed in the system runs are independent on the DNA strand length or content being studied. To be clear, frequency variations were present in the in silicon replications of the DNA in ionized solutions, but we were unable to conclude that those variations were not due to other system factors. There were several tests employed to determine alternative factors that caused these variations. Chief among the factors is the possibility that the water box itself is the source of a large amount of vibrational noise that makes it difficult or impossible with the tools that we had at our disposal to isolate any

signals emitted by the DNA strands. Assuming the water-box itself was a source of large amounts of vibrational noise, an emergent hypothesis was generated and additional post-hoc testing was undertaken to attempt to isolate and then filter the water box noise from the rest of the system frequencies. With conclusive results we found that the water box is responsible for the majority of the signals being recorded, resulting in very low signal amplitudes from the DNA molecules themselves. Using these low signal amplitudes being emitted by the DNA, we could not be conclusively uniquely associate either DNA length or content with the remaining observed frequencies. A brief look at a future possible isolation technique, wavelet analysis, was conducted. Finally, because these results are dependent on the tools at our disposal and hence by no means conclusive, suggestions for future research to expand on and further test these hypothesis are made in the final chapter.

ACKNOWLEDGMENTS

I would be remiss without acknowledging those individuals that made, either directly or indrectly, this dissertation possible. My advisor and now friend Dr. Michael Proctor's contagious enthusiasm and patient guidance has seen me through this sometimes arduous and confusing process. For that, I am most grateful. I also owe a debt of gratitude to my father, editor, fellow engineer, and patient sounding board, Victor Boyer, Sr., for providing not only editorial assistance, but clarity of thought during those times when the next steps seemed a step too far. Finally, to Dr. Steven Childers, my friend and confidante, thank you for your patient cheerleading and mentoring of me during this research.

TABLE OF CONTENTS

LIST OF FIGURES x

LIST OF TABLES xiii

CHAPTER 1: INTRODUCTION 1

Chapter 1 Abstract 1

Chapter Overview and Motivation for DNA Research 1

History of DNA Research 2

The DNA Molecule 6

The Physics of DNA 7

DNA and Vibrations 7

DNA Dynamics 9

Potential Impacts and Motivation 10

The Case for a Systems-View of DNA Vibrations 11

CHAPTER 2: FROM MOLECULAR MODELS TOWARD COMPUTATIONAL BIOLOGY 12

Chapter 2 Abstract 12

Molecular Modeling: Where We Came From 12

History 12

Enabling Works (Or: Biology is Basically Chemistry) 13

Computational Chemistry: Where We Are 16
Development of Molecular Dynamics Models (Or: Chemistry is Basically Math) . 16
The State of the Art 17
Computational Biology: Where We Are Going 25
Towards a Systems View (Or: Biology is Basically Math) 25
Interactions Are Key 26
Please, Mind the Gap (Between the Present and the Future) 27
Conclusions 29
CHAPTER 3: RESEARCH HYPOTHESES, METHODOLOGY, AND MODEL SELECTION 31
Chapter 3 Abstract 31
Research Hypotheses 31
Hypothesis 1: DNA's vibrational motion is dependent on strand length 32
Hypothesis 2: DNA's vibrational motion is dependent on strand content 33
Hypothesis 3: DNA's vibrational motion uniquely varies with strand length and/or content combinations 34
Hypothesis 4: DNA's vibrational motion forms a mathematical relationship 34
Simulation Tool Selection 35
CHARMM 37

GROMOS 39

Amber 41

Selection Rationale 42

Research Methodology 43

Experimental Design to Test Hypothesis 1 45

Experimental Design to Test Hypothesis 2 47

Experimental Design to Test Hypothesis 3 47

Experimental Design to Test Hypothesis 4 48

Statistical Methodology for Testing Hypotheses 1 and 2 48

Statistical Methodology for Testing Hypotheses 3 and 4 49

Data Collection Methodology 50

Experimental Predictions 51

CHAPTER 4: RESULTS 52

Chapter 4 Abstract 52

Experiment 1: Test of Hypothesis One: DNA's Vibrational Motion is Dependent on Strand Length 53

Statistical Analysis for Experiment #1 55

Experiment 2: Test of Hypothesis 2: DNA's Vibrational Motion is Dependent on Strand Content 67

Statistical Analysis for Experiment #2 67

Experiment 3: Test of Hypothesis 3: DNA's Vibrational Motion Uniquely Varies with Strand Length and/or Content Combinations 74

Experiment 4: Test of Hypothesis 4: DNA's Vibrational Motion Forms a Mathematical Relationship 78

An Emergent Hypothesis: Water Box as the Source of Noise..... 85

A New Alternative: Wavelet Analysis..... 91

CHAPTER 5: CONCLUSION 96

Chapter 5 Abstract 96

Summary..... 96

Conclusions..... 98

Experimental Limitations..... 100

Lessons..... 101

Parting Thoughts and Future Research..... 102

APPENDIX A: EXPERIMENT 2 COMPARISON OUTPUT TABLES 104

APPENDIX B: EXPERIMENT 4 FOURIER FITTING EQUATIONS 112

APPENDIX C: CLUSTER SPECIFICATIONS 119

APPENDIX D: MODEL CONSTRUCTION..... 121

APPENDIX E: NAMD SIMULATION PARAMETER FILES 131

APPENDIX F: PRESSUREPARSER TOOL 142

APPENDIX G: EXCEL PRESSURE MATCHING TOOL 154

APPENDIX H: MATLAB PROCESSOR FILES 156

LIST OF REFERENCES 173

LIST OF FIGURES

Figure 1: Parallel DNA Strand Arrangement 58

Figure 2: Linear DNA Strand Arrangement 59

Figure 3: All Coefficient Power Output of Example System 60

Figure 4: Significant Coefficient Power Output of Example System 60

Figure 5: Power Spectrum Linear Case (Left) and Gaussian Data (Right) 68

Figure 6: Power Spectrum Parallel Case (Left) and Gaussian Data (Right) 69

Figure 7: Linear Case Chi-Square Test Against Random System 69

Figure 8: Parallel Case Chi-Square Test Against Random System 70

Figure 9: Linear System 20 Sequence Power Overlay Graph 73

Figure 10: Parallel System 20 Sequence Power Overlay Graph 73

Figure 11: Cross Power Spectral Density Graphs (Linear Configuration) 76

Figure 12: Cross Power Spectral Density Graphs (Parallel Configuration) 77

Figure 13: 10-mer Linear Fitted Fourier Models 79

Figure 14: 12-mer Linear Fitted Fourier Models 79

Figure 15: 16-mer Linear Fitted Fourier Models 80

Figure 16: 10-mer Parallel Fitted Fourier Models 80

Figure 17: 12-mer Parallel Fitted Fourier Models 81

Figure 18: 16-mer Parallel Fitted Fourier Models 81

Figure 19: Periodogram 10-mer Linear System..... 82

Figure 20: Periodogram 12-mer Linear System..... 83

Figure 21: Periodogram 16-mer Linear System..... 83

Figure 22: Periodogram 10-mer Parallel System..... 84

Figure 23: Periodogram 12-mer Parallel System..... 84

Figure 24: Periodogram 16-mer Parallel System..... 85

Figure 25: Example Pressure Signal 87

Figure 26: Example Pressure Signal After LMS Filter..... 88

Figure 27: Pressure Output Spectrogram 10-mer Linear Water-Only..... 89

Figure 28: Pressure Output Spectrogram 10-mer Linear Water + DNA System 90

Figure 29: Linear System Configuration Scaleogram, Run 1..... 93

Figure 30: Linear System Configuration Scaleogram, Run 2..... 93

Figure 31: Parallel System Configuration Scaleogram, Run 1 94

Figure 32: Parallel System Configuration Scaleogram, Run 2..... 94

Figure 33: Options used for the Make-NA Server..... 122

Figure 34: AutoPSF Dialog Boxes for Segments 1 and 2 123

Figure 35: Solvate Options Dialog Box..... 126

Figure 36: Solvated Parallel System..... 126

Figure 37: Autoionize Dialog Box..... 127

Figure 38: Solvated and Ionized System 128

Figure 39: PressureParser tool screen shot showing batch interface 143

LIST OF TABLES

Table 1: Linear System Matching Statistically Significant Frequencies 63

Table 2: Parallel System Matching Statistically Significant Frequencies 63

Table 3: Power in Matching Coefficients Comparison..... 64

Table 4: Results of Runs Test for Randomness 65

Table 5: Comparison of Water/DNA and Water-Only Power..... 91

Table 6: Linear Significant Frequency Points 105

Table 7: Parallel Significant Frequency Points..... 106

Table 8: LMS Filtered Linear Significant Frequency Points..... 108

Table 9: Parallel LMS Filtered Significant Frequency Points 109

CHAPTER 1: INTRODUCTION

Chapter 1 Abstract

The development of DNA-focused scientific inquiry has been moving forward ever since Watson and Crick described the molecule more than a half-century ago. Yet despite decades of concerted inquiry into this unique molecule, and while much progress has been made in understanding its secrets, science still cannot explain some basic questions of DNA: why does DNA 'breathe', why is DNA robust, yet fragile under certain circumstances, and why does error correction work sometimes, but not every time? These questions, and many like them, are the subject of scrutiny, and in order to understand some of them, inquiry into DNA's structure, and behavior at the molecular level will be necessary.

This chapter presents the history of DNA research and makes a case for the significance of same. It lays out an outline-view of the current scientific understanding of DNA, the current focus of understanding genomes, and presents a brief gap analysis of some of the gaps in the fundamental molecular-level understanding of DNA. The proposed topic area of vibrational mechanics as one of those gaps is explored and the chapter concludes with a brief outline of possible explanations and the need for a systems-view of DNA.

Chapter Overview and Motivation for DNA Research

DNA (an abbreviation of deoxyribonucleic acid) has been regarded by biologists as one of the fundamental building blocks of carbon-based life, and yet, it remains shrouded in mystery. From the mystery of homologous pairing, to its resonant properties and vibrational signatures, there remain a large number of unexplained phenomena surrounding DNA. We have only recently been able to sequence an entire genome, and yet the significance of so-called 'junk DNA', the 'breathing dynamics' of DNA (the transient opening and re-closing of the strands of

the double helix) (Englander, Kallenbach, Heeger, Krumhansl, & Litwin, 1980), and DNA's apparent nonlinearity in the transmission of energy (Peyrard, 2004) all remain phenomena whose ultimate purpose in the function of DNA remains poorly understood.

While DNA is extraordinarily robust and capable of extraordinarily precise operations such as error-correction (through DNA polymerases, or enzymes), self-repair, recombination, and replication; it is also seemingly delicate, capable of being broken by not only ionizing radiation, but possibly non-ionizing radiation like radio waves (Alexandrov, Gelev, Bishop, Usheva, & Rasmussen, 2010; Korenstein-Ilan et al., 2008). It is suspected that the nonlinear nature of DNA's transmission of energy contributes to this seemingly contradictory nature.

In this chapter, we will discuss the history of DNA research, its molecular structure, and some of the underlying chemical physics that are relevant to this research. It is hoped that through understanding these physical phenomena that occur within DNA, we may be able to advance the state of the art in genetic diagnostics, industrial health and safety, and other branches of genetic science. Finally this chapter makes a case for the existence of vibrations occurring in DNA, the relative uniqueness of those vibrations, and concludes with possible explanations for those changes in vibration and what we may infer from those changes.

History of DNA Research

We would be remiss to discuss the history of DNA research without first discussing work of German scientist Fr. Gregor Mendel. Mendel worked in the 19th century and showed that inheritance of what we now call genetic traits followed a pattern: the pattern of dominant and recessive alleles (an allele is a particular expression of a genetic trait). Like 20th century German chemist Fritz Klatte (the accidental inventor of polyvinyl acetate), Mendel's work was not thought of as significant until later. He published a paper in 1865 titled "Experiments on Plant

Hybridization" with the findings of the two laws of inheritance, the Law of Segregation and the Law of Independent Assortment. The paper was largely ignored until 1901, when it was rediscovered, re-published, and within 30 years became the cornerstone for the study of genetic inheritance.

The first experiments into the chemical nature of DNA were done by a Swiss scientist, Friedrich Miescher. In 1869, Miescher discovered that inside of every cell's nucleus was a weak acid, which he called "nuclein". He published this discovery in 1871, and while it was not ignored, its significance was not well understood until a colleague of Miescher, German biochemist Albrecht Kossel, researched on the topic of nucleic acids from 1885 until 1901, and he discovered and gave the names to the five primary nucleotides: adenine, cytosine, guanine, thymine and uracil (substituted for thymine in single-stranded RNA). Kossel's work earned him a Nobel Prize in 1910 for these discoveries. A student of his, a Russian-American named Phoebus Levene, extended his work and discovered the 2-deoxyribose molecule in 1929. From this he was able to extrapolate that the phosphate-sugar groups that had previously been identified but whose use was not known, were used to build the 'spine' of DNA and allow the nucleotides to link together into a long chain. He posited that DNA's structure was tetranucleotide, meaning that DNA was based on four components, but that those components were all in equal amounts (and therefore could not encode any information). Research at the time was looking towards proteins as the method of genetic inheritance, and Levene's hypothesis was therefore largely accepted. When the search shifted away from proteins after Levene's death in 1940, work began on identifying the proposed "aperiodic crystal" that supposedly stored the material of genetic inheritance from Erwin Schrödinger's 1944 book *What Is Life?* Schrödinger was ahead of his time—realizing that some form of information encoding chemical

structure containing the material of genetic inheritance would control *how* proteins were expressed rather than thinking that proteins were the material of genetic inheritance.

In parallel with this work in chemistry, physicists, biologists, and genetic scientists were examining a new inter-disciplinary research area: molecular biology. The pioneers of this field sought to understand how, at a molecular level, processes within the cell and the cell's functions worked. Two developments, X-ray crystallography and the discovery that radiation can cause mutations, catalyzed what would become the field of molecular biology. Hermann J. Muller, in a way, began the search for the underlying theories that led to the study of molecular biology as a field. As a geneticist interested in the recently discovered X-rays, he bombarded fruit flies with X-ray radiation and studied what happened to them: they mutated, sometimes lethally so. His 1926 paper "The Problem of Genetic Mutation" was eagerly received by the scientific community, and within two years his results had not only been replicated, but also generalized to other living things: wasps and maize. As a result, he became one of the first radiation safety proponents. Muller's work with X-rays foreshadowed a much more important development in molecular biology, one which had been invented a decade before his discovery, but not applied to biology until almost two decades later: X-ray crystallography. This technique, first developed in 1914 by William Henry Bragg, used the diffraction of X-rays to map out the atomic structure of a crystal. Long before the electron microscope, Bragg had developed a technique that could be used to determine how a particular crystal was structured. In his work at the California Institute of Technology from 1925 to 1926, Linus Pauling applied this technique to chart out how molecules were put together. This work included the structure of proteins. Pauling determined that proteins were largely α-helical, that is, made up of a spiral structure with a right-hand twist. Two relative unknowns in the field, Rosalind Franklin and Maurice Wilkins developed a

technique to apply x-ray crystallographic techniques to samples that were not in crystal form, such as DNA, which does not crystallize easily. Franklin and Raymond Gosling took what is likely the most famous X-ray diffraction image in history: Photo 51, which depicted a helix, made by DNA from a calf's thymus in water solute showing DNA's B-form. It was Photo 51 that inspired James Watson and Francis Crick to build their model of the double-helix structure of DNA in 1953. For this, Watson, Crick, and Wilkins received a Nobel Prize in Physiology or Medicine in 1962. Sadly, Franklin had passed away from ovarian cancer four years prior, and the Nobel Prize cannot be awarded posthumously. Following Crick's success in modeling the double-helix structure of DNA, he went on to publish his "central dogma of molecular biology" (Crick, 1958), which proposed that, in short, DNA would be used to make RNA, which was used to make proteins. This work, republished in 1970 (Crick, 1970), laid the foundation for the understanding of DNA replication, DNA transcription (converting DNA to RNA), and RNA translation (using RNA to produce proteins). His experiment conducted with Brenner, et al. (Crick, Barnett, Brenner, & Watts-Tobin, 1961) in 1961 demonstrated that in order to code for one amino acid, three base pairs were required. These groups of three base pairs were then referred to as *codons* and to this day, codons are at the center of understanding how DNA encodes for proteins, and it is, in effect, the 'instruction set' for biological organisms that use DNA, much in the same way that there is a uniform instruction set for a computer processor.

The instruction set of DNA is fairly simple to understand, while the implications were astounding. In essence, with DNA arranged in blocks of three, and with each block allowing 4 different nucleotides at the start, a total of 64 'instructions' were available. While some codons encode for starting a second, and others for stopping, most encode for the production of various amino acids: the building blocks of proteins, the building blocks of life.

With the chemistry of DNA largely resolved, recent developments into DNA research have primarily focused upon three things: finding faster/better methods of sequencing (for the aid of geneticists, medical users, and forensic experts), DNA replication/recombination (to better understand how errors occur in DNA coding, mutations, cancers, etc.), and genetic engineering (the changing of genes without using traditional Mendelian hybridization techniques). While there have been other aspects researched, these three represent the bulk of what has been done from the 1960s through today. The Human Genome Project completed the sequencing of nearly the entire human genome by 2003. Having that sort of information available may enable researchers to find all manner of genetic mutations, differences and predispositions to certain medical conditions. However, being able to use that information without resorting to costly and/or slow genetic sequencing for testing will be a key driver of future uses of genetic data (Energy, 2013).

The DNA Molecule

The DNA molecule is the primary data storage mechanism by which cellular organisms are able to produce proteins. Despite its extremely high density (1 gram of DNA could store as much as 455 exabytes, or 477 million terabytes (Church, Gao, & Kosuri, 2012) of data), it is relatively simple in its composition: DNA is a polymer, specifically a polynucleotide, and its components are nucleotides, which are molecules composed of a nitrogenous base, a five-carbon sugar, and at least one phosphate group. There are only four base nucleotides used in the production of DNA: adenine, guanine, cytosine, and thymine. These nucleotides are usually referred to in DNA sequences by their initial letters: A, G, C, and T, respectively. Each nucleotide (or 'base') will only combine with its complimentary nucleotide in making DNA: adenine only with thymine, and cytosine only with guanine. These can be linked in four ways:

A→T, T→A, G→C and C→G, encoding two 'bits' of information for every base pair. The aforementioned phosphate groups in each nucleotide serve to interlink each 'step' of the DNA strand and produce the familiar double helix. These interlinks, which form pairs between bases as well as the links in the helix between each base pair, are hydrogen bonds. These are of particular interest due to their resonant and vibrational properties and will be discussed at length later in this document.

The Physics of DNA

Because of the research focus on DNA's biological impacts, genes, gene sequencing, and correlating those sequenced genes with diseases, comparatively little research has been done on the chemical physics of DNA. While we have sequenced the entire human genome, the cause and purpose of physical phenomena such as breathing (the temporary unzipping and spontaneous re-zipping of parts of the strand), how and why those breathing dynamics change in the presence of radio waves (Alexandrov et al., 2010; Bock et al., 2010), and the cause and purpose of harmonic vibrations (Chechetkin & Turygin, 1995), are not well understood. However, understanding these effects is important; for example, it has been posited that terahertz radiation may be able to damage DNA despite being a form of non-ionizing radiation, previously not thought to be harmful (Alexandrov et al., 2011; Korenstein-Ilan et al., 2008). Recent work on the chemical physics of DNA has focused on sequence-dependent changes in DNA, such as deformability/plasticity (Olson, Gorin, Lu, Hock, & Zhurkin, 1998) and flexibility (Kaukinen, Venalainen, Lonnberg, & Perakyla, 2003).

DNA and Vibrations

Of particular interest in this research are vibrations of DNA. This is an area of chemical physics that has received scant attention, but that has the potential to make large impacts in the

field. Understanding the features of intra- and intermolecular vibrations in DNA may be key to understanding how radio frequency energy can affect gene expression—a phenomena observed by the authors at least three papers (Alexandrov et al., 2011; Bock et al., 2010; Korenstein-Ilan et al., 2008)—as well as enable label-free methods of diagnosing genetic disorders (Miyamoto et al., 2005; Nagel et al., 2002; Woolard et al., 1997). Understanding the properties of the vibrations could provide insights into ways to conduct genetic testing without cycling and sequencing, as well as having health and safety implications in understanding whether and how exposure to non-ionizing radiation might cause DNA mutations, cancer and disease. Mathematical equations have been developed to describe the movement of atoms within a molecule (Plazanet, Fukushima, & Johnson, 2002; Smith, 1996), mostly derived from Hooke's law and classical Newtonian mechanics, but no larger systems-view of the movement of a DNA helix as a whole has emerged. One group of researchers reported a temperature-dependence in the anharmonic vibrational spectra of base nucleotides, but this dependence was over a very wide temperature range (room temperature and 4 K), and while it does demonstrate a temperature-dependent change in the vibrational spectra, how this discovery may impact future discoveries is unclear (Shen, Upadhya, Linfield, & Davies, 2003). Therefore, several key questions in the area of DNA molecular vibrational motion remain unanswered: is the motion periodic? Are the periodicities dependent on DNA strand content? Are the periodicities dependent on strand length? And, finally, is the motion unique?

A strong argument in favor of base-pair dependent vibration comes from Olson, et al. (1998), whose research into the sequence-dependent deformability of DNA found that the changes in conformation of DNA vary depending on the sequences occurring within a given DNA section. These changes are "…reminiscent of the normal modes of vibration of small

molecules." Furthermore, the authors found that "Some steps (CA, TA, AG), however, incorporate significant translational changes in the deformations whereas others (CG, AT, AC) involve essentially no base pair displacement"— showing that structural changes in the way the base pairs "stack" (or build steps) occur depending on sequence. These changes in location, conformation, and displacement across the six degrees of freedom (twist, tilt, roll, shift, slide, and rise) vary depending on the step. Because the displacement, location and conformation vary depending on the step, it is highly suggestive that the vibrational characteristics of each DNA strand will be unique, since, according to the authors, the set of steps that they gathered "complete the 'fingerprint' of each DNA dimer [base pair]" (Olson et al., 1998).

DNA Dynamics

In order for DNA to be replicated, repaired, transcribed, etc., it must be moved out of its double helix shape into various configurations, formally referred to as conformations. Because of this, the chemical structure of nucleic acid permits it to be flexible, with strong phosphodiester bonds between each 'rung' of the DNA ladder, and weak hydrogen bonds between bases. This flexibility means that both DNA and single-stranded RNA will behave like a complicated network of springs, which would likely be largely anharmonic. However, because DNA has a regular chemical structure, certain harmonic modes should also be expected. In 2003, Kaukinen, et al., conducted a molecular dynamics simulation investigation in nucleic acid chains where they found that the flexibility and energy levels between molecules varied and were "strongly dependent" upon the base sequence. Crucially, this research showed that not only is there evidence that the dynamics of a nucleic acid chain are dependent upon base sequence, but also that, as a result, those changes in the dynamics were transmitted over relatively long distances, such that changes in the inter-strand energies were not only affected by "…the neighboring

nucleic acid bases, but also those further apart in the molecule, …" (Kaukinen et al., 2003). Due to the nonlinearity of DNA's energy transmission, it may be expected that not only may the energies vary uniquely based on each base pair, but possibly also over the entire strand.

This nonlinearity of energy transmission was found to be sequence-dependent for single molecules by a group of German researchers. In short, the energies required to cleave the hydrogen bonds between nucleotides in each dimer varies depending on the base pair sequence (Rief, Clausen-Schaumann, & Gaub, 1999). This finding adds to the body of evidence that there are base-pair sequence-dependent changes that occur in DNA. While it does not directly address vibrations and only the energy required to cleave the bonds, the application of Young's modulus holds that the energy required is directly related to that material's elasticity and therefore, according to Hooke's law, the material's ability to transmit energy.

Potential Impacts and Motivation

Understanding the relationship between DNA, DNA states, DNA sequences, and vibrational harmonics may enable a host of useful technologies and techniques. It may be possible, if DNA vibrations are relatively unique, to diagnose genetic diseases more quickly without needing to label gene sequences or needing to sequence a genome outright to derive the desired information. DNA sequencing may be made more efficient by understanding whether the harmonics that exist are unique to certain codons or strings of codons. Such uniqueness could form a sort of 'fingerprinting' mechanism that enables the reading of sequences without requiring atomic-level resolution, leading to faster and more accurate genetic test results. Furthermore, understanding why such vibrations occur or how they can be elicited may grant insight into the molecular interactions between DNA and its functional enzymes such as helicases, polymerases, topoisomerases, ligases, etc.

The Case for a Systems-View of DNA Vibrations

In summary, while the research has revealed various aspects of DNA vibration—e.g. they assist in homologous pairing, they vary based on temperature, they are nonlinear, etc.—there is no overarching or unifying theory as to the nature of these vibrations. There is a strong case to be made that further research is required to identify whether there are larger themes at work in this area. The answers to questions such as to whether these vibrations are periodic, unique, and/or sequence dependent could very well be key to advancing the state of the art in DNA and medical research. Chapter 2 of this dissertation makes the case for developing further research towards what will hopefully become this systems view. A brief gap analysis will be presented with some of the numerous gaps that remain in order for us to better understand the DNA molecule. Particular attention will be paid to the gaps surrounding DNA vibration, dynamics, (an)harmonics, and phonon modes.

CHAPTER 2: FROM MOLECULAR MODELS TOWARD COMPUTATIONAL BIOLOGY

Chapter 2 Abstract

It is important to not underestimate the amount of thought that has gone into the creation and use of molecular models. This chapter presents a treatment of the history of molecular modeling from ancient history through the modern era and into the computer era to provide context for the exploration of DNA's vibrational mechanics as both a chemical and a modeling exercise. A quantum mechanical explanation of chemical modeling is introduced as a precursor to the faster, more optimized empirical force field models that power the majority of today's molecular dynamics models. Over the course of this chapter, the changing understanding of biology from its own art, to a field with significant input from molecular chemistry to molecular chemistry being a field with significant input from physics and the resulting mathematical concepts that make such an understanding possible are introduced. Finally, a gap analysis between the current models and the future goal of a systems view of biology is presented. This gap analysis presents additional motivation for this research which will be presented in chapter 3.

Molecular Modeling: Where We Came From

History

As far back as the 6th century BC, philosophers in India had a theoretical basis for the existence of the atom (from the Greek: "indivisible"), the fundamental particle upon which the universe was assembled. In the 1600s, scientists were again interested in understanding the structure of the physical world, when one Johannes Kepler theorized that the symmetrical nature of snowflakes was related to some invisible framework we would later come to know as the crystal. In that same century, Robert Boyle, the scientist that extended the work of Richard

Towneley and Henry Power to posit the ideal gas law, published *The Sceptical Chymist* in 1661 containing his argument that all matter consisted of elementary particles called "corpuscles" rather than the classical elements of earth, fire, air and water. This same theory was, a few years later, extended by Sir Isaac Newton to include light, and largely accepted for more than a century, though it was not known at the time that Christian Huygens' wave theory of light was also, at the same time, correct. Robert Hooke attempted to explain the structure of crystals as a sphere-packing problem. However, none of these early models considered stereochemistry (3D chemistry), instead holding that molecules bonded in a flat plane.

Enabling Works (Or: Biology is Basically Chemistry)

Almost two centuries later, in the early 1800s, French mineralogist René Just Haüy proposed that crystals had a regular lattice structure of atoms, similar to the same regular lattice that could be seen on the macro level – simultaneously, crystallography and stereochemistry had been invented. At the time, Haüy did not realize that they had planted the seeds for the discovery of stereochemistry, but a Dutch chemist, Jacobus Henricus van't Hoff, Jr., did. Van't Hoff's work included the discovery of the concept of osmotic pressure, the rules of chemical kinetics, and stereochemistry in 1874. In 1894, William Barlow FRS published *Über die Geometrischen Eigenschaften homogener starrer Strukturen und ihre Anwendung auf Krystalle* (*On the geometrical properties of homogeneous rigid structures and their application to crystals*), which included, among other things, the structural models of NaCl (ordinary salt) and CsCl (cesium chloride) which would later be confirmed as accurate with x-ray crystallography. These discoveries: chemical kinetics, stereochemistry and x-ray crystallography were three major enabling discoveries in the field of chemistry for scientists to begin to build accurate molecular models.

There was a fourth key discovery, but one from the realm of physics, not chemistry: spectroscopy. Spectroscopy allowed scientists to determine which elements a particular substance was composed of, as each compound had a different spectral pattern, which appeared as lines along the color spectrum. Joseph von Fraunhofer, in the 1800s, built more accurate spectrometers and invented the diffraction grating to quantify the spectral pattern of any observable substance, or even light from stars, or the sun. These spectral lines, to this day, are known as Fraunhofer lines. By the mid-1800s, Gustav Kirchhoff and Robert Bunsen (a physicist and a chemist) had embarked on a study to determine whether spectral patterns were unique for each chemical element. In so doing, they invented analytical spectroscopy, and chemical trace analysis. (Brand, 1995).

But there were major limitations to this understanding, the biggest being that scientists were now theorizing about structures that could not be visualized with a microscope directly. Up until the 1900s, scientists used rudimentary two-dimensional 'ball and stick' models to represent chemical structures, as first devised by August Wilhelm von Hofmann in the 1860s, with some three-dimensional changes as suggest by van't Hoff and French chemist Joseph Achille Le Bel, but there was much missing: what did the bonds between atoms actually look like, and what energies were represented by the bond structure? These questions and other related ones would start to be answered by the 1920s, as mathematical models of molecules began to be developed. Originally just an approximation of Hooke's law (Plazanet et al., 2002; Smith, 1996), accomplished by treating the bonds as springs and the atoms as masses, these models were not very useful. In 1946, a more accurate model was suggested by T. L. Hill which included steric effects (this is similar to the crystal structure studies in that each atom takes up a given amount of space), as well as Newtonian mechanics which included stretching, bending and torsional

vibrations (Hill, 1946). Hill's model of the force field that defines atomic interactions between the atoms in molecules began the study of computational chemistry and remains as the molecular mechanics model that modern models trace their roots to.

Biology is Basically Chemistry

At the same time as these discoveries were being produced in the chemistry community, there was a growing realization in the biology community, nearly 300 years in the making, that biological processes were, essentially, chemistry in motion. The word *metabolism* comes from the Greek, a term later applied to the studies of an Italian physician named Sanctorius. Although he did not realize it at the time, his empirical studies, published in his 1614 book *Ars de static medecina*, into the weight of his food, himself, and his excreta, his theory of 'insensible perspiration' and his studies into the temperature and pulse rate of humans were the foundational elements to understanding metabolism. Fast forward to the early 1900s, and the divergent paths of medicine and chemistry came back together into a field so new, a term was coined to describe it: biochemistry.

It was a known fact as early as the late 1700s that the stomach secreted acids to aid in digestion. Similarly, the action of saliva breaking down starches into sugars was known. But how those mechanisms worked, why they worked, and the processes that generated those secretions were all unknown. (Williams & Williams, 1904). It took almost a century, however, for the scientific community to begin to realize that other processes, such as fermentation and putrefaction were also part of these unknown processes. Louis Pasteur theorized that yeast was alive, because fermentation could not be explained by simple chemical means. He called the force within the yeast cells, in vogue with the vitalist thinking of the time, "ferments" (Manchester, 1995). Wilhelm Kühne coined the term *enzyme* to apply to the yeast fermentation

process in 1878, while a few years later in 1897, Eduard Buchner used yeast extract (with no living yeast cells) to ferment sugar. He named the enzyme responsible *zymase*, and received the Nobel Prize in Chemistry in 1907 for that discovery. But many questions remained: how do these enzymes work? What is their structure? How can something nonliving accomplish these feats of chemistry? The answers to those questions of biochemistry are still, to this day, being answered, and, more and more, those answers are coming from computational chemistry.

Computational Chemistry: Where We Are

Development of Molecular Dynamics Models (Or: Chemistry is Basically Math)

Although Hill's model represented the first molecular mechanical model with force fields that defined, as precisely as was possible at the time, the relationship between atoms in the various molecules, the models lacked both fidelity and usability. The first algorithm that pointed the way towards computer-based study of molecular dynamics was first published in 1953, and it planted the seed that computers could be used to simulate molecular dynamics on a far finer scale than could hand-computation. The Metropolis Monte Carlo algorithm simulated the movement of molecules on an atomic scale (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). But, being a Monte Carlo simulation, it did not actually simulate the dynamics of the molecular systems, rather it used a probabilistic approach with the Boltzmann distribution to determine the energy states of the particles. While it still lacked fidelity, and while it treated molecules as simple spheres, it served as a powerful demonstration as to the potential use of computers in molecular dynamics simulation: after all, chemistry was now, essentially, physics, which is, essentially, mathematics.

A few years later, a short letter by two theoretical physicists in the Journal of Chemical Physics changed the imprecise nature of molecular simulation, and brought with it the fidelity

that was needed if chemistry was to be reduced to a matter of mathematical precision. Alder and Wainwright's work on a UNIVAC computer provided precise information on about a hundred atoms by using Newtonian mechanics. Though the molecular systems were small, this was the first time that anyone had accurately calculated the dynamics of a molecular system: computational chemistry was born. (Alder & Wainwright, 1957). A few years later, Hendrickson reported in the Journal of the American Chemical Society doing molecular dynamic computations to study conformations of molecules with a force field model that he derived largely from previous works (Hendrickson, 1961). From the 1960s to the late 1970s, the field exploded with mathematical models and simulations covering everything from proteins to plastic. Most importantly, however, force field simulations began to appear in the 1970s, each one bringing with it greater fidelity. Entering the age of the personal computer in the 1980s and the age of the graphical user interface (GUI) in the 1990s brought with it exponential increases in computing power as well as much more useful visualization options for using molecular dynamics simulation. With every new model and refinement of the existing models, the ability of a model to accurately represent observed phenomena has improved and the existing models have met with wide acceptance in the chemistry and physics communities. (Schlecht, 1997). However, no process-oriented simulations had yet appeared.

The State of the Art

Looking at the current state of molecular dynamics (MD) simulations, three major types exist: quantum mechanical (QM), molecular mechanics (MM, often called "force field" or "classical" models), and the hybrid models that use both QM and MM techniques to either increase fidelity or speed for particular applications. The QM techniques yield precise information and spin and electron state of particles on the sub-atomic level, an incredibly high

level of fidelity. However, this fidelity comes at a steep computational cost: QM simulations are essentially restricted to small problem domains, and/or massively parallel supercomputers. MM techniques, on the other hand, provide reasonably high fidelity while being computationally parsimonious, and therefore even modest parallel computing setups can process reasonably complex systems in a reasonable amount of time.

Quantum Mechanical Models

To best understand molecular mechanics/force field (MM) models, one must begin with quantum mechanical (QM) models. QM models are capable of modeling each sub-atomic particle (electronics, protons, and neutrons) very precisely *ab initio* (Latin: "from the beginning" or from first principles, without needing additional assumptions). The amount of precision is essentially the same precision available to an electron microscope user: spin, electron state and charge density are all obtainable from these calculations. This precision is possible because of two developments: Schrödinger's equation, and wave-particle duality theory. (Leach, 2001)

QM is possible because of work that began with Christian Huygens and Sir Isaac Newton and that was completed many years later by Planck, Einstein, Heisenberg, de Broglie, and others. Essentially, at its most fundamental level, quantum mechanics is the understanding that all elementary matter has wave-like properties, and as a consequence, fundamental particles (electrons, protons, neutrons, quarks, gluons, bosons, muons, taus, etc.) may be treated mathematically as waves because they are one and the same. What this implies is that the motion of those particles can be described by known wave functions (or, in some texts, state functions, because they describe the state of a particular wave-particle), which will describe the motion of the sub-atomic particles of an atomic system. Erwin Schrödinger posited an equation that describes how these quantum systems change over time in an electric field (though not a

magnetic field, the Pauli equation provides the solution for particles in a magnetic field), much in the same way that Newton's second law (that the net force acting on an object changes linearly the object's momentum) describes change over time in classical mechanics. The equation, which follows conservation of energy in its terms takes the following form when it is time-dependent (i.e. it shows changes over time rather than considering the standing wave case):

$$\left\{\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right\}\Psi(\mathbf{r},t) = i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) \text{ where } \nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) \qquad (1)$$

In this equation, *m* is the mass of the particle, **r** is a vector with the position in Cartesian coordinates, *t* is time, *i* is the imaginary number, $\hbar$ is Planck's constant divided by 2π (useful in cases like this when one is considering angular frequency), V(r,t) is the total potential energy imparted by an external field (electrostatic forces only, this equation does not consider magnetism), while Ψ is the wave function (the function that describes the motion of the particular particle being solved for). (Leach, 2001).

The Schrödinger equation can accurately describe the motion of all the subatomic particles in the system under consideration. Given this, why is it not used for simulation? There are several limitations: Schrödinger's equation works exactly only for very particular cases (none of which are applicable to the sorts of complex biological systems we are considering), and only for a very small number of particles (otherwise the problem space becomes too large to compute), and only for non-interacting particles. For larger systems, heuristics and approximations must be used, the most common of which is the Born-Oppenheimer approximation. The Born-Oppenheimer approximation takes advantage that the mass of the proton is 1836 times greater than the mass of the electron, which introduces a simplification: for multi-particle systems, we discard the electrons and only consider the protons and neutrons, because the electrons will move more-or-less instantly in response to changes of protons and

neutrons. Because of this, it is possible to treat the electron terms separately from the nuclei terms, or in equation form:

$$\Psi_{\text{total}}(\text{nuclei}, \text{electrons}) = \Psi(\text{electrons}) \times \Psi(\text{nuclei}) \quad \textbf{(2)}$$

Here, Ψ is the wave function, but the equation can just as easily be thought of as an energy equation thanks to classical conservation of energy, such that:

$$E_{\text{total}} = E(\text{electrons}) + E(\text{nuclei}) \quad \textbf{(3)}$$

This concept, combined with the fact that, in general, models can be applied to related molecules (as opposed to having to calculate new models for each molecule), means that these simplifications permit the force field family of MM models to function. (Becker, MacKerell, Roux, & Watanabe, 2001; Leach, 2001) .

The Born-Oppenheimer approximation makes solving large particle systems tractable, but it comes with several downsides, one of which makes it unsuitable for many biological systems simulations. The first downside is that an assumption of the Born-Oppenheimer approximation breaks down (or becomes invalid) in the case where the gap between the energy states is smaller than the movement of the atomic system (in metals, for example, the gap is zero, and therefore the approximation is invalid). The second is that the approximation breaks down in semiconductor and nanomaterial analysis (Pisana et al., 2007), which itself would not be a problem, save for the fact that evidence exists that DNA can exhibit properties of semiconductors and nanomaterials (Fink & Schonenberger, 1999; Meggers, Michel-Beyerle, & Giese, 1998). Despite these limitations, it is important to note that without the underlying QM models, MM models would likely not exist.

Molecular Mechanics Models

Molecular Mechanics (MM) models are simplified models compared to their Quantum Mechanical cousins, and dispense with quantum-level details such as spin, electron configuration, polarization, etc. in order to reduce the time required for a model to be processed. Essentially MM models take the atomic configuration of a system, ignore the elementary particles using heuristics for the various atoms, and produce the resulting energies of the system, just like a QM model does, minus electron-level detail. Although MM models vary widely depending on the terms that are considered and their biases, they all share several characteristics: they are molecular, considering only the nuclear particles *en masse* as a single atom and not considering electrons at all; they are empirical, meaning that there is not necessarily one 'correct' model; they are heavily optimized (simplified) and drop many of the terms and features found in QM models; and finally, they all consider two main kinds of forces: those arising from bonds between atoms (stretching, angle bending, and torsion), and those arising from non-bonded interactions (such as electrostatic and van der Waals forces). These simplifications result in a significant decrease in computation time for a given molecular system, but with some losses of fidelity: bonds are not considered to break or be made (because the terms dealing with bonds are treated harmonically, keeping the bond energy terms from exceeding an equilibrium value and therefore making or breaking); the temperatures are therefore restricted to an area around room temperature (although this suits biological processes just fine); and particle-level detail is unavailable. Despite this, MM models can provide accurate, fast results for large atomic systems that would be intractable or impossible to compute with QM models. (Becker et al., 2001; Leach, 2001). It should be noted however that MM models have one drawback: no transferability between models. The energy data that is produced by one model will generally

not be transferable to another model because of the different ways each model processes the energy outputs of a given system.

All so-called "force field" molecular mechanics models work in a similar way: a potential energy function is solved to determine the energies of each individual atom in the system by evaluating two terms, internal energies and external energies. Letting E represent the potential energy and *R* the three-dimensional structure of a molecule or entire system, Becker notes the equations as:

$$\mathrm{E}(R)_{total} = \mathrm{E}(R)_{internal} + \mathrm{E}(R)_{external} + \mathrm{E}(R)_{other} \text{ , where} \quad \textbf{(4)}$$

$$\mathrm{E}(R)_{internal} = \sum_{bonds} K_b(b - b_0)^2 + \sum_{angles} K_b(\theta - \theta_0)^2 + \sum_{dihedrals} K_\chi[1 + \cos(n\chi - \sigma)] \quad \textbf{(5)}$$

$$\mathrm{E}(R)_{external} = \sum_{\substack{nonbonded \\ atom\ pairs}} \left(\varepsilon_{ij} \left[\left(\frac{R_{\min,ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{\min,ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{r_{ij}} \right) \quad \textbf{(6)}$$

(Becker et al., 2001; Guvench & MacKerell, 2008; Leach, 2001)

Three major terms are considered: the internal terms, arising from bonded interactions between atoms, the external terms, which arise from interactions between atoms caused by electrostatic or van der Waals forces, and finally other forces, which are imposed on the model by the researcher's parameterization setup when the model is initialized. Equation 5, with the internal force terms accounting for bond stretching energy, angle bending energy and dihedral energy (torsional energy caused by the twisting of bonds), assumes that the system is in equilibrium. As such, it is not possible to model bond breakage with this equation, but it has been shown to be as accurate as a quantum model for all other cases. Examining the first term in equation 5 more closely, *b* is the interatomic bond length, b_0 is the expected or natural bond length (such that there is no energy applied from this term if the bond does not deviate from the natural, free-state bond length), K is a term that describes the relative stiffness of the bond (such

that the stiffer the bond, the higher the energy any stretching or shrinkage imparts)—this is, essentially, an application of Hooke's Law for springs. In the second term of equation 5, this application of Hooke's Law remains, but the equation concerns θ and θ_0, the bond angle, and the natural bond angle, respectively. A better approximation than the quadratic term for bond stretching is given by Morse (Morse, 1929), and the Morse equilibrium is used by many force field models. In the third and final term, the equation differs from the other two terms because the torsional forces are not linear, they instead vary sinusoidally through 2π of rotation, and are therefore best expressed by a sinusoidal function.

More interesting than the relatively straightforward internal energy components of the molecular mechanics force field are the external forces: electrostatics and van der Waals forces. The electrostatic term is the simpler term. Decomposing equation 6, the electrostatic term is:

$$\frac{q_i q_j}{r_{ij}} \qquad \textbf{(7)}$$

This is, quite simply, Coulomb's Law, with q_i and q_j being the sum of the charges of the two atoms (i and j) and r_{ij} being, again, the distance between the atoms. The other component is the van der Waals force, this force is actually a sum of all the non-interactive and non-electrostatic forces at work between two atoms. This equation considers the attractive forces between dipoles (permanent and induced), as well as considering the Pauli Exclusion Principle—which states that two identical fermions (a class of particle that includes electrons) cannot have the same quantum number. In short, as two atoms, or groups of atoms move towards each other, they become increasingly attracted to each other to a limit—the Pauli force acts at near distances to prevent the atoms coming into contact with each other. While the two forces eventually cancel each other out, the Pauli force is modeled as being the square of the attractive force, this is a rough approximation, but it is more than able to properly account for the underlying quantum effects

and it is therefore used in Amber, CHARMM, GROMOS, and other force field codes. (Becker et al., 2001; Guvench & MacKerell, 2008; Leach, 2001).

The third major class of models are the hybrid Quantum Mechanical/Molecular Mechanics (QM/MM) models, which we will discuss briefly as they represent the future of molecular dynamics modeling. These models attempt to meld the best characteristics of QM models (accuracy, electron-level precision, etc.) with the speed and simplifications that make very large problems tractable with MM models (Leach, 2001). These models work by selecting part of the system to simulate in QM (generally a small subsection of particular interest) and the rest of the system to simulate in MM. This approach yields fine detail about the desired area, such as a protein binding area, while modeling the entire system as well. There are problems, such as the inability to handle electrons that are covalent with the QM region from the MM region, but novel codes like ONIOM have been developed to alleviate some of the inaccuracies caused by these limitations (Vreven et al., 2006). These models have been regarded with renewed interest lately due to the desire to bring more accurate simulation models to areas such as automated drug screening. Lately, new approximations called semi-empirical QM/MM models have been developed to extend these models and allow for high-precision calculations that are much faster than the *ab initio* Hartree-Fock model which precisely considers every electron. Stewart (2009) reports a hybrid model using the MOZYME MD code to simulate a 14,000 atom system of proteins using a semi-empirical QM/MM model, which produced much higher quality data than an MM model while taking similar time on commodity hardware. For additional reading on emerging Quantum Mechanical approaches to molecular dynamics simulation, we refer the reader to the excellent review by Bryce and Hillier (2013).

Computational Biology: Where We Are Going

Towards a Systems View (Or: Biology is Basically Math)

One of the major shortcomings of molecular dynamics models as applied to biology is the lack of an overarching architecture describing the activity of biological systems on the molecular level *in silico*. This field, called computational biology is, however, the future. A direct framework to go from math to biological processes is needed to solve all kinds of biological problems, from cancer research to drug design, in less time than before. Already this fledgling interdisciplinary field (with its roots, variously, in computer science, mathematics, chemistry, physics, medical, and many others) has spawned conferences and journals with the aim of informing and developing this growing field. It has also helped launch the related field of bioinformatics, which the NIH (2000) defines as "Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store organize, archive, analyze or visualize such data."

Although computational biology has officially existed since the very late 90s, most of the field has been concerned with areas such as computational genomics, DNA processing, genome sequencing, and macro-level analyses as opposed to a systems-level view with computational biologic techniques as a way to get from mathematics, to chemistry, to biology in one framework (Pevzner, 2000). Ouzounis editorialized that this might be because the field has grown in so many directions since its founding that it has become less focused on a systems view, and more focused on individual challenges within the field (Ouzounis, 2012). Some have reported the existence of systems-view tools that have been developed to solve particular problems, such as predicting how proteins will behave given their sequence, though these views still lack roots back to empirical molecular mechanics or *ab initio* quantum mechanics views (Juncker et al.,

2009). However, as longer-run models are being enabled with faster computers and better approximations (Klepeis, Lindorff-Larsen, Dror, & Shaw, 2009), it is expected that the field of biology as a whole will begin moving towards a paradigm of viewing the functioning of biological systems as systems not only on the macro level, but also on the molecular level (Ouzounis, 2012; Pevzner, 2000).

Interactions Are Key

In understanding computational biology, the look to the future largely includes interactions of all kinds: between genes and proteins, between proteins and enzymes, and in all manner of chemical machinery that run organisms. Noble describes this process as 'linking' in his review of the state of biological simulation: the linking of different levels of biological systems (such as chemical to genomic, or genomic to physiologic) represents a stepping stone on the way to a framework view some have termed "theoretical biology" (Noble, 2002), or a framework akin to theoretical physics. The difficulty with expanding on this interactive view of biology is that "…frequently multifunctional, sets of elements interact selectively and nonlinearly…" and that "…functions in biological systems rely on a combination of the network and the specific elements involved" (Kitano, 2002). To wit, to be able to properly get to a systems view, one must have an abstraction of the underlying biological molecular mechanics, and then the underlying macro-scale biological processes, such as protein interactions, an area where there has been concerted research effort (Shoemaker & Panchenko, 2007). This is a problem domain with many fragmented solutions and still no clear systems view, but it is clear that understanding interactions will be key to synthesizing such a view. Interactions between genes, between various levels of biological abstraction, proteins, pathways, and cells need to be

understood, mapped out, and, if we are to have effective models, solved with mathematical equations and not just numerical solutions. (Kitano, 2002; Noble, 2002).

Please, Mind the Gap (Between the Present and the Future)

Noble, in speaking about the gap between the current understanding of computational biology and a future shift to an integrated, systems view of biology such as theoretical biology, notes that "we have only tentative ideas on what this set of principles might be, but they would include evolution and the theory of complex systems". It is clear, however, that unless an individual or group of researchers posit a unifying theory of computational and/or mathematical biology, much like Einstein did with theoretical physics, it will be necessary to continue to work on individual pieces of the larger problem of developing a systems view of biology and, at some point in the future, begin to integrate them (Noble, 2002). Somewhat unexpectedly, therefore, one of the major gaps in the field is the lack of a systems view of computational biology at all. However, within the areas of computational and mathematical biology, there are several individual gaps identified by the literature, one or two of which may be addressed by this research.

At the time of writing, there has been little progress on identifying factors that may permit the construction of pattern recognition heuristics, algorithms, rules of thumb, etc. between a given observation and a given DNA pattern. The current state of the art requires that DNA be PCR (Polymerase Chain Reaction) cycled, sequenced, and then that sequence compared with the known genome to determine whether a particular gene is present. It may be possible to identify factors occurring within DNA or that interact with DNA to permit the quick recognition of certain patterns, sequences, or errors without sequencing the individual genome. While researchers have posited that vibrational spectra (Miyamoto et al., 2005), harmonic vibration

(Plazanet et al., 2002; Shen et al., 2003; Tatiana et al., 2006), and direct reading of DNA binding states using very high-frequency radio waves (Nagel et al., 2002) may operate as recognition mechanisms, there is much fundamental research that remains to be conducted on which factors may be relevant in the recognition of such patterns, let alone which technologies are best for capturing that data. Porcar et al. (2011) identified a need for unifying technological mechanisms for understanding cell mechanics as one of the ten "grand challenges" of synthetic biology.

In the same vein as the identification of possible factors, a gap exists in understanding whether those factors or combinations of those factors are highly sensitive and specific. In diagnostics, sensitivity is related to type 1 statistical error (false positive rate), while specificity is related to type 2 statistical error (false negative rate). Determining whether these factors are either highly sensitive and specific, or even unique may enable a host of new applications, diagnostic tests and other enabling technologies that would make the process of analyzing DNA less arduous than it currently is. For example, the determination of a unique signature for a certain genetic mutation could mean that cancer patients can receive the right kind of chemotherapy more quickly because a factor-based test could replace a genetic test that sequences a particular gene to determine whether that patient's biology can tolerate a given chemotherapeutic agent.

A possible factor that has been identified by the literature (Chechetkin & Turygin, 1995) is the periodicity of the base pairs that occur within DNA. These periodicities vary in species and may also vary between different areas within a complex chromosome (Worning, Jensen, Nelson, Brunak, & Ussery, 2000). Several gaps remain, specifically, questions of: whether those periodicities are unique (or usable as fingerprints), whether those periodicities form larger shifts in the vibrational spectra that could be captured using Fourier analysis of a spectroscopic

examination or other analysis technique of the DNA, and whether those periodicities had high sensitivity and specificity for particular characteristics such as transposition error, gene mutations, particular coding for a given protein, etc.

Finally, there has been some work in the area of sequence-dependent DNA variations. One team of researchers has identified that the macroscopic motions occurring within DNA are sequence-dependent (Matsumoto & Olson, 2002). This finding, combined with previous work (R. J. Calloway, 2011) point to a gap in understanding how these macroscopic motions vary. More importantly than the question of whether these motions vary depending on base-pair sequence is the question of whether these motions are unique or at least sufficiently unique to form a sort of signature that may be used to extend the goal of analyzing DNA without needing to resort to sequencing. At this time, a paper is being prepared for submission in the Journal of Computational Chemistry, which contains work by our research group that included a preliminary study of sequence-based vibrational variation in double-stranded DNA (dsDNA) molecules (R. Calloway, Proctor, Boyer, & Napier, 2014). That research inspired and informed this effort to more closely study the vibrational characteristics of DNA.

Conclusions

The area of DNA mechanics simulation has been considered with particular interest since the dawn of the personal computer age, when it became possible to do complex calculations relatively quickly on modern commodity computer hardware. While the raw mechanics of these simulations come from a relatively settled area of physics, the implications of these simulations and the extension of these basic theories into the realms of mathematical and computational biology remain largely unresearched and undocumented. It is hoped that this simulation research

will, in a small way, help to extend these basic principles to contribute to the developing systems view of biology in this field.

CHAPTER 3: RESEARCH HYPOTHESES, METHODOLOGY, AND MODEL SELECTION

Chapter 3 Abstract

In chapter 3, the hypotheses of this research are presented and the chosen experimental design(s) are presented. The domain of available molecular mechanics model force fields is reviewed, and evaluated of appropriateness for application to the research hypotheses. A rationale for the selection of the selected force field model is presented.

Research Hypotheses

In this research, several hypotheses have been posited and developed all with the common goal of better understanding the relationship between DNA's physical characteristics and its motion. These hypotheses were selected based on three criteria: the identification of potentially unique characteristics of DNA that vary according to the physical structure, the ability to simulate the tests required on a COTS nanomolecular dynamics simulation engine, and possible applicability to future research, particularly in the areas of label-free genetic diagnostics and industrial health and safety.

The question of how this applies to the areas of Industrial Engineering practice such as health and safety, as well as ergonomics, is answered though the potential contribution of this research to the improved understanding of DNA's characteristics. While it has long been established that DNA is corrupted or damaged by ionizing radiation such as X-rays and gamma rays (Muller, 1927), it has been recently suspected that DNA may be damaged by non-ionizing radiation as well such as terahertz frequency radio waves (Alexandrov et al., 2010; Korenstein-Ilan et al., 2008). Cases of cataract development in radar operators and technicians were first believed to have been caused by the thermal effects of microwaves, though later research

demonstrated another, unknown, "nonthermal", mode of damage (Zaret & Snyder, 1977). Because the mechanism of damage in the case of non-ionizing radiation is not clearly understood, there remains a gap in understanding for which contributions to the understanding of that mechanism, and in assisting the community with developing standards for safe exposure would be very valuable. However, that cannot be done without first understanding the link between DNA's vibrational frequencies and its composition. It is through understanding that relationship that future research into the vulnerability of DNA to damage via non-ionizing radiation may be possible, and it will be necessary to have a clear understanding before we can begin to model the impact of non-ionizing radiation in an environment.

Hypothesis 1: DNA's vibrational motion is dependent on strand length

When DNA is solvated in water at normal 'room temperature', it experiences a natural vibration due to the Brownian motion of the particles in the water colliding with the DNA. These vibrations produce pressure waves in the water much in the same way as a piano string, once plucked, produces vibrations in air. (R. J. Calloway, 2011). While spectroscopic data exist as to the normal vibrational spectra of individual nucleobases (Shen et al., 2003; Ten, Burova, & Baranov, 2009), the vibrational pressures that occur in the separating water between DNA molecules has not been investigated, likely due to the technical challenges presented when attempting to do so *in vitro*. Basic physics would suggest that the vibrations of these DNA strands would vary depending on the length of the strand, much in the same way that varying length strings in a piano produce different fundamental vibrations (and different harmonic vibrations as well). The null hypothesis is that there is no evidence to suggest a relationship between DNA's motion and strand length, while the alternative hypothesis is that there is evidence to suggest a relationship between DNA's motion and strand length. The nature of this

relationship, whether linear, exponential, logarithmic or non-continuous is not hypothesized but will be discovered by testing if there is evidence to suggest a relationship between DNA's motion and strand length.

Hypothesis 2: DNA's vibrational motion is dependent on strand content

In a similar vein to the first hypothesis, this hypothesis posits that DNA's vibrational motion may be dependent on strand content. Specifically, the aim is to discover whether the particular sequences (A-T, G-C, etc.) cause the vibrational energy to vary in a manner dependent on the content of the strands. There is evidence in the literature to suggest that a potential relationship exists—Rief, et al. (1999) used atomic force microscopy to stress DNA molecules and discovered that the forces required to convert B-DNA (a relatively uncoiled conformation [c.f. A-DNA, which is tightly coiled] with a right-hand spiral that is found in hydrated environments and inside most cells) to S-DNA (a stretched conformation) varied depending on the bond: A-T bonds required statistically significantly less energy than G-C bonds—while others Matsumoto and Olson (2002); Olson et al. (1998) have shown that DNA's deformability also varies depending on its structure. Therefore, it is reasonable to investigate whether the motion is dependent on the content of the strand—much in the same way that the stiffness of a string can change its fundamental and harmonic vibrational frequencies when plucked. The null hypothesis is that there is no evidence to suggest a relationship between DNA vibrational motion and strand content. The alternative hypothesis is that there is evidence to suggest a relationship between DNA vibrational motion and strand content. Just like the relationship between DNA motion and strand length, the nature of this relationship is not hypothesized but rather will be discovered during experimentation if there exists evidence to suggest a relationship between strand content and DNA vibrational motion.

Hypothesis 3: DNA's vibrational motion uniquely varies with strand length and/or content combinations

The third hypothesis aims to discover whether the motion of DNA, when it varies by strand length and/or content, is sufficiently unique to allow it to be identified in the absence of knowledge of the type of DNA that is present (for example: if there is sufficient uniqueness it would be possible to discriminate between an A-T bond and a G-C pair simply by analyzing the vibrational output of that solvated DNA). There is, in the literature, some research that has linked these vibrations to the spectroscopic properties of a given DNA base pair (Miyamoto et al., 2005), and that they are "practically superpositions of the spectra" (Ten et al., 2009). These results from spectroscopy hold promise that there may be, in the naturally occurring DNA vibrations, some form of unique vibrational characteristics that could be ferreted out by future detection techniques (such as the developing terahertz-radiation detectors) and obviate the need for sequencing of DNA molecules in genetic testing. Therefore, the null hypothesis is that there is insufficient evidence to find uniqueness in DNA's vibrational motion when DNA content or length is varied. The alternative hypothesis is that there is sufficient evidence to find that DNA's vibrational motion differs in a statistically significant manner when DNA content or length varies. If there is evidence to suggest that there is a uniqueness in DNA's vibrational motion under varying DNA strand content or length, then additional experimentation will be needed to determine whether these variances are unique across small changes as well as large changes, or large changes only.

Hypothesis 4: DNA's vibrational motion forms a mathematical relationship

Finally, it is necessary to investigate whether these vibrations, if dependent on strand length and/or content, and if sufficiently unique, add/subtract/divide/multiply into a composite

'signature'. In other words, might it be possible to identify a strand of unique DNA by base pair configuration solely from its vibrational signature, or if not, might it be possible to identify changes existing inside of a known DNA strand (such as a corruption, transposition or mutational error)? While the sequence-dependent changes occurring to DNA's mechanics have been established (Matsumoto & Olson, 2002; Olson et al., 1998; Rief et al., 1999), whether those sequence-dependent changes extend to the non-spectroscopic vibrational characteristics of DNA has yet to be investigated. The null hypothesis is that there is no evidence to suggest that vibrational changes change the signature in a mathematically related fashion (whether linearly, multiplicatively, exponentially, logarithmically, etc.). The alternative hypothesis is that there is evidence to suggest that vibrational changes change the signature in a mathematically related fashion. These hypotheses will be tested by taking DNA segments of known signature, linking them, and seeing whether the vibrational outputs resulting from that linked DNA can be identified as the constituent parts of the two strands before they were joined.

Simulation Tool Selection

In order to best approach these research questions, it will be necessary to determine the appropriate simulation package to use. Although there are dozens of options available, only a few contain the features required for this research, and of those, fewer support the desired force field models. In short, the software should be either free or of low cost, well-supported, either by its authors and/or a community, be capable of handling DNA, and permit offloading of tasks to multiple CPU cores across a gigabit Ethernet network (in order to fully take advantage of the computational resources available in the Synthetic Environment Learning Laboratory or SELL Lab). Quantum mechanical (QM) simulation is neither needed nor desired for this research (due to high CPU requirements and the resulting long simulation runtimes), so QM capabilities will

only be discussed briefly if the force field supports it. This narrows the list to only a few force fields: CHARMM, GROMOS, and Amber, and therefore, a few simulation tool packages: NAMD, GROMACS, and Amber. What follows is a brief review of these three force field models to determine their suitability for the research at hand. This is not an exhaustive review, for that, the reader is referred to Guvench & MacKerell (2008) for an excellent review and comparison of these force field models.

Before speaking about the differences between models, it is important to remember that, at their core, all Molecular Modeling (MM) force fields have many more things in common than they have differences. All MM force fields use the same basic idea for calculating total energy, as was presented in equation 4, the total energy is the sum of the energy from: internal, external and other sources. Where the total energy equals the internal energy (from bonded interactions: bonds, angles, and dihedrals), plus the external energy (from nonbonded interactions such as Lennard-Jones potentials, Pauli-exclusion principal potentials, van der Waals forces, electrostatic effects), plus model-specific energy quantities. The main differences between models occur in the last two terms: nonbonded and other, model-specific energy quantities. All force field models carry some important limitations simply by the fact that they are not quantum mechanical (QM) models, these include: limited support for reaction computations, limited, specialized support for conformational changes, limited support for phase change energy computations, limited transferability between models, and, owing to their structure, minimal to no support for metal ions. In essence, force field models are ideal for steady-state modeling of well-characterized and known molecules, but lack the ability to fully model a system undergoing chemical or state changes without first being "made aware" that such changes in chemical state are possible. This is similar to the challenges faced in finite element analysis models whereby

the models are sufficient for computing stresses up to the material's failure limit, but unsuitable to calculate results such as crumpling or tearing without first being optimized for such outcomes. In contrast, a QM model can simulate these conditions without difficulty—though at a cost of being hundreds to thousands of times slower than an empirical model, and that cost makes QM models unsuitable for this research.

CHARMM

Chemistry at HARvard Molecular Mechanics or CHARMM, is a well-known force field code used for molecular dynamics simulations. First released in 1983, this force field code has proceeded over the years to incorporate an extremely rich feature set, including optimizations for proteins, nucleic acids, and lipids. CHARMM is not a direct quantum mechanical (QM) model, and therefore operates using parameterization—a technique that 'tunes' its force field equation (the potential energy function computed for each atom) for given chemical structures. There are benefits and constraints to this approach: the primary benefit of parameterization is the optimization of processing time given that each molecular parameterization has been prepared using spectroscopic and crystallographic data. This allows the model to neglect terms that are not required to reproduce the molecule's behavior—while not every molecule is in the CHARMM force field's extensive database of parameters, CHARMM is still able to provide a stable approximation of molecules that are not parameterized; however, this is rarely an issue today because of CHARMM's wide ranging support for all common and many uncommon biological molecules. Initially, the CHARMM force field was intended as a special-purpose tool highly optimized for a handful of particular biological molecules, but it has developed in time to permit simulation of almost any biological molecule. Original releases of CHARMM were unable to cope with nucleic acids in a solvent such as water, but the CHARMM27 force field is capable of

doing so (MacKerell, Banavali, & Foloppe, 2000). CHARMM27 'understands' chirality of molecular structures and conformation of sugars, nucleic acids, and proteins, allowing it to accurately process complex biological molecules in varying conformations under varying conditions—such as shifting DNA molecules between B- and A-DNA forms depending on the presence of water (Foloppe & MacKerell, 2000)—and doing so efficiently. CHARMM's available optimizations include the ability for the user to specify the cutoff distance at which nonbonded interactions are no longer considered (to save CPU time and prevent the system from becoming a polynomial-time problem), numerous precompiler commands such as FASTer (to save time in computing energies of portions that are not critical to the research at hand), EXPAND (which expands the loops and removes as many IF statements as possible, easing the burden on the CPU's branch predictors), and lookup tables (which determines solvent-solvent interactions without resorting to calculating them). However, due to its fully parametric nature, studying bond breakage and formation, such as in reaction pathways, requires special attention. Because CHARMM assumes that bonds are harmonic unless explicitly told otherwise, in cases where bonds are expected to be made or broken, special preparation must be undertaken, namely that the Protein Structure File (which lists every bond, angle, angle type and other data required to produce the energy of the system) be updated so that the bond to be studied for breakage/formation be appropriately anharmonic with its energy parameters already provided. This is a significant limitation in that bond formation cannot be studied *ab initio*, but must be prepared earlier using known data sources, and that therefore, only known molecules can be studied for bond breakage/formation. It must be noted however, that while CHARMM's provisions for studying reactions is limited due to the nature of MM models, CHARMM does provide facilities to build hybrid MM/QM models to study these reactions *ab initio* with relative

ease. Finally, CHARMM supports parallel tempering, a technique that applies Monte Carlo analysis to systems by simulating multiple runs of an identical system at the same time point at different temperatures, and then exchanging model components with each run to its neighbor (Swendsen & Wang, 1986). This technique, known in MD simulation as REMD (Replica-Exchange Molecular Dynamics) results in a much higher probability that the energy levels that will be found will be true global minima or maxima, something not necessarily true with standard MD simulations (Earl & Deem, 2005). CHARMM supports doing REMD fully with all available features of the engine. (Brooks et al., 2009).

GROMOS

The GROningen MOlecular Simulation program, GROMOS, was first released in 1980 as a tool for investigating the structure and nature of polymers. Since then, it has significantly grown to allow for the investigation of not only a wide range of polymers, but of glass, crystals and biological molecules. Like many MD packages, GROMOS has a large library of native molecules, a wide range of supported hardware platforms, heavily optimized codes for simulation and a large number of output options for post processing simulation results. GROMOS is also aware of conformational changes in proteins, and how to solvate biological molecules. However, GROMOS has some unique features that bear mentioning in this review. One of the most significant for this research is GROMOS's ability to stop and resume a simulation run in a fully deterministic manner—GROMOS is a fully checkpointed simulation code which permits the simulation run to be stopped, all state data 'lyophilized' and the system 'reconstituted' and restarted from that exact point later on (it should be noted that CHARMM is partially checkpointed, but not fully as GROMOS is). This feature carries with it a significant advantage: GROMOS can be used, in combination with the GROMACS simulation engine

(Pronk et al., 2013), in classical multiple-run experimental designs. So instead of doing one long 2μs run, one might run 20x100ns runs and then be able to perform tests of statistical inference on the output data (Lange, van der Spoel, & de Groot, 2010). Another feature of the newest GROMOS release that can be useful for biomolecular simulations is the ability to do model runs at levels higher than the atomic level. GROMOS supports coarse-grained model components, for example treating a molecule as a single discrete large particle rather than its components. Besides the obvious negative side effect of losing atomic-level detail, there is a significant advantage: in complex systems, especially those in solvent, the computations can be orders of magnitude faster than with full atomic-level resolution—the authors report 10^3 to 10^5 times faster runs with complex molecules that can be simplified into coarse-grained molecules with accuracy losses that only number in the low percentages. Finally, there is the matter of model thermostats. In a model environment, the temperature and pressure (which is simply the net energy in the system) can change over time in a way not consistent with actual biological molecules. To cure this, MM force fields provide thermostats (and barostats) to ensure the system stays within experimental temperature range as a system would in the lab or *in vivo*. Because of the function of a thermostat, it is important for them to follow a Boltzmann probability distribution (sometimes known as a Gibbs distribution), which is a probability function that describes the mechanics of a system that is in thermal equilibrium. These are known as strongly coupled thermostats—other thermostats are weakly coupled, such as Berendsen, and they do not properly ensure that this distribution holds. This can lead to experimental error if used in long-running simulations; but they can be useful during the warm-up stage because it will tend to dampen wide swings in temperature and pressure when starting up a simulation that is very far away from its equilibrium point (Guvench & MacKerell, 2008). GROMOS supports several thermostat

models for constant-temperature and constant-temperature and pressure simulations, whereas CHARMM only supports the Langevin piston temperature/pressure model (conceptually similar to a Hoover thermostat), GROMOS supports several different models, including the strongly coupled Woodcock, Nosé-Hoover Langevin dynamics thermostats, and the weakly coupled Berendsen thermostat. GROMOS also fully supports REMD, but has the added advantage of being resume-able at any point in the simulation due to its check-pointing ability. In short, GROMOS provides a comprehensive, well-documented interface with a wide target audience and is useful for a wide range of molecular dynamics simulation needs. (Christen et al., 2005).

Amber

The Amber force field model was first released in 1979, and its current release, version 12, presents a strong MD package with many capabilities in biomolecular simulation. It is impossible to write about the Amber force field model without some discussion of its tools. The Amber model is both a force field and a toolset (in contrast to CHARMM and GROMOS which are purely models that can run under different MM tools such as NAMD or GROMACS). Like other molecular dynamics packages, Amber provides a wide range of parameters supporting most organic solvents, amino acids, carbohydrates, and lipids (Cornell et al., 1995). The software is commercial, but inexpensive for non-commercial use, $400 USD at the time of writing. Amber also provides, as free and open-source, a library known as AmberTools that creates the setup files compatible with the various Amber force fields (which are implemented in other tools such as NAMD and that are available for free). Unlike the other MD simulation packages, Amber is actually three separate engines: Sander, pmemd, and pmemd.cuda. Sander is the oldest engine and provides all features of Amber, while pmemd and pmemd.cuda provide a subset to focus on providing production-grade high-performance computing in multi-CPU

(pmemd) or multi-GPU (pmemd.cuda) situations. There are some important limitations to pmemd and pmemd.cuda, the biggest for users investigating nucleic acids is that there is no support for the Langevin thermostat model within Amber's implementation of REMD (Salomon-Ferrer, Case, & Walker, 2013), nonetheless, this is a significant improvement over version 9, which only supported weakly coupled thermostats. For users of Amber that aim to simulate biological processes, this could present issues due to the inability of a thermostat algorithm that is not strongly coupled to reproduce a Boltzmann distribution in the simulation run (Guvench & MacKerell, 2008); while one could use sander, its focus on being the development branch of Amber could mean having a slower-performing simulation (especially if using GPUs). Unique to Amber is constant pH control, a valuable tool for anyone investigating the properties of acids or bases when the system under study could change its pH during the experimental runs. As of version 12, pmemd supports controlling the pH of a system in solvent which can resolve issues with change in pKa (the acid dissociation constant) when there is a change in system conformation (Mongan, Case, & McCammon, 2004). On the performance front, Amber's novel method of dividing the Fast Fourier Transform (FFT) into blocks permits distributed computers with a large number of CPU cores to calculate portions of the FFT independently without resorting to a global FFT map, with attendant performance improvements. Amber is therefore a comprehensive, widely-used and well-supported force field and simulation package for studying biological molecules. (Duan et al., 2003).

Selection Rationale

The chosen simulation tool needed to have the following qualities: sufficient, validated fidelity of the simulation to provide vibrational and pressure profile data that could be analyzed; sufficient capabilities to do what is being asked of it; speed, so that runs can be done in a

reasonable period of time (weeks versus months); and parallelizability, so that the tool could be run over a clustered network of computers dedicated to the task to increase the speed and capacity of the simulations. Further, it was desirable if the tool selected would have optimizations and/or be aware of nucleic acid structures (if not a QM model), and, ideally, it would be either free or low cost. Based on these criteria and the review undertaken, the NAMD simulation package (released by the Theoretical and Computational Biophysics Group at the University of Illinois at Urbana-Champaign) with the CHARMM force field would be the most appropriate set of tools for this investigation (Phillips et al., 2005), and will therefore be the toolset used in this research. For visualization and manipulation of model data, the VMD (Visual Molecular Dynamics) package will be employed. Testing of the NAMD simulation package began in late 2013 and after optimizing both the computer equipment in use and the simulation parameters, the equipment setup in the SELL Lab (see appendix C for specifications) produced runs of a small parallel test system provided by Dr. Calloway (with two 5' TATAAACGCCTATAAACGCC 3' sequence DNA molecules in a 15 angstrom (Å) solvated water box) at a speed of nearly 1,900 2fs (femtosecond) frames per 'wall clock' second. Each 'wall clock' second amounted to 48 CPU-seconds, and about 525 billion floating point operations per second (GFLOPS).

Research Methodology

Because, essentially, all of these hypotheses are inter-related by one experimental variable—vibrational motion of DNA—it is proposed that the investigation take the form as four parallel-developed molecular dynamic simulation models. However, it will be necessary to determine *a priori* which measurements will be significant for the input and output variables. To begin with, a standardized timeslicing quantum of 2fs has been selected. This selection is based

on the following rationale: biological molecules tend to have normal mode resonant vibrations in the low THz range (Norizawa, Herrmann, Tabata, & Kawai, 2005), and a 2fs timeslice equates to a maximum frequency-sampling rate of 250 THz according to Nyquist's theorem. The dependent variable is vibrational motion of DNA, which will be converted to frequency data via Fourier transform. The content of the independent variables will be chosen per experiment, but will largely be sequence and strand length. In order to permit this research to proceed in a reasonable time span, system size will be limited to 10 base pairs for hypothesis 1 (approximately 30,000 atoms) and 20 base pairs for hypothesis two (approximately 50,000 atoms), and the confidence interval desired will be 99.9%. Larger systems sizes would significantly increase simulation time requirements by approximately the square of the system size. Therefore, the run lengths selected below were done to ensure that all simulations could be completed within a 6-month timespan, while allowing time for the inevitable debugging of the simulation engine and simulation parameters. The variables of temperature and pressure will be controlled with the simulation software's Langevin piston thermostat/barostat capability, while the variable of solvent (water) density will be controlled by the software's solvent routines. No other variables besides sequence, strand length, strand position, and vibrational motion will be considered by this research.

The variables of sequence and length will be controlled in this case. Unlike simulations using biologic molecules based on naturally occurring DNA, the molecules to be studied will be constructed using the Nucleic Acid Builder (NAB). Using NAB will permit complete control over DNA content, allowing for fewer aliasing effects when looking at the vibrational motion of DNA. It will also permit the creation of random DNA strands that have the same number of

each nucleotide in different sequences to test for possible aliasing effects as well as widen the range of experimental data to be gathered.

To analyze the frequency of DNA vibrational motion, it is anticipated that Fast Fourier Transform (FFT) analysis will allow for the determination of the sample's frequency spectrum. From there, a signal to noise ratio will be computed and compared with the background frequency noise of the system. Statistically significant points (which would show as peaks on an FFT graph) would indicate that there is evidence to reject the null hypothesis, while a lack of statistically significant points would therefore result in a failure to reject the null hypothesis. Further analysis may be undertaken for a particular hypothesis, which will be discussed in that specific hypothesis test section.

Experimental Design to Test Hypothesis 1

Hypothesis 1 proposes that DNA's vibrational motion will change in a way related to strand length. To test for this effect, two identical DNA strands will be placed next to each other in a solvated water box (System 1) and two identical DNA strands will be placed end-to-end linearly in a solvated water box (System 2). These design choices are based on the results of R. J. Calloway (2011), where he reported that these two configurations have the highest signal to noise ratio of the four configurations tested. The separation distance between molecules will be 25 Å from edge to edge, this will ensure that there will not be confounding effects caused by electrostatic or van der Waals forces, while also ensuring that the systems do not become mirrored due to periodic boundary conditions. The size of the water box will extend approximately 12Å from the last atom of the molecular system; this is because van der Waals and electrostatic forces are disregarded beyond a 10Å cutoff in molecular simulations due to their effect becoming negligible, and allowing 2Å to avoid molecules that move near the edge of

the box from having their charges wrapped around due to periodic boundary conditions (Guvench & MacKerell, 2008). Both systems will be solvated, brought up to room temperature and allowed a short run to equilibrate into its most stable low energy state, a process known as minimization (the use of Langevin piston thermostatic dynamics should allow the system to equilibrate without excessive temperature excursions). At that point, the simulation will be paused, and data collection for pressure profile (which NAMD supplies as a pressure tensor 3x3 matrix) will be enabled, and the simulation restarted from that checkpoint to gather the pressure data for the analysis.

A run length of 20 per configuration is being proposed. This should, with the statistical methods to be used, provide adequate protection against type II error while being sensitive to relatively small changes in the pressure distribution. Each of the two systems will be run 20 times for 1 nanosecond to gather initial data. After initial data gathering runs, the strands will have 15% more nucleotides added and 40 more runs (20 per configuration) will be run as described above. Strands will again be made 30% longer than original and 40 additional runs will be collected. This will, with two, 10 base-pair fragments of DNA, take approximately 3 months of computing time. While these length changes are somewhat arbitrary, Quake, Babcock, & Chu (1997) noted in their *in vitro* work with DNA molecules that the relationship between molecule length and their "normal mode" (or vibrational spectra) is roughly linear, and that therefore we can expect to see a roughly linear change between the base system and the additional runs; further, the odd percentage steps are to prevent potential negative effects from periodic boundary conditions (which are used to allow the system to 'wrap around' in the water box as previously discussed). The resulting pressure profile data will be analyzed comparing each system configuration to itself (and therefore each configuration is its own control) using a

Fourier Transform to search the frequency space for statistically significant changes in frequency content coincident with strand length. Further discussion of the statistical methods is found near the end of this chapter.

Experimental Design to Test Hypothesis 2

Hypothesis 2 proposes that DNA's vibrational motion will change in a way related to strand content. To test for this, a design similar to Hypothesis 1 is proposed. Twenty (20) samples of randomly sequenced 20 base-pair dsDNA will be constructed with NAB that contains an identical number A/T bonds and C/G bonds (which will keep the strand charge-neutral). The sizes and distances to the water box edges will remain the same as Hypothesis 1's design for expediency. Each system will be warmed up and equilibrated as in test 1, and then allowed to run in both the parallel (System 1) and linear (System 2) configuration 20 times, each run being a different unique DNA strand. After pressure profile data is gathered from these 40 runs, the pressure profile data will be subjected to a Fourier transform, analyzed, this time searching for statistically significant differences in the vibrational spectra of each system's separating water. This randomization should provide good contrast without the aliasing effects of electrostatic charge differences.

Experimental Design to Test Hypothesis 3

Hypothesis 3 proposes that DNA's vibrational motion will partially or uniquely identify the underlying strand length or content. Hypothesis 3 is therefore a meta-analysis of the data generated by the tests for Hypotheses 1 and 2. It is proposed that the data collected from Hypothesis 1 and 2 will be further analyzed to determine whether the vibrational signature varies in such a manner as to serve as an identifiable parameter. Fourier Transform analysis will be used to compare each system configuration and strand length/content to determine whether

changes cause statistically significant changes in the output pressure profile that are robust for the purposes of identification of such changes.

Experimental Design to Test Hypothesis 4

Hypothesis 4 proposes that DNA's vibrational motion forms a mathematical relationship, whether additive, subtractive, multiplicative or divisive. Hypothesis 4 is therefore a companion of Hypothesis 3's meta-analysis. Additional data analysis will be undertaken on the Fourier Transform frequency data to determine, via regression and other appropriate statistical methods to determine whether any statistically significant form of relationship exists between DNA's vibrational motion and system changes.

Statistical Methodology for Testing Hypotheses 1 and 2

The statistical methodology for Hypotheses 1 and 2 are presented together due to both Hypotheses testing for the same output: the resulting pressure profiles of the systems. While FFT coefficients are not, on the face of it, continuous, it must be remembered that FFT coefficients are merely the amplitude of the wave at a particular frequency, and the frequency spectrum is continuous (Blackford, Salomon, & Waller, 2009). Furthermore FFT coefficients are only considered independent and identically distributed (IID) if their underlying distribution is also IID, and then only independent as far as the underlying data's independence as FFTs transform time-series data, which may have natural dependent characteristics (Shumway & Stoffer, 2011). Note that, inferring from the physical principles that underlie this process, the distribution of the pressures of the water media is likely to be independent (due to the Brownian motion of the water causing vibrations that are Gaussian-Markov in nature), it is unknown whether they are identically distributed. While Shumway and Stoffer (2011) provide several statistical techniques for computing Fourier coefficient statistics, it is possible that a non-

parametric statistical test for comparing samples might be desired. A test such as the Multivariate Permutation Test or MPT (Blackford et al., 2009) is nonparametric and has a reasonably low set of requirements for replications at the desired 99.9% confidence interval. However, its requirement of exchangeability between observations may rule it out in this case due to insufficient knowledge about the underlying signal data. Thankfully, extraordinary measures are not required in this case. Due to the large number of frequencies to be sampled, the Central Limit Theorem (CLT) assures us that while the underlying distribution is unknown and possibly not normal, the sampling distribution would, if sufficiently large, be normal. Peligrad and Wu (2010) published a proof showing that the CLT applies to Fourier transform functions. Therefore, in light of this proof, the Chi-Square test will be employed to compare frequency distributions.

Statistical Methodology for Testing Hypotheses 3 and 4

The expected relationship between strand length and frequency is linear due to Quake et al. (1997), but it is not known whether that means the frequencies observed will increase monotonically across the spectra, or whether the distribution of frequencies will shift linearly, or whether certain parts of the spectra will be positively linear or negatively linear. To begin with, the computation of cross-spectral density will provide a measure of the 'covariance' (a simplification) of the Fourier spectra being compared. If the signals show cross-spectral density, it can be expected that there is a relationship between them. Therefore, the process of Fourier fitting will then be employed, and an F-statistic will be computed (Thibos, 2003) to test hypothesis 3 for the variable-length case. Similarly, if a model can be fitted to the variable-length DNA molecules, there is a mathematical relationship between length and pressure profile, and it would satisfy hypothesis 4 at the same time.

The variable-content case is more complex, however. In this case, it is not a matter of the frequency distribution being expected to shift, but rather, the distribution will change in as-yet-unpredictable ways. It is likely that certain frequencies will be dominant with certain sequences and regress to the noise floor with others. Therefore, it would be useful to determine whether there are rhythmic changes occurring. It is proposed that the output data will be charted, the most prominent frequencies will be selected, and Chi-Square testing will be used to determine whether significant changes occurred in the selected frequencies. To determine whether there is a mathematical relationship, the procedure of model fitting described above will be employed.

Data Collection Methodology

As stated prior, each of the four hypotheses will be tested with data collected from various runs of the NAMD simulation tool on the target molecular systems. Data collection in molecular dynamics models is continuous, analogous to logging in a flight recorder, and the models will be run with full profile data enabled to ensure capture of all relevant data: molecular positions at each time step, vectors, and pressures. The general technique to be used for each test is as follows: a system will be created, then permitted to 'warm up' to room temperature and stabilize before the simulation data collection begins. Per Bhandarkar et al. (2012), two NAMD runs will be conducted for each system: the first (molecular dynamics) run will output all position and energy data as well as limited pressure profile information, the second (pressures only) run will output the full pressure profile for the system. Those data will be merged using a custom program and the results of the pressure analyzed in MATLAB via Discrete Fast Fourier Transform. The results of the Fourier transform will be analyzed using the statistical methods described above. Finally, data summaries will be compiled into spreadsheet form using custom written Visual Basic for Applications (VBA) routines for comparison and visual analysis. It is

anticipated that these custom applications will be released as a toolset to automate the analysis of pressure profile and DNA vibrational spectral data. With these applications, future research (such as research involving DNA molecules and electromagnetic radiation, once the simulation engines allow for this) can benefit from having already-built custom tools to perform the analysis.

Experimental Predictions

In research as fundamental as this, it is difficult to even make an educated guess let alone a studied conjecture as to the probable outcomes of this work. However, some general inferences can be drawn from the literature to provide a guess as to the outcomes. For Hypothesis 1, it is difficult to draw any conjectures as it is a new investigation. Classical physics would suggest that, as alluded to earlier in the piano string analogy, there is a significant change with strand length, and that, according to Quake (1997), the change will be linear in nature. For Hypothesis 2, results from classical and NMR spectroscopy (Santamaria, Charro, Zacarías, & Castro, 1999) do suggest there is a relationship between vibrational pressure profile and base-pair configuration, but this has yet to be demonstrated concretely, let alone simulated. For the analyses in Hypotheses 3 and 4, it is unknown whether there will be a unique or mathematical relationship, and while arguments could be made from first principles that there is likely to be a relationship, it truly is unknown. Regardless of any of the outcomes, this research is going to continue to examine some previously unknown areas of molecular dynamics models as they relate to the simulation of DNA molecules and the statistical challenges of analyzing the resulting data and is therefore significant.

CHAPTER 4: RESULTS

Chapter 4 Abstract

The testing of the four hypotheses by means of two experiments and four tests was accomplished using constructed DNA models with molecular dynamics simulations as proposed. Experiment 1 tested $\mathbf{H}_{(\text{Length NULL})}$, a hypothesis that proposed the idea of strand length being related to vibrational motion. That experiment, with multiple repeated runs of the same simulation model in six different cases, failed to find significant effects with system dimension changes. Furthermore, post-hoc tests on the matching significant variables between runs demonstrated an apparently random response pattern under repeated runs of the same system. Experiment 2 tested $\mathbf{H}_{(\text{Sequence NULL})}$, the hypothesis that strand sequence is related to significant frequency change. This experiment yielded inconclusive results: although a very small number of coefficients were in fact significant, they were too few to test further. Experiments 3 and 4 were meta-analyses, using the data collected from experiments 1 and 2 to run further tests. Experiment 3 applied cross-spectral power density to attempt to find differences between system configurations that stood out as unique (the hypothesis known as $\mathbf{H}_{(\text{Unique NULL})}$). The results were not sufficient to reject $\mathbf{H}_{(\text{Unique NULL})}$. The final experiment tested $\mathbf{H}_{(\text{Relation NULL})}$, the hypothesis that there is a mathematical relationship between DNA vibrations and its spectrum. This was accomplished using doubled strands repeating the same sequence, which had been shown to potentially increase the amplitude of certain frequencies. No difference was detected between the two cases, in part due to the high system noise, and thus failed to reject $\mathbf{H}_{(\text{Relation NULL})}$. Despite these results, this research answers previous research questions, identifies needed improvements to computational tools, points the way towards an alternative analysis approach, and identifies questions that may be answered by future researchers investigating this topic area.

The effort for these experiments expended an excess of 4,560 compute hours (or slightly more than a quarter-million CPU hours), and the effort for Experiment 2 consumed 1,680 compute hours (100,800 CPU hours)—both of these represent a much larger expenditure of CPU time than has been possible in the past. These figures do not include failed runs, of which there were several.

Finally, a brief discussion of additional post-hoc analysis using wavelet analysis is presented. Based on the scalograms obtained, it is likely that the underlying processes being studied in this research are either non-stationary or, at the very least, have a longer period than what could be studied in this research. Evidence for those possibilities is presented, along with a wavelet analysis of the possibility that the output is solely noise.

Experiment 1: Test of Hypothesis One: DNA's Vibrational Motion is Dependent on Strand Length

For this experiment, a molecular system with a randomly chosen nucleotide sequence was generated by the tool provided by Maduro (2003) for the 10-mer case, and additional random nucleotides were chosen to lengthen that strand to 12- and 16-mer in length. The process to setup the molecular dynamics runs used for this experiment are the same in the following three experiments. Once the randomly chosen nucleotide sequences were obtained, the DNA molecular sequence protein structure files were built using the excellent Nucleic Acid Builder tool (Stroud, 2006). These molecules were imported into a new system in pairs. The resulting molecules were placed apart in their respective configurations (linear and parallel) with 25Å spacing between the ends (in linear) or sides (in parallel) of the molecules. The systems were then solvated with water to a distance of 12Å past the position of the last atom of the system in all six directions. These distances were deliberately chosen for two reasons: one, there is an

approximately 10Å cutoff for the electrostatic and van der Waals forces, and two, this avoids the system being an even length through the reflection of the periodic boundary conditions (Guvench & MacKerell, 2008). Ionization to a level of 0.5 moles/liter (mol/L) with sodium chloride ions was undertaken to ensure that the system would be electrically neutral (a necessary condition for molecular dynamics simulation) and similar to the salt levels found in the human body.

After the creation of an ionized system, three preparatory steps are required so that the system will be ready for the full simulation runs. These steps are minimization, heating and equilibration. Minimization is the process of permitting the molecules to find their lowest energy state in the bonds, essentially relaxing the molecules into their potential wells. Heating is done after minimization, and the system is slowly heated to 310K (36.8C or 98.3F, approximately body temperature) with NAMD's integrated ramping function. This step is required because, when a system is first created, it is at 0K, and there is no energy. Heating the system puts energy into the system, which, as it does in the real world, causes the molecules to vibrate. Finally, equilibration is required. Equilibration is a much longer process that is done to allow the newly heated system to get back to a normal lowest-energy state. Guidance for finding the equilibration point of DNA molecular simulation was found in the NAMD tutorial (Isgro et al., 2012). Given that there was good convergence (as indicated by the Root Mean Square Distribution) by 3,500 steps, 10,000 steps were chosen for the equilibration run. This standard procedure was documented and used for the startup of every system in this research.

After equilibration, the experimental runs began. After equilibration, the experimental runs began. For Experiment 1, the three systems were each run 20 times, one in online mode, and once in offline mode. These double runs were necessary due to a quirk of the NAMD molecular dynamics simulation software package: it is not possible for NAMD to output all terms for pressure at the same time. As one will recall from equation 4, there are three kinds of interactions that are considered: internal forces such as bonds, external forces such as electrostatics, and other forces considered by

the model. NAMD can output internal forces and certain non-bonded interactions, but the electrostatic terms must be computed separately from the trajectory data using Particle-Mesh-Ewald (PME) sums. This is what necessitates having two runs per system: the first time, the dynamics are run online and a trajectory file is output along with total pressure exclusive of electrostatic interactions; the second time, the trajectory file is used to inform the PME calculation and supply the electrostatic interaction term. The two can then be summed, and a custom tool was written to automate that process.

Statistical Analysis for Experiment #1

In order to investigate the hypothesis proffered for experiment 1, that strand length causes significant changes in the noted vibrational motion (abbreviated $\mathbf{H}_{(\text{Length NULL})}$), each of the 60 pressure profiles will be subjected to Fast Fourier Transform (FFT) analysis. The FFT is a specialized version of the discrete Fourier transform (DFT) that is optimized for rapid computation on high-speed data processing equipment. The FFT works by decomposing the input signal into sinusoids (sines and cosines), with a given intensity and frequency offset. The FFT will output an array of frequency coefficients, equally spaced, from the lowest frequency detectable given the time span it analyzes to the highest (which will be one half the highest frequency seen in the data due to Nyquist's theorem). It is essentially taking a time-domain signal and converting it into the frequency domain. The FFT outputs a frequency spectrum, not dissimilar to the electromagnetic frequency spectrum, except because the FFTs in this research are derived from pressure, it is a vibrational frequency spectrum.

At the time of writing, Fourier analysis of molecular dynamics has not been widely studied, though a recent paper in JACS demonstrates Fourier analysis for proteins in a molecular dynamics simulation (Lindorff-Larsen, Trbovic, Maragakis, Piana, & Shaw, 2012). There are likely several reasons for this, chief among them being that Fourier analysis has long been the

domain of pure mathematics and largely applied through the disciplines of electrical and computer engineering, finding limited biological applications in magnetic resonance imaging, infrared spectroscopy, computed tomography, and recently in terahertz imaging. Additionally, it has not been thought to be useful to apply Fourier analysis to characterizing the motion of biological systems—certainly not if the referee comments from prior submissions of (R. Calloway et al., 2014) are anything to go by—though attitudes towards its potential use appear to be changing. Finally, studies of DNA vibrational spectra have been largely limited to spectra of its component parts and individual nucleobases, in both the traditional infrared (as in Raman spectroscopy) and THz ranges (Shen et al., 2003). Using FFTs to analyze molecular model pressure profile data is new, but not without precedent (R. Calloway et al., 2014; Tamaoki, Yamauchi, & Nakai, 2005). Finally, Lindorff-Larsen et al. (2012) report using FFT against MD data. Their validation against NMR shows good general agreement between MM models and proteins, and we can generalize to DNA from proteins.

In crafting the analysis of $\mathbf{H}_{(\text{Length NULL})}$, it was noted that conventional descriptive statistics (which speak to mean and standard deviation) would not return usable information about the spectra of each run. Because of the natural resonant properties of water (which resonates around 10^{12} Hz) and of the DNA molecules, it was likely that the mean and standard deviation of all runs would be similar. There is one test that stood out in its ability to provide meaningful insight into the statistical significant of Fourier confidents: a version of H. O. Hartley's F-Max test (Thibos, 2003). Several "off the shelf" statistical tests were considered prior to Hartley's test, including the relatively new Multivariate Permutation Test (Blackford et al., 2009), Hotelling's T^2 test (a generalization of Student's t test capable of treating Fourier coefficients as vectors), and a relatively new version of the T^2 test built for situations such as

those encountered in this research where this is a small first degree of freedom and a very large second degree of freedom (Wu, Genton, & Stefanski, 2006). The MPT was rejected for two reasons. First, it requires *a priori* knowledge of the exchangeability between observations in order to be valid—we could assume exchangeability, but could not prove it in this case. Second, while the MPT may otherwise have been effective, it appears to be incapable of examining changes between multiple runs since it is a paired methodology, used to detect changes pre- and post-treatment. Hotelling's T^2 test is unsuitable due to the large number of variables and small number of replications relative to the number of variables (at best it would return a negative test statistic—a meaningless result, at worst the covariance matrix would fail to be positive definite, so the test statistic could not be computed at all), and the test described in Wu is difficult to adapt to this particular test situation, one of repeated runs. Therefore, Hartley's test along with additional processing steps were incorporated to determine whether there was significant change in the power spectrum between each length setting.

In this experiment, the molecular systems were arranged in two distinct patterns. These two patterns were chosen because they would likely give the greatest frequency response, as reported in Calloway (2011). In terms of the underlying natural processes, the forces of DNA strands when placed in the side-by-side and end-to-end cases present two standard cases for the biological processes of strand break repair, and replication, respectively. As shown in the figures that follow, Figure 1 is the parallel side-by-side configuration, and Figure 2 is the linear end-to-end configuration. Each figure has been enhanced to show the DNA helix structure and the ions more clearly, with the water molecules being represented by the red and white sticks.

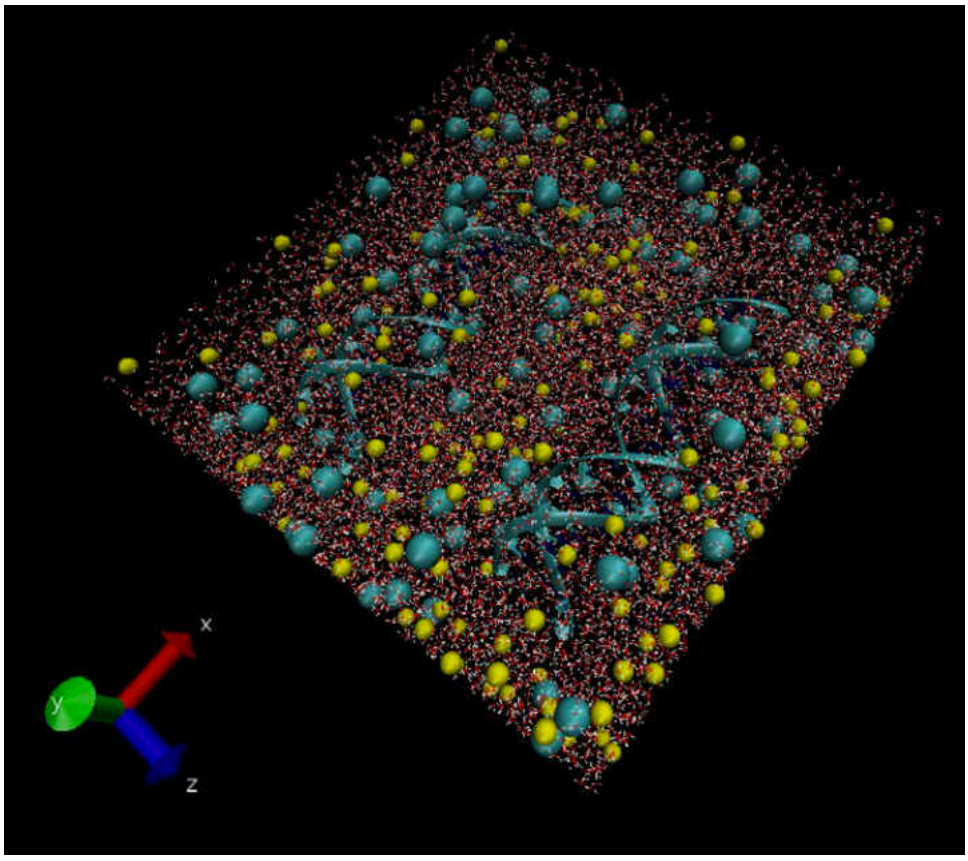


Figure 1: Parallel DNA Strand Arrangement

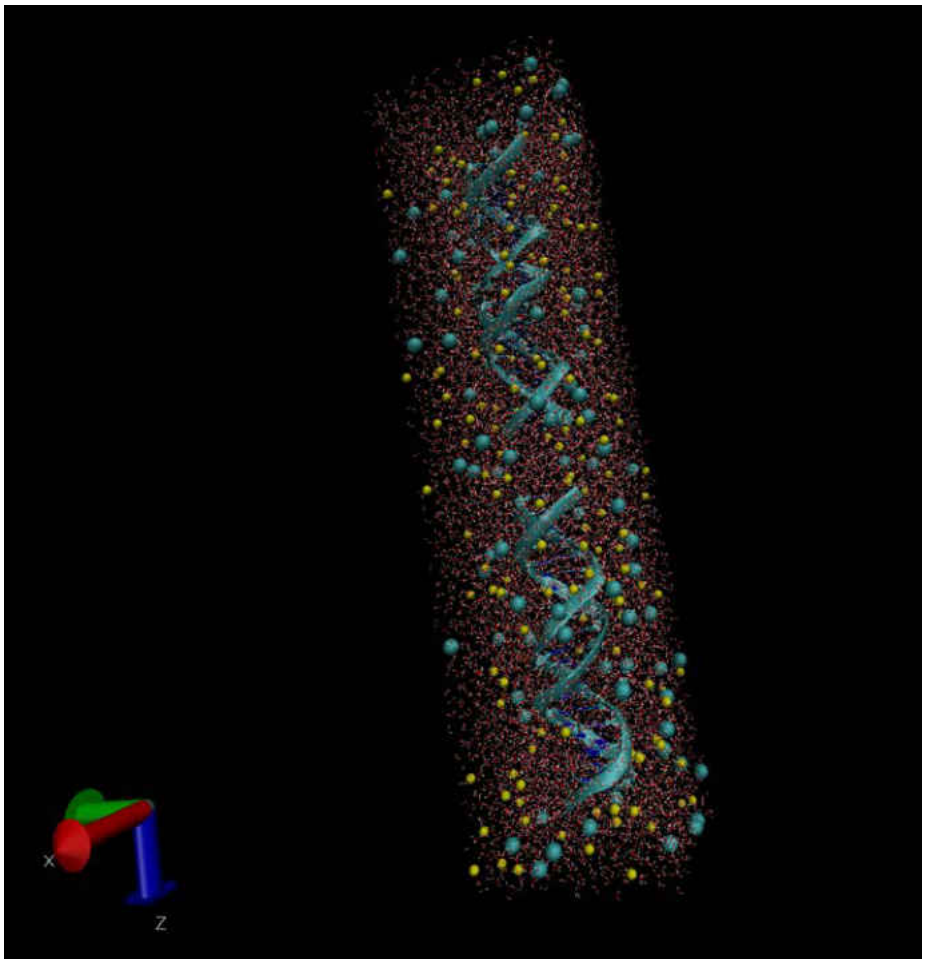


Figure 2: Linear DNA Strand Arrangement

The analysis began with the creation of a custom Matlab program to facilitate the ingest of pressure profile data as output by the PressureParser tool (see appendices for a description of each program and source code), running of the Fast Fourier Transform (FFT), computing the power of each frequency band, inspecting the resulting power (using Hartley's F-Max test for heteroscedasticity) for statistical significance, and then outputting the statistically significant power signatures from each run to a spreadsheet for further analysis via matching and a Chi-Square test. The Matlab program also output graphical data showing both the shape of the resulting power spectrum and that same power spectrum with only significant coefficients, as illustrated respectively in figures three and four below.

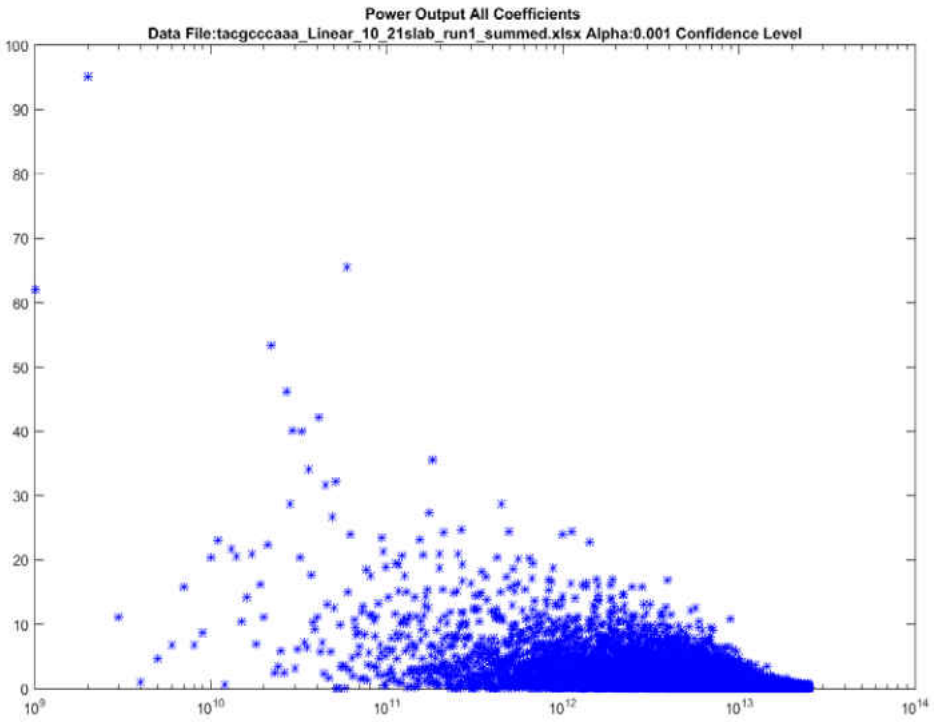


Figure 3: All Coefficient Power Output of Example System

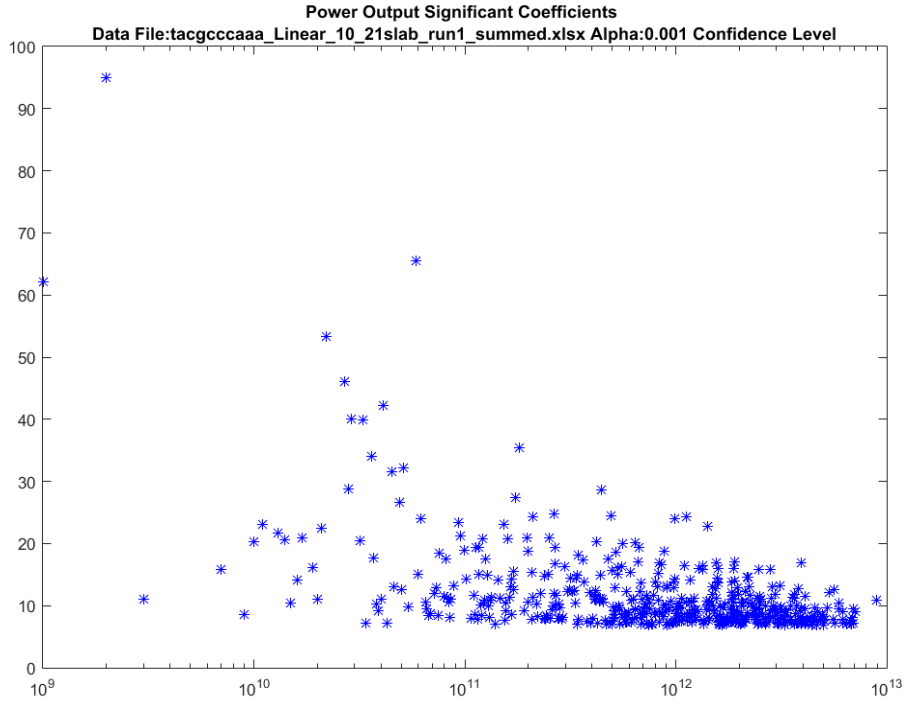


Figure 4: Significant Coefficient Power Output of Example System

The ingested pressure profile data was subjected to FFT, and then the resulting coefficients were converted to power. Before we continue, it is important to understand the difference between raw Fourier coefficients and the power of that coefficient. Fourier coefficients are composed of a real and an imaginary number. The real number is the quantity of that particular frequency that is present, while the imaginary number is the phase shift of that particular frequency. In this way a Fourier transform represents the incoming time-domain signal in the frequency domain. This frequency domain decomposition enables identification of frequency content by energy levels even in the presence of extremely noisy signals. The power is a positive number (unlike the Fourier results which could be negative or positive depending on phase angle) that represents the amount of energy being output by the process at that frequency. This number is obtained by normalizing the coefficients, a process informed by Parseval's Theorem, that involves summing the squares of the Fourier coefficient amplitudes with the square of the average and dividing by two (Thibos, 2003). As an equation, the power p_k for the kth frequency is:

$$p_{\mathrm{k}} = \left(\hat{\mathrm{a}}_k^2 + \hat{\mathrm{b}}_k^2\right)/2 \qquad (8)$$

Because this variable is, in the statistical sense, standardized, and because we know this to be a Gaussian variable (because it is Fourier transform of the standardized pressure profile output, which comes from a process that is at its root, Gaussian), we know that it has a mean of zero and a variance of one and will be distributed chi-square with a single degree of freedom.

This fact is what enables the application of Hartley's test in this situation. Hartley's test examines the null hypothesis that the amplitude of the k-th Fourier harmonic is zero. It does so by generating a test statistic, often called the H statistic, and comparing that against an F distribution. One will recall that an F distribution is essentially a chi-squared distribution after

normalization. Thibos derives an equation from Hartley's test that is easily applied to test whether the k-th Fourier harmonic is zero. (Thibos, 2003).

$$H = \frac{p_k}{\frac{1}{R}\sum_{j \neq k} p_j} \sim F_{2,2R} \quad (9)$$

The H-statistic is therefore the power of the k-th harmonic (p_k) divided by the quantity (1/R), times the sum of all powers except p_k, where R=(D-3)/2 and represents the total power of the residuals (that is, all power that is not accounted for in p_k), and D is number of samples. The values for F can be looked up in an F table for the desired α (or significance level). In the case of these experiments, the objective of using Hartley's test is to provide a list of candidate significant frequencies that can be examined by comparison of their coefficients to determine whether significance exists in between runs and in between system configuration (linear, parallel, length of DNA sequence, makeup of DNA sequence) changes.

After calculating the statistically significant frequencies for each system, a set of Excel workbooks were created with a custom macro function that would sort the frequencies in each column and then output those that matched across each run. There were only a few matching frequencies in each set of runs, a fact that was somewhat surprising given that each set of runs was that of an identical system configuration, only a different random seed was applied. The summary tables below show those matching frequencies and notes any overlap (with highlighting) between the 10-, 12-, and 16-mer systems in the linear end-to-end and parallel cases, respectively.

Table 1: Linear System Matching Statistically Significant Frequencies

10-mer	12-mer	16-mer
2000080003 Hz	1000040002 Hz	6000240010 Hz
15000600024 Hz	2000080003 Hz	11000440018 Hz
27001080043 Hz	12000480019 Hz	
28001120045 Hz	20000800032 Hz	
39001560062 Hz		
1.42006E+11 Hz		
2.4001E+11 Hz		
9.82939E+12 Hz		
9.9554E+12 Hz		
1.07414E+13 Hz		

Table 2: Parallel System Matching Statistically Significant Frequencies

10-mer	12-mer	16-mer
6000240010 Hz	4000160006 Hz	4000160006 Hz
10000400016 Hz	5000200008 Hz	8000320013 Hz
22000880035 Hz	30001200048 Hz	13000520021 Hz
26001040042 Hz	40001600064 Hz	14000560022 Hz
37001480059 Hz	45001800072 Hz	28001120045 Hz
45001800072 Hz	47001880075 Hz	38001520061 Hz
71002840114 Hz	1.03004E+11 Hz	48001920077 Hz
74002960118 Hz	1.10004E+11 Hz	57002280091 Hz
83003320133 Hz	1.22005E+11 Hz	69002760110 Hz
1.14005E+11 Hz	1.29005E+11 Hz	1.21005E+11 Hz
1.15005E+11 Hz	1.73007E+11 Hz	1.26005E+11 Hz
1.42006E+11 Hz		1.35005E+11 Hz
2.86011E+11 Hz		1.62006E+11 Hz
		1.97008E+11 Hz

As can be seen from the above tables, there was only one overlapping frequency between the systems in the linear case, and two overlapping frequencies in the parallel case, which appeared to support rejecting the null hypothesis that there were no statistically significant differences between the various strand lengths and their pressure profile outputs. However, one must consider carefully two questions: why were there so few significant frequencies that matched among all 20 runs (considering that there is a pool of 25,000 frequencies to choose

from, and nearly 700 significant frequencies from each run), and why were those frequencies at such low overall power from the rest of the system? As can be seen in the table below, the percentage of power accounted for by these matching coefficients is exceedingly small.

Table 3: Power in Matching Coefficients Comparison

Case	Total System Power	Power in Significant Coefficients	Power in Matching Coefficients	% of Total System Power
Linear 10	24990.411	11959.73	202.25	0.81%
Linear 12	24955.292	11820.46	150.10	0.60%
Linear 16	24810.643	11256.69	54.284	0.22%
Parallel 10	24998.294	11489.34	335.611	1.34%
Parallel 12	24998.132	11949.02	236.550	0.95%
Parallel 16	24994.381	11763.03	182.23	0.73%

The power in the few matching coefficients is less than the power in the significant coefficients by a factor of nearly 100 in most cases. In the best case, the total percentage of power accounted for by the matching coefficients is slightly more than 1%. At such a low power level, we cannot discount the possibility that the matching coefficients are themselves an error. The other significant finding is that the power in significant coefficients is less than half the total system power. This suggests that the underlying model implied by Hartley's test (that the system is strongly not Gaussian) is in some way flawed for the purposes of analyzing this system.

In order to help answer this emergent concern that the significant coefficients are actually noise, or that the underlying model used by Hartley's test is flawed for this use case, it was decided to test the significant matching frequencies for randomness using the runs test. The runs test for randomness, as described in Bradley (1968), is a nonparametric, distribution-free test that tests the null hypothesis that a run of data is random. A rejection of the null at the desired significance level would indicate that the data is not likely to be random. An example of this would be analyzing flips of a coin, if there were long 'runs' of heads or tails, then the test would

reject the null hypothesis, indicating that the process appears to be non-random. For each matching significant frequency, all 20 runs of that frequency coefficient were fed into the runs test to provide a post-hoc assessment of randomness. The results are summarized in the table that follows.

Table 4: Results of Runs Test for Randomness

System Configuration	System Length	Frequency	p-value	Reject null?
Linear	10	2000080003 Hz	0.4768	No
Linear	10	27001080043 Hz	0.3029	No
Linear	10	28001120045 Hz	0.5980	No
Linear	10	15000600024 Hz	0.2298	No
Linear	10	39001560062 Hz	0.0898	No
Linear	10	1.42006E+11 Hz	0.5099	No
Linear	10	2.4001E+11 Hz	0.5932	No
Linear	10	9.82939E+12 Hz	0.8599	No
Linear	10	9.9554E+12 Hz	0.3182	No
Linear	10	1.07414E+13 Hz	0.7570	No
Linear	12	1000040002 Hz	0.1348	No
Linear	12	2000080003 Hz	0.9600	No
Linear	12	12000480019 Hz	0.4539	No
Linear	12	20000800032 Hz	0.2298	No
Linear	16	6000240010 Hz	0.2157	No
Linear	16	11000440018 Hz	0.5980	No
Parallel	10	6000240010 Hz	1.000	No
Parallel	10	10000400016 Hz	1.000	No
Parallel	10	22000880035 Hz	1.000	No
Parallel	10	26001040042 Hz	0.2316	No
Parallel	10	37001480059 Hz	1.000	No
Parallel	10	45001800072 Hz	0.9600	No
Parallel	10	71002840114 Hz	0.0898	No
Parallel	10	74002960118 Hz	1.000	No
Parallel	10	83003320133 Hz	0.5932	No
Parallel	10	1.14005E+11 Hz	0.4539	No
Parallel	10	1.15005E+11 Hz	0.9600	No
Parallel	10	1.42006E+11 Hz	0.5980	No
Parallel	**10**	**2.86011E+11 Hz**	**0.0125**	**Yes**
Parallel	12	4000160006 Hz	0.2316	No
Parallel	12	5000200008 Hz	0.7570	No
Parallel	12	30001200048 Hz	1.000	No
Parallel	12	40001600064 Hz	0.7570	No
Parallel	12	45001800072 Hz	0.8599	No

System Configuration	System Length	Frequency	p-value	Reject null?
Parallel	12	47001880075 Hz	0.8600	No
Parallel	12	1.03004E+11 Hz	0.8599	No
Parallel	12	1.10004E+11 Hz	0.0492	No
Parallel	12	1.22005E+11 Hz	0.5980	No
Parallel	12	1.29005E+11 Hz	0.5165	No
Parallel	12	1.73007E+11 Hz	0.0702	No
Parallel	16	4000160006 Hz	0.3652	No
Parallel	16	8000320013 Hz	0.4768	No
Parallel	16	13000520021 Hz	0.1894	No
Parallel	16	14000560022 Hz	0.4164	No
Parallel	16	28001120045 Hz	0.5444	No
Parallel	16	38001520061 Hz	0.4164	No
Parallel	16	48001920077 Hz	0.4539	No
Parallel	16	57002280091 Hz	0.3029	No
Parallel	16	69002760110 Hz	0.9600	No
Parallel	16	1.21005E+11 Hz	0.1348	No
Parallel	**16**	**1.26005E+11 Hz**	**0.0204**	**Yes**
Parallel	16	1.35005E+11 Hz	0.8599	No
Parallel	16	1.62006E+11 Hz	0.8599	No
Parallel	16	1.97008E+11 Hz	0.7570	No

Out of 54 matching significant coefficients, only two were nonrandom at the 95% confidence level. This finding, coupled with the previous findings, strongly suggests that the significant matching coefficients are in fact "loud" noise rather than the signal this research is searching for.

Because of these significant power spectrum shifts between repeated runs, the low total percentage of total system power accounted for by significant coefficients, and because of the low number of matching coefficients, it is highly likely that the underlying process is either non-stationary or has a longer period than what was captured. This was unexpected due to the relatively high frequencies (around 10^{12} Hz) of the resonant spectra of the component parts (water and DNA nucleotides) of the system. Given this evidence, it is clear that the hypothesis $\mathbf{H}_{(\text{Length NULL})}$ cannot be rejected by this experiment.

Experiment 2: Test of Hypothesis 2: DNA's Vibrational Motion is Dependent on Strand Content

In a similar vein to the experiment for testing $\mathbf{H}_{(\text{Length NULL})}$, the setup for this experiment proceeded in an identical manner to that described previously with one exception: 20 runs of each system configuration were undertaken, each one having a randomly generated DNA sequence. The goal was to determine whether changes in the DNA sequence caused significant changes in the power spectrum, and the null hypothesis is referred to as $\mathbf{H}_{(\text{Sequence NULL})}$, that is, that there are no significant differences between the power spectrums of differing strand sequences. The alternative hypothesis is that there is evidence to support the conclusion that strand sequence affects the power spectrum. Due to simulation runtime constraints, this experiment was not designed with multiple repeated runs as Experiment 1 was. It was considered that Experiment 1 would shed sufficient light on the process's properties in order to help make the determination of the further experiments being conclusive.

An unexpected issue with this experiment was the large number of significant changes in the power spectrum between repeated runs in experiment 1. These were runs that had no changes between them save for the random seed that initialized the simulation. The high level of noise in the significant frequencies severely clouds any signal that might be present. This means that we must temper any expectations of statistically significant results with the high likelihood that any changes seen between systems may be from random chance, rather than from actual changes in the power spectrum.

Statistical Analysis for Experiment #2

In comparing the inter-run frequencies, it was necessary to first verify that the significant frequencies found between each run were a statistically significant departure from error. To that

end, a 2x2 chi-square table was constructed to test the number of significant coefficients (against the total) vs the number of coefficients we would expect to be error at a 99.9% confidence interval (or having less than 0.01% error). The lowest number of significant coefficients returned in each case (linear and parallel) was used, since if it was significant, it would obviate the need to test the other 19 runs. In both cases, the analysis code was initialized using a random Gaussian distribution with the same mean and variance as the linear or parallel system, respectively. The number of significant coefficients were noted, and then 2x2 chi-square tests were run to check against the number of significant coefficients in the run with the fewest significant coefficients. The results are shown graphically in the figures that follow.

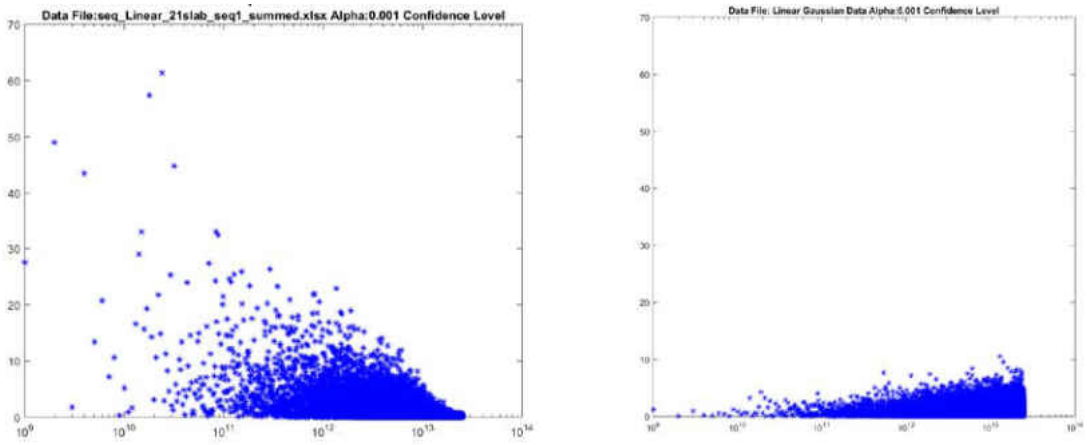


Figure 5: Power Spectrum Linear Case (Left) and Gaussian Data (Right)

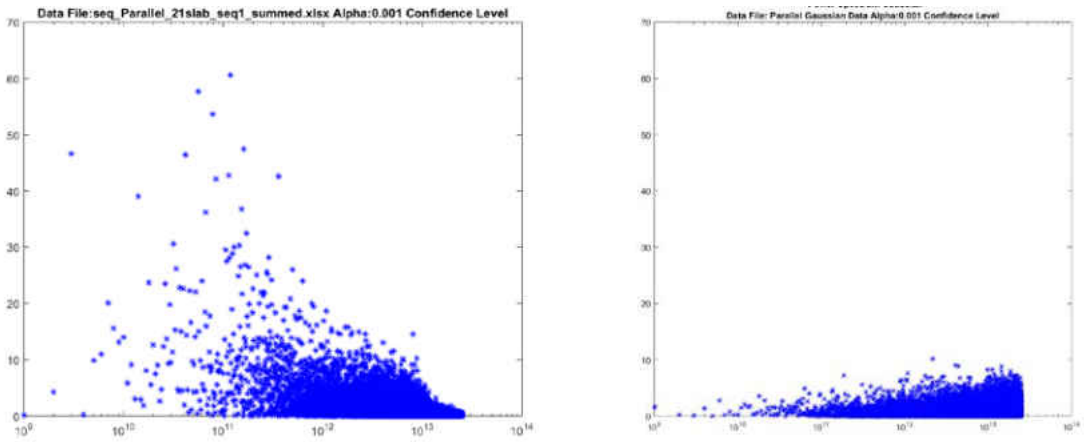


Figure 6: Power Spectrum Parallel Case (Left) and Gaussian Data (Right)

	Gp 1	Gp 2	Gp 3	Gp 4	Gp 5	Gp 6	Gp 7	Gp 8	Gp 9	Gp 10	
Cond. 1:	17	468									485
Cond. 2:	24982	24531									49513
Cond. 3:											0
Cond. 4:											0
Cond. 5:											0
Cond. 6:											0
Cond. 7:											0
Cond. 8:											0
Cond. 9:											0
Cond. 10:											0
	24999	24999	0	0	0	0	0	0	0	0	49998

Output:

Calculate | Reset all

Chi-square: 423.492

degrees of freedom: 1

p-value: 0

Yates' chi-square: 421.616

Status: Status okay

Yates' *p*-value: 0

Figure 7: Linear Case Chi-Square Test Against Random System

	Gp 1	Gp 2	Gp 3	Gp 4	Gp 5	Gp 6	Gp 7	Gp 8	Gp 9	Gp 10	
Cond. 1:	23	465									488
Cond. 2:	24976	24534									49510
Cond. 3:											0
Cond. 4:											0
Cond. 5:											0
Cond. 6:											0
Cond. 7:											0
Cond. 8:											0
Cond. 9:											0
Cond. 10:											0
	24999	24999	0	0	0	0	0	0	0	0	49998

Output:

Calculate | Reset all

Chi-square:	404.282
degrees of freedom:	1
p-value:	0
Yates' chi-square:	402.455
Yates' *p*-value:	0

Status: Status okay

Figure 8: Parallel Case Chi-Square Test Against Random System

As can be seen, there is a clear difference between the Gaussian random data and the representative system output. Both systems represented statistically significant departures from the random case. In both cases, the number of matching variables in the random case (17 and 23) closely matched the expected value if there had been 0.01% errors (25), and this therefore indirectly serves to validate the expected error rate of Hartley's test.

While there is a statistically significant difference between the quantity of significant coefficients in each system and randomness, the question is whether there are significant differences between those coefficients. The most obvious method is to look at the sparse points of the overlapping systems diagrams and derive a confidence interval. If there are other points present inside that envelope, then it is not unique. To begin, each of the 20 systems are compiled into a graph charting power vs frequency on a semilog plot. From there, the point cloud was studied for outliers, and it is then possible to compute the distance from those outlying points to

all other points and check to see if they are within the confidence interval or not. For a frequency to be considered unique, it is necessary for it to not be anywhere closer than the confidence interval distance. This was originally done manually, but was automated to check the entire range quickly.

Just as it is possible to compute a confidence interval for real-valued statistical functions, it is also possible to compute a confidence interval for a vector. This approach is used by Hartley's test (to ensure that no part of the vector crosses the origin), which is a special case of comparing two vectors for significance. (Thibos, 2003) derives the following equation for the confidence interval surrounding a vector point.

$$\rho^2 = \frac{F_{2,2R}}{R} \sum_{j=1}^{R} p_j \qquad (10)$$

In equation 10, ρ^2 is the diameter of the confidence interval, $F_{2,2R}$ is the F-distribution for the desired probability, 2 degrees of lesser freedom and 2R degrees of greater freedom (where R=(D-3)/2 and D is the number of coefficients returned), and where p_j is the sum of powers on the positive side of the Fourier spectrum returned. Taking the square root of this function returns the radius of the confidence bound around the point. From this, we have a test we can employ: if any point of any sequence's coefficient is within that confidence interval bounds, we cannot reject the null hypothesis that they are statistically the same point. To do this, a simple subtraction calculation can be used to calculate the Euclidean distance between two points: once for the real value, and once for the complex value. If any of the distances between the points falls within the confidence bound, we will fail to reject the null hypothesis for that point. If none of the point test reject the null hypothesis, then we can say, within the confidence interval, there is insufficient evidence to support rejecting $\mathbf{H}_{(\text{Sequence NULL})}$. The comparison itself is fairly straightforward. Using Matlab, it is possible to automatically calculate the Euclidean distances

between every point in a given vector or matrix. The custom code scans through each frequency, then each point, comparing each point to every other point. If a point has zero distances that are less than the confidence interval distance, then that point is significantly different from all the others at that frequency. For the purposes of this test, it was decided to output the results one frequency at a time, the reason for this being that, given that there are 20 sequence runs, that is 20C2 (read: "twenty choose two") combinations to test, or 190, per frequency—4.7 million comparisons in total. To winnow the list quickly, the function runs the distance calculation, and writes out a variable and prints to the console if there were no coefficients that lay within the confidence bound at that point (source code is available in the appendix). The next two figures illustrate this graphically, showing the significant coefficients from every system overlaid on the graph showing frequency versus power. One can see a lot of overlap between these systems, with very few "outlier" points (where we would expect to find significance).

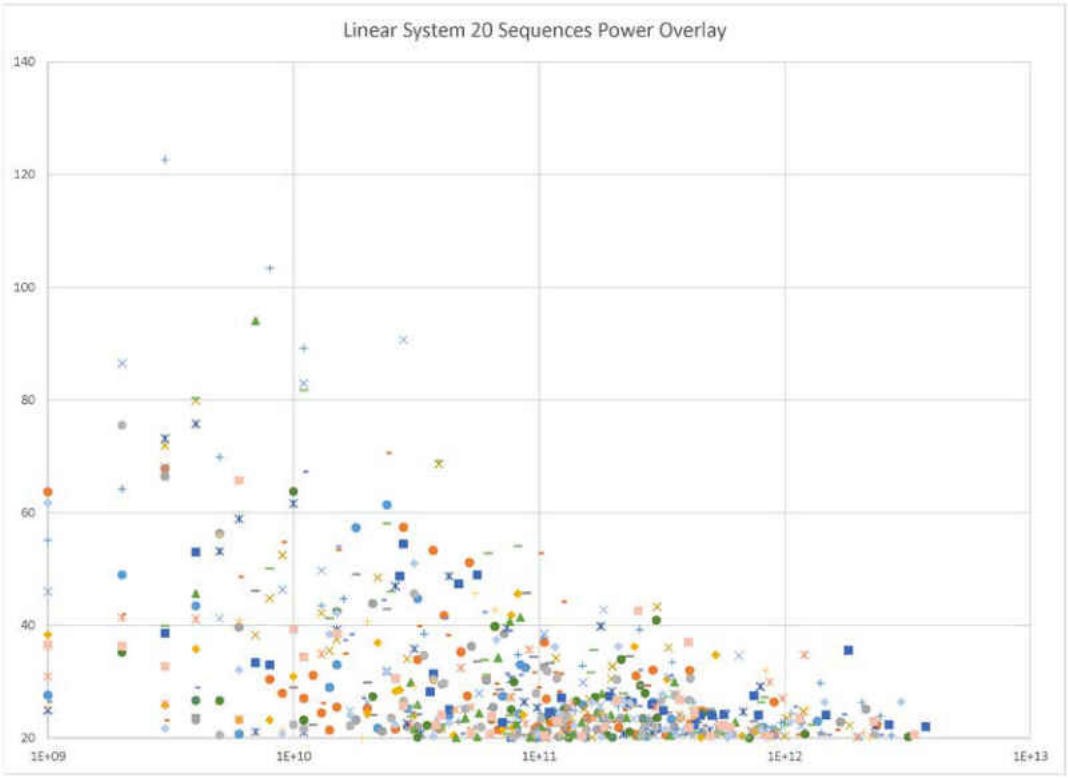


Figure 9: Linear System 20 Sequence Power Overlay Graph

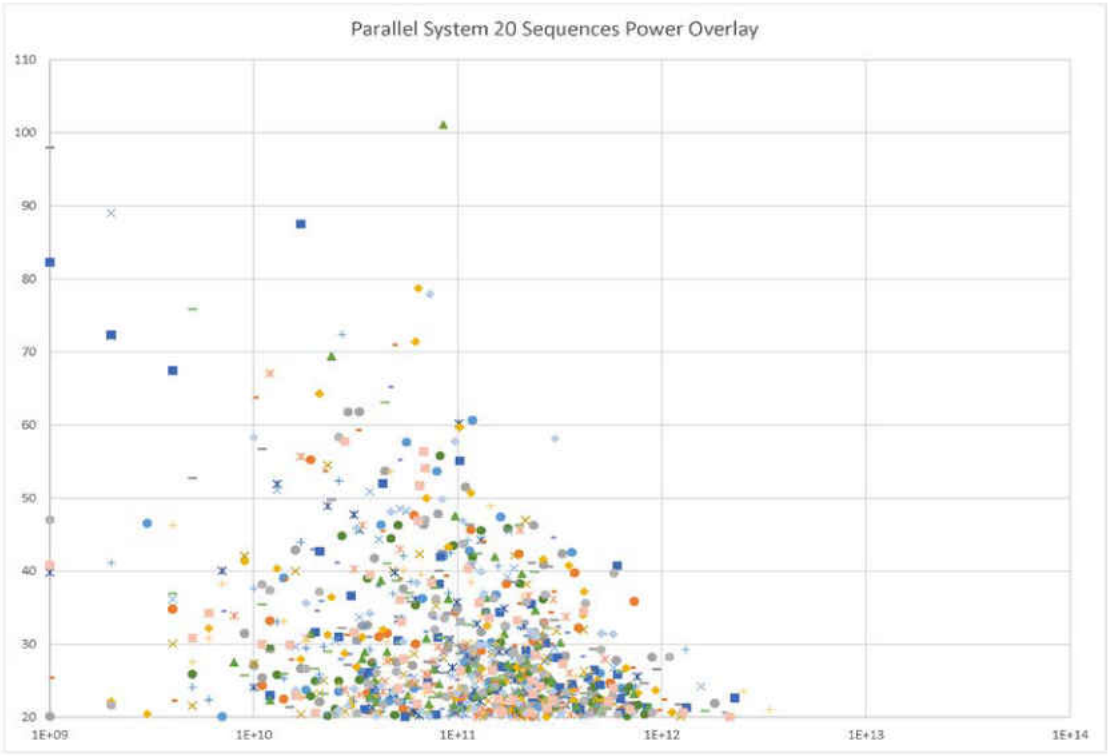


Figure 10: Parallel System 20 Sequence Power Overlay Graph

Indeed, upon automated comparison (output tables are in the appendix), there were found to be 52 unique points in the linear system and 71 unique points in the parallel system. While this is better than no unique points, and while it does suggest that some of these frequencies may be unique, the very small number of significant points (an average of just 3.6 unique frequencies per system, out of 25,000) means that fitting an equation to test the uniqueness theory will not work well—the amount of "error" excluded from the model would be significant—most of the signal. As such, though there is technically sufficient evidence to reject $\mathbf{H}_{(\text{Sequence NULL})}$ at the 99.9% confidence level, such rejection should be taken with the proviso that additional research is needed to confirm the nature of these changes—a much larger study, perhaps with hundreds of runs, could tease out the effect within the significant noise of this system. In the meantime, it would be wise to regard the results of this experiment as inconclusive.

Experiment 3: Test of Hypothesis 3: DNA's Vibrational Motion Uniquely Varies with Strand Length and/or Content Combinations

The data to test this experiment's hypothesis comes from the first two experiments, as this experiment is a brief meta-analysis of the experimental data collected. Null hypothesis 3, abbreviated $\mathbf{H}_{(\text{Unique NULL})}$, is that there is no evidence of a uniquely-identifiable difference between strand configurations. The alternative hypothesis is that we find evidence of a characteristic in the frequency response that will permit unique identification of the different strand configurations. Given the failure to reject $\mathbf{H}_{(\text{Length NULL})}$, and the inconclusive result of $\mathbf{H}_{(\text{Sequence NULL})}$, any rejection $\mathbf{H}_{(\text{Unique NULL})}$ should be interpreted as there being a potential for uniqueness, but in any case, further study is indicated.

In order to consider the question of whether the vibrational motion of the DNA being studied changes in a unique way, a special subtype of Fourier analysis, cross power spectral density (CPSD) analysis, was undertaken. The CPSD of a signal is, in short, the power spectral density (as employed in experiment 1 and 2's power analyses) compared with another signal using cross-correlation. Signals will have a high coefficient of CPSD where they share frequencies, and a low coefficient where they do not. By looking at peaks and valleys of the CPSD, one can determine what portions of the frequency response are shared, and what portions are different. If there is uniqueness in the power spectrum of the various cases, we would expect to see valleys in the areas where those unique frequencies lie. The particular method used for CPSD analysis here is Welch's method, which, due to its overlapping function, helps reduce the effects of noise (Welch, 1967), and we have already seen that these datasets are extremely noisy. No special code was needed to output the comparison between systems, as Matlab implements the CPSD as a function call in the Signal Processing Toolbox.

The CPSD was run among systems in each configuration to compare each to the other (10-mer to 12-mer, 10-mer to 16-mer and 12-mer to 16-mer). The graphical results of those runs are shown in the figures below.

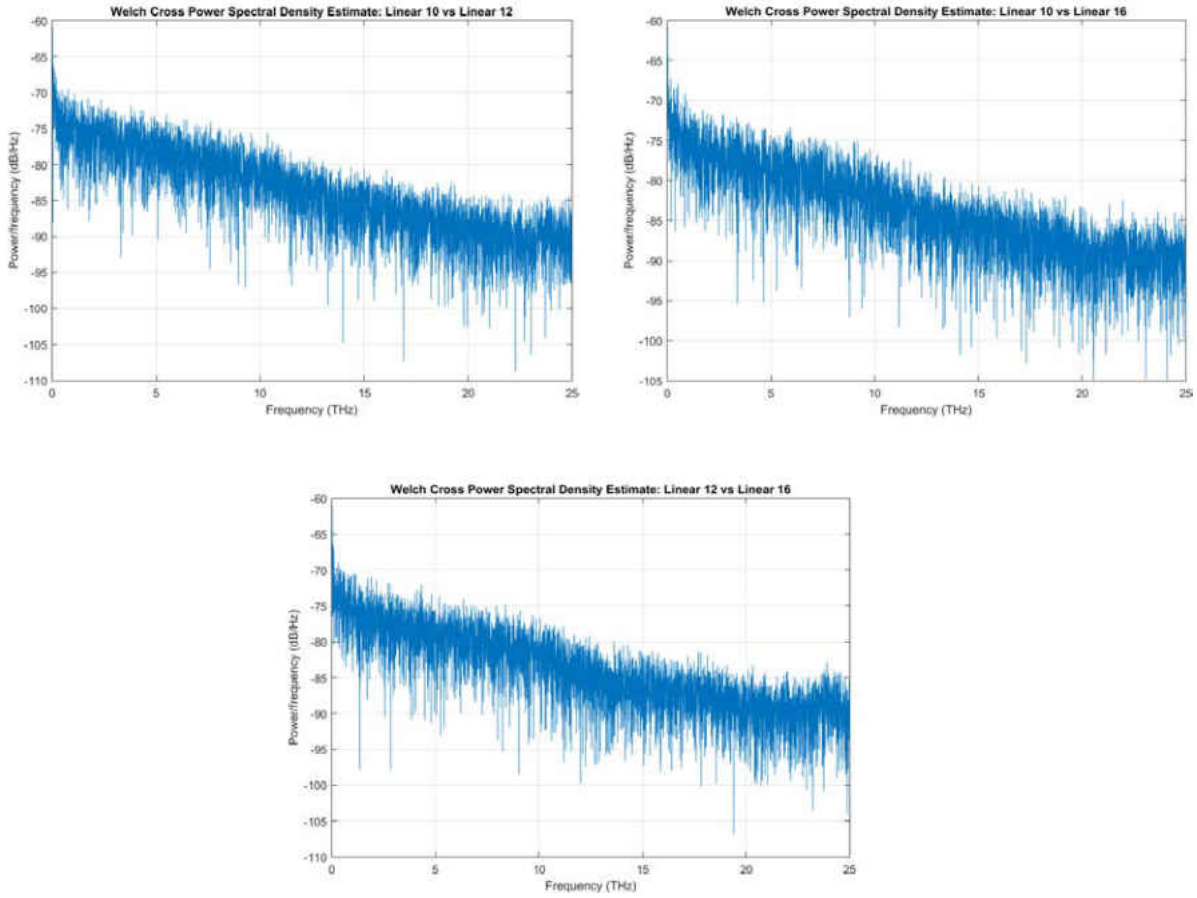


Figure 11: Cross Power Spectral Density Graphs (Linear Configuration)

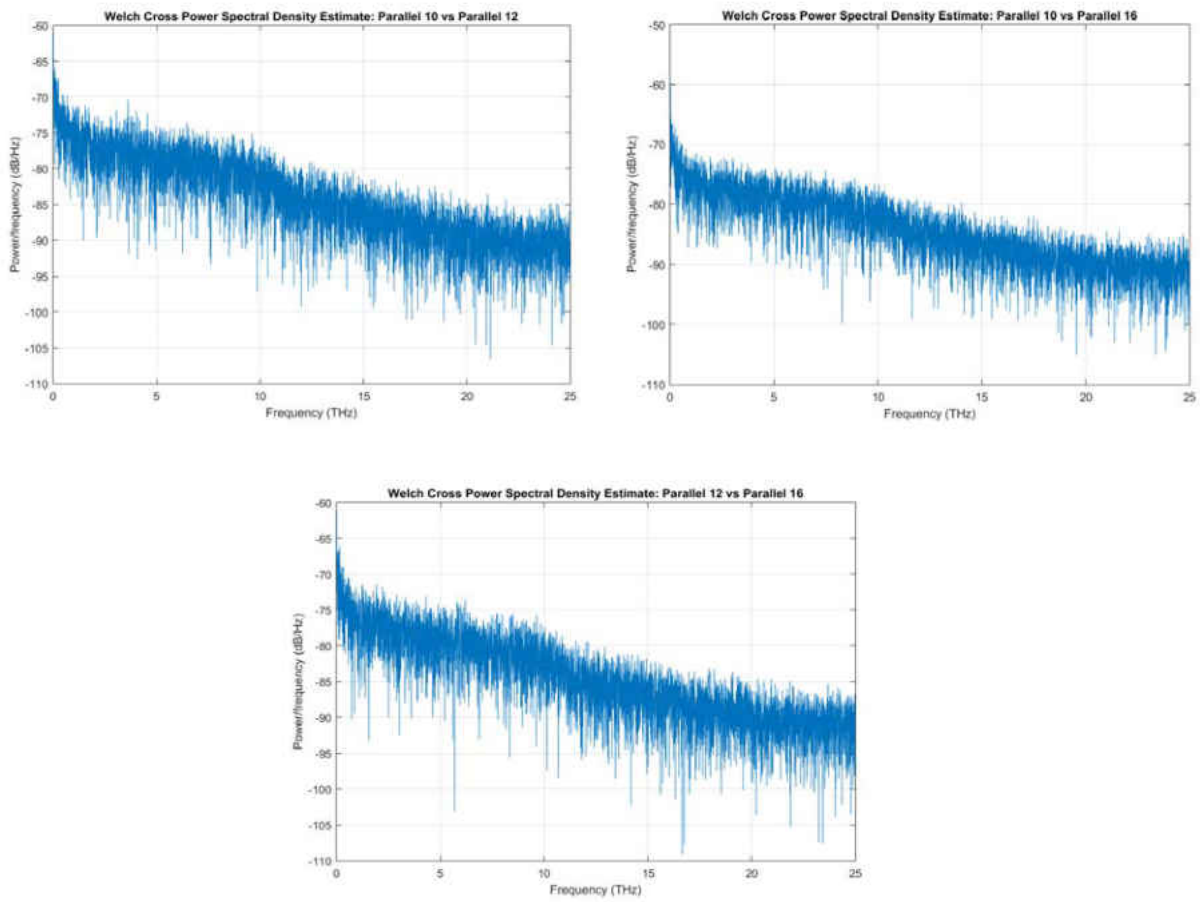


Figure 12: Cross Power Spectral Density Graphs (Parallel Configuration)

From these comparison graphs, several features become apparent. First, as seen in the previous figures from experiment 1, the power spectrum starts off at a relatively high power and falls to a relatively low power by the end of the studied range. Besides this obvious linearity, there are only a few small dips that indicate differences—in the linear systems, these occur around 12-13 THz and around 20 THz; in the parallel systems, this occurs around 12-13 THz. Besides these two small jogs downward, the trend lines of these CPSDs are very similar. More importantly, those small dips were much smaller in amplitude change than the noise of the system as a whole. This is significant because it means that any model fit to these CPSDs will not easily capture the true nature of any underlying process. While it may be possible to discern between the linear and the parallel cases by examining the CPSD charts and noting the difference in slope and

oscillation between cases, it is not possible to make any determinations between system configurations. Due to the inability to clearly discern a trend in the CPSD analysis, we must consider the results of $\mathbf{H}_{(\text{Unique NULL})}$, to be inconclusive.

Experiment 4: Test of Hypothesis 4: DNA's Vibrational Motion Forms a Mathematical Relationship

Finally, in this experiment, we sought to understand whether a mathematical relationship between the DNA molecules' vibrations and their Fourier coefficients exists. The null hypothesis, to be called $\mathbf{H}_{(\text{Relation NULL})}$, is that there is no evidence to support a mathematical relationship, with the alternative hypothesis that there is evidence to support the conclusion that a mathematical relationship exists between the vibrations and their resulting Fourier coefficients. In order to determine whether a mathematical relationship exists between either strand length or strand sequence, the process of Fourier fitting was undertaken. Fourier fitting is very much like linear or exponential model fitting, only with Fourier coefficients that describe how a system oscillates rather than with simple linear equations (Ramsay, Graves, & Hooker, 2009; Thibos, 2003). The output comprises pairs of terms which are used to drive alternating sine and cosine pairs (one pair per term). Fourier fitting, much like other methods of generating a model, supply us with confidence intervals for the coefficients which can be examined to determine whether any of the parameters should be excluded due to not being statistically significant. There are some limitations to Fourier fitting, generally the maximum number of coefficients is very low due to computation time required, and it tends to not work well in systems that are very noisy. Six comparisons were undertaken, with the known sequences TACGCCCAAA, TACGCCCAAACT, and TACGCCCAAACTAGCC (the 10-, 12-, and 16-mer strands), in both the parallel and linear configurations. These known sequences were used because of the large

number of experimental runs available from which to draw inferences. Each sequence was linked to itself (doubling its length) and simulated using the same procedure as experiment 1. As such, six comparisons were possible; two for each case, looking at the original and the linked cases in both linear and parallel. Each comparison outputs a list of coefficients (available in the appendix) and two graphs showing the fitted model and the original pressure data. The figures are below.

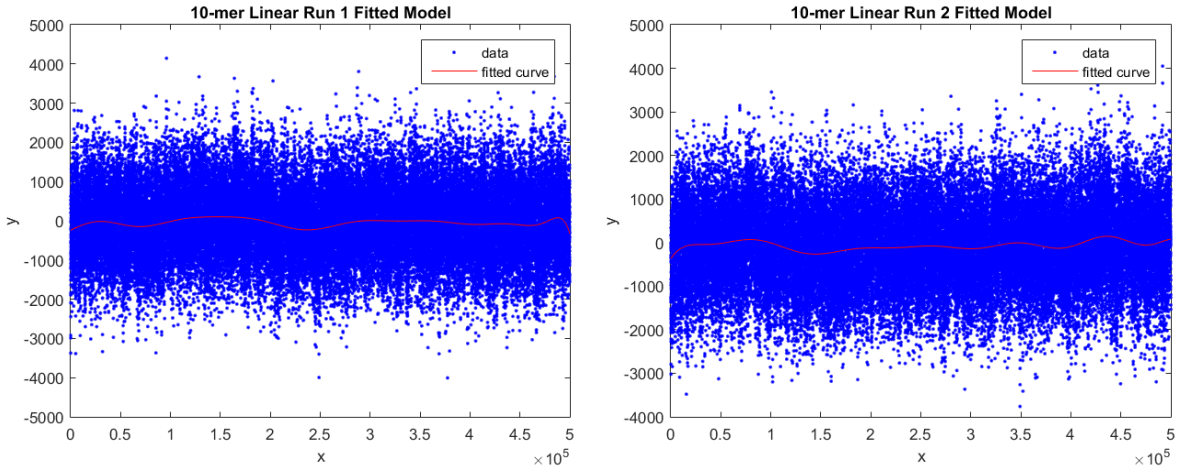


Figure 13: 10-mer Linear Fitted Fourier Models

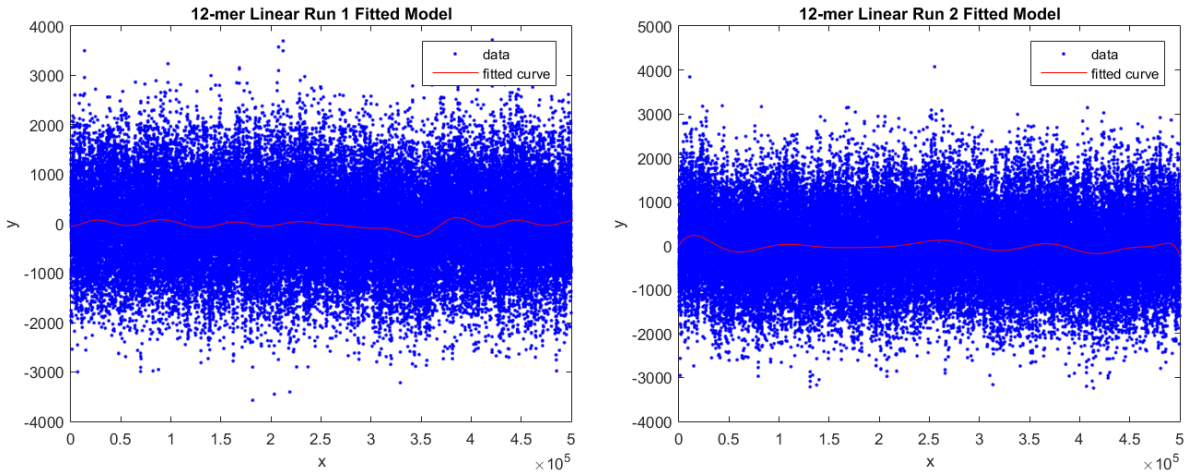


Figure 14: 12-mer Linear Fitted Fourier Models

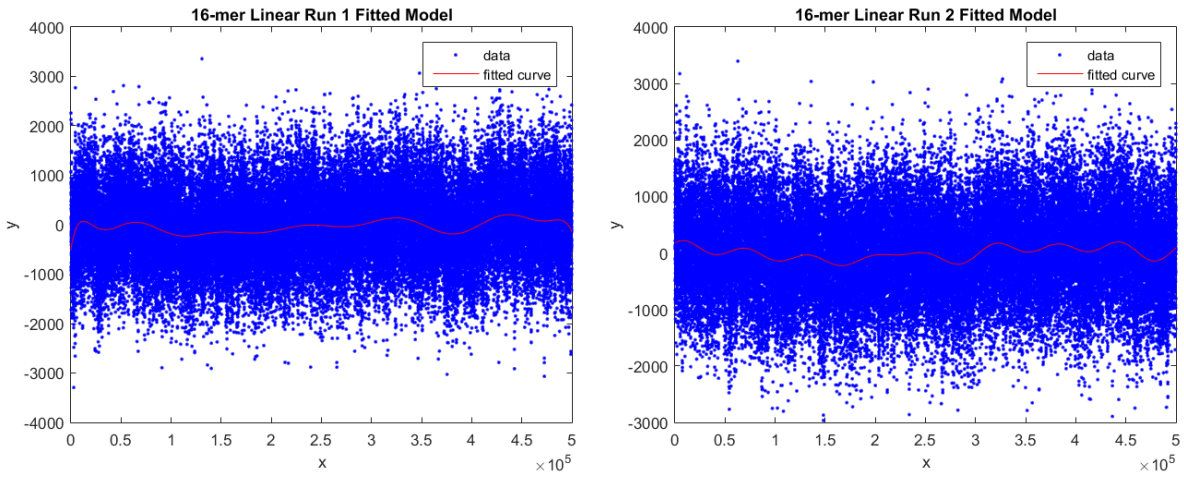


Figure 15: 16-mer Linear Fitted Fourier Models

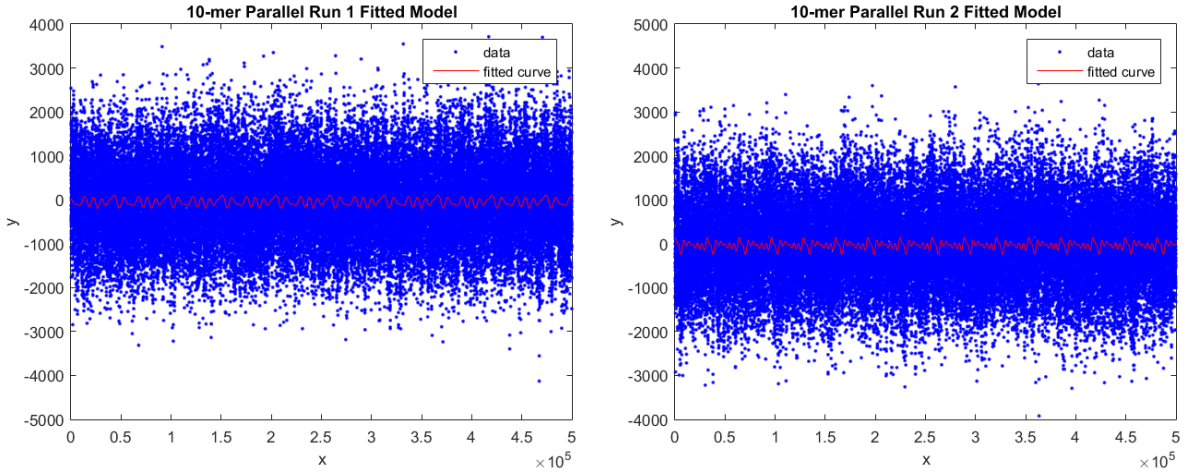


Figure 16: 10-mer Parallel Fitted Fourier Models

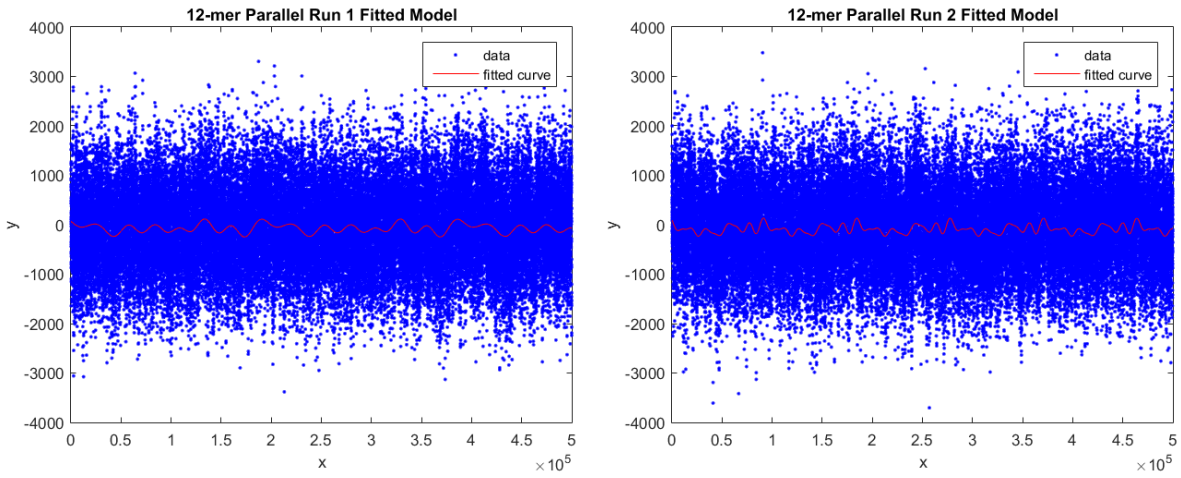


Figure 17: 12-mer Parallel Fitted Fourier Models

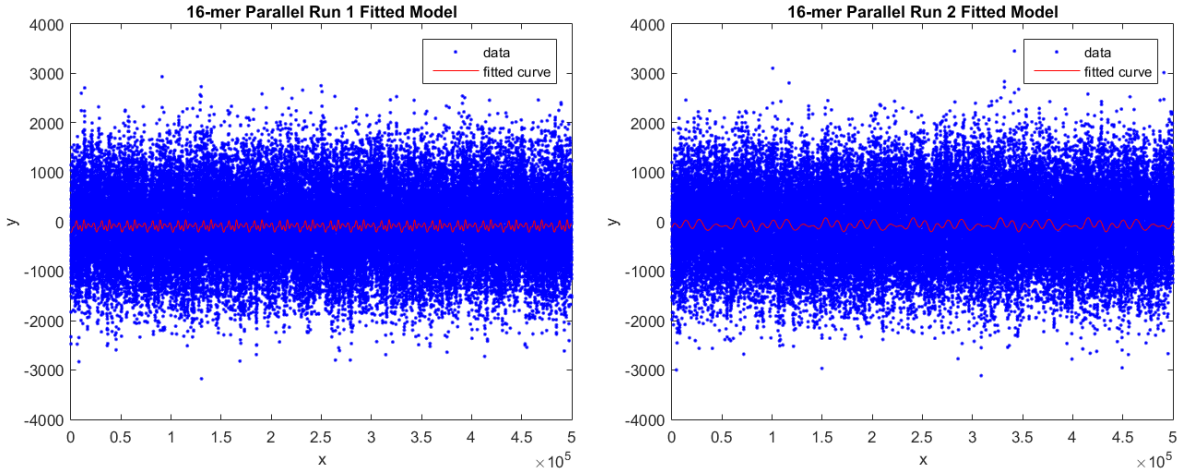


Figure 18: 16-mer Parallel Fitted Fourier Models

An eight term model (the maximum available) was fitted to each sequence. However, in every case, most if not all the coefficients' confidence bounds crossed the zero value line. That fact signifies that the coefficient is not statistically significant, because it intersected the origin. Due to the lack of statistically significant coefficients, it is not necessary to run F-tests for fit, as the models were known not to fit within the confidence interval via the intervals provided for each coefficient. Therefore, we can only fail to reject $\mathbf{H}_{(\text{Relation NULL})}$.

Because of this apparent lack of discernable mathematical relationship between the original and linked cases, a post-hoc evaluation using periodograms was employed. Periodograms are Fourier analysis graphs that show any significant periodicities in a system. They are a weighted (by the square of the power) graph that can aid in identifying any oscillations or periodicities in very noisy data. Additionally, periodograms, being a statistical computation, can show confidence interval bounds. For that reason, they are employed here as a check on the above results; they are shown below.

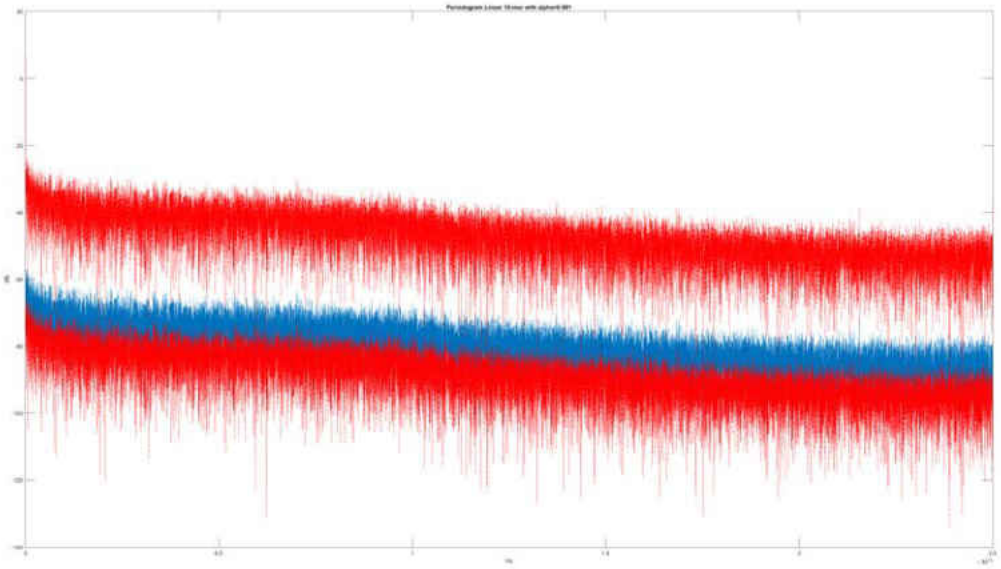


Figure 19: Periodogram 10-mer Linear System

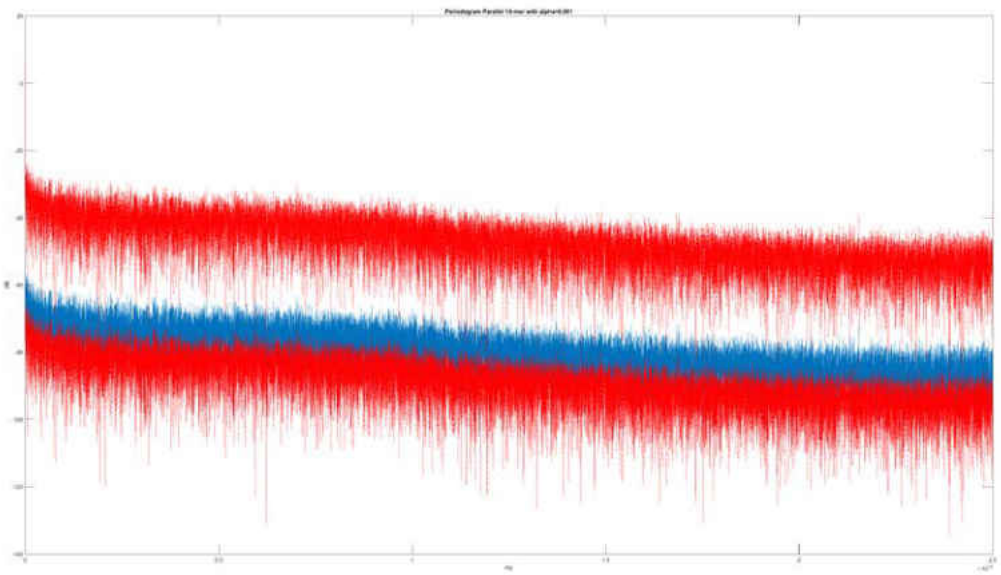


Figure 20: Periodogram 12-mer Linear System

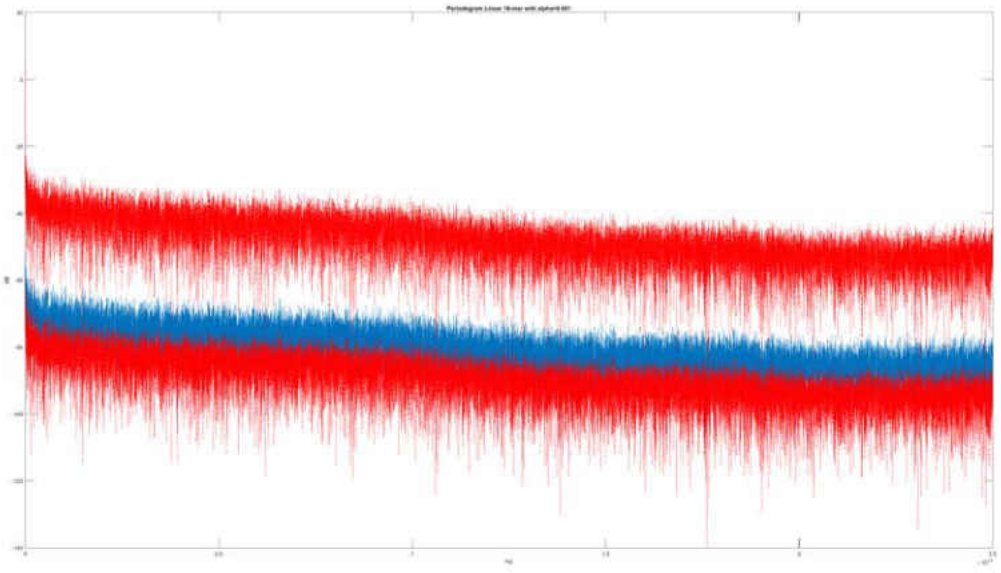


Figure 21: Periodogram 16-mer Linear System

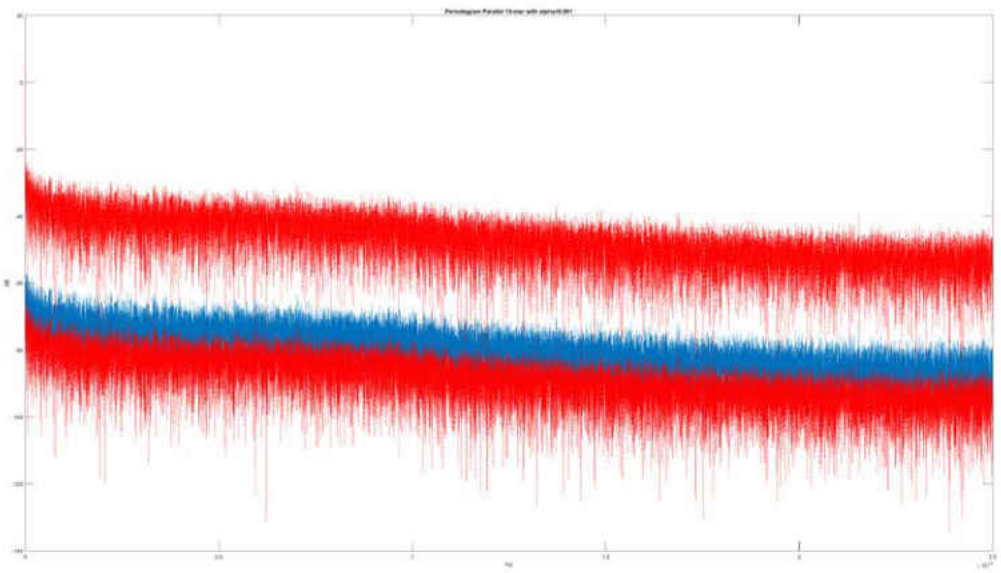


Figure 22: Periodogram 10-mer Parallel System

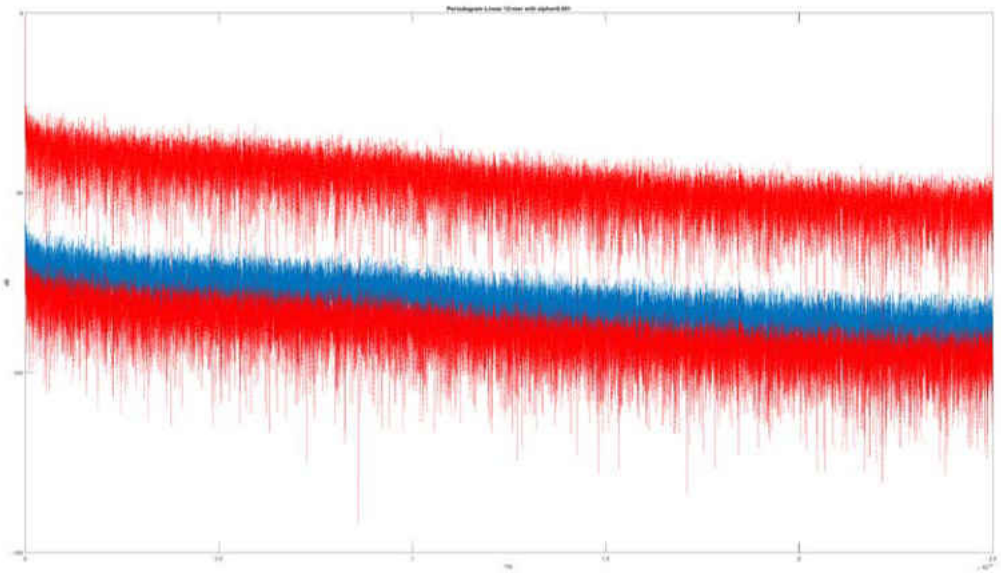


Figure 23: Periodogram 12-mer Parallel System

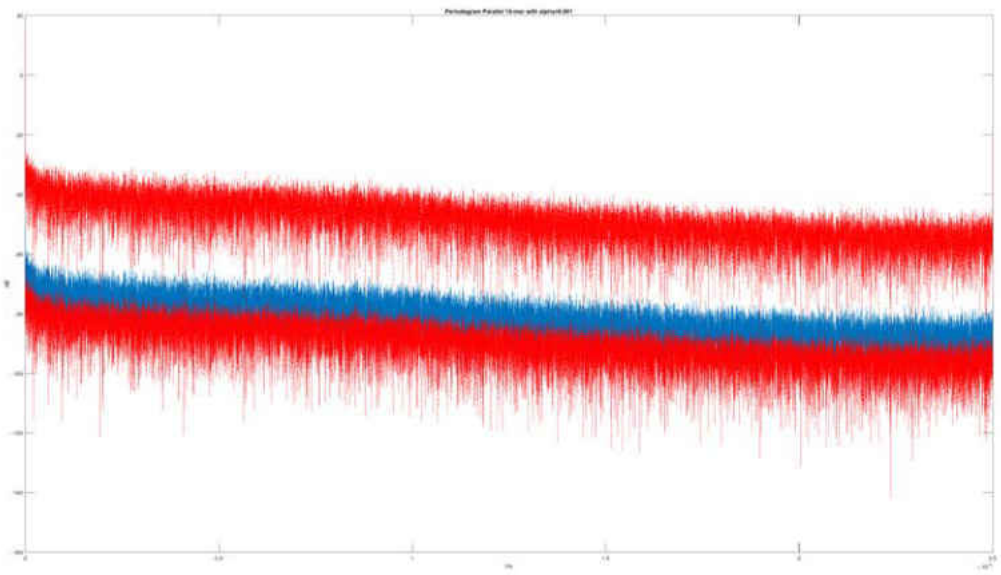


Figure 24: Periodogram 16-mer Parallel System

In these periodograms, the blue line represents the signal and the red lines represent the upper and lower confidence bounds. The x-axis is frequency in Hz and the y-axis is intensity in dB or negative dB. We note several things about each periodograms: general agreement between upper and lower CIs, no crossing of the CIs (which would indicate a significant coefficient), and good agreement without crossing of the CIs by the signal. All of these observations about the periodograms are confirming what was suspected in the fitted models: that there is no significant departure by any coefficient that would be statistically significant if it were fitted to a model.

An Emergent Hypothesis: Water Box as the Source of Noise

The results of Experiments 1 through 4 indicate that while it remained a possibility that there were variations in the signal due to the DNA content, there was too much system noise to make a clear determination whether it was the size and content of the water box alone, or the DNA contents of that water box that were causing the variations. In order to test the emergent hypothesis that the water box created the noise, as well as attempt a post-hoc experiment

whereby the water box noise might be removed or filtered out, a new experiment was devised. This experiment would create water boxes of the same size, configuration, ionization level, and temperature as those used in the previous experiments, but with one key difference: they would not have DNA molecules in them. They would, in effect, be 'control' water boxes for the DNA-carrying water boxes.

In order to create these water boxes, the VMD tool was employed to create water solvent boxes of the exact same dimensions as each of the system configurations, six in total (10-, 12-, and 16-mer, in both parallel and linear configuration). Those solvent boxes were then ionized, minimized, heated, and equilibrated through the exact same processes as used for the previous experiments. Five random-seeded examples per system configuration were constructed, to provide randomized runs and sufficient sample size from which to draw inference from, without unduly extending the experimental time. The random seeds for these examples was chosen to be the same as the original run (so that a water box in the same configuration as the 10-mer linear system would share the random seeds for runs one through five of that system), thus providing statistical pairing. The runs for these 30 new water boxes consumed 1,080 compute hours (64,800 CPU-hours) of time, not including failed runs.

Analysis of these new water boxes was undertaken via statistical comparison as well as spectral comparison, with the goal of determining whether there were any significant differences between the original systems and these new water and ion-only boxes. The results were intriguing. In every case tested, the system output matched up nearly identically at the 95% confidence level, this despite the fact that the system was missing the DNA present in the original experiment—a difference of several thousand molecules. This striking similarity between the DNA-containing water box systems and the water-only systems assisted in

explaining the nature of the experimental results: why there were no discernable differences allowing us to separate the effects of strand length and strand sequence.

Initial attempts centered on using classical filtering techniques such as the LMS (least mean squares) filter (Widrow & Stearns, 1985). This filter belongs to a class known as adaptive filters, which attempt to filter out noise by following a gradient estimate of the error. The filter traverses the signal and attempts to smooth it by locating discontinuities and adjusting its weights adaptively. This type of algorithm can provide insights into the noise of a system when the properties of that noise are not known (such as random Gaussian noise). Applying the LMS filter to these data resulted in a dataset that was still very noisy with respect to any unique signal, and was therefore not usable for comparison purposes. Although not useful for comparisons, filtering the resulting pressure profiles from Experiment 2 did reveal some slight convergences on some frequencies in both the linear and the parallel cases where there was previously no consensus (see Tables 8 and 9 in Appendix A). This convergence suggests that with appropriate filtering, it should be possible to identify those frequency bands which are common in each water box size and subtract them from the signal, reducing the problem space.

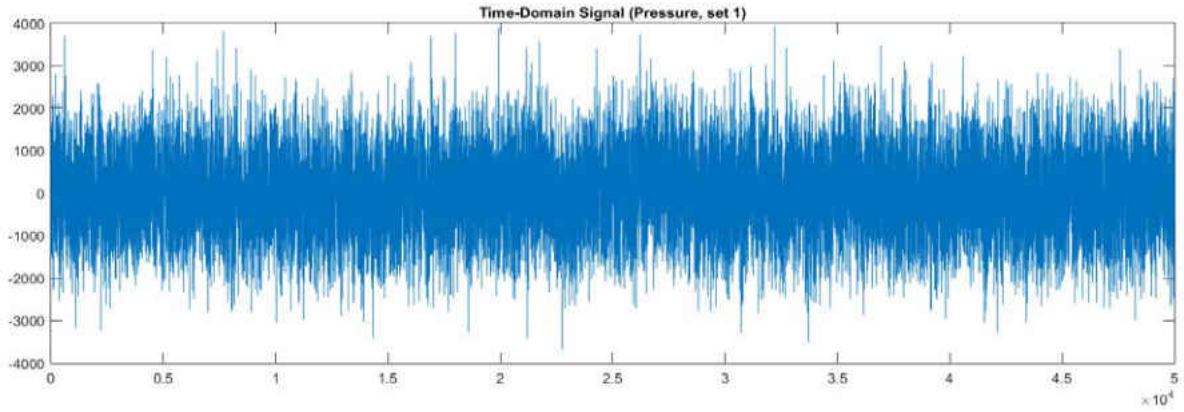


Figure 25: Example Pressure Signal

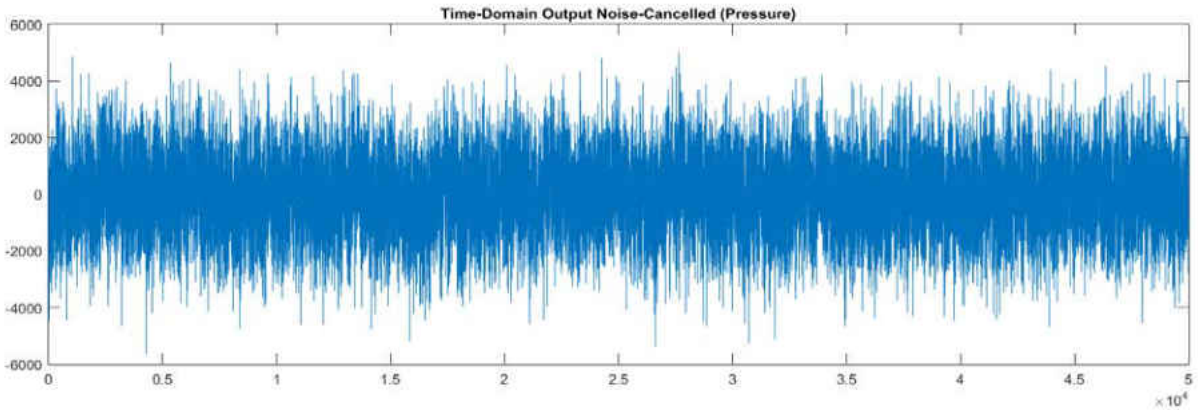


Figure 26: Example Pressure Signal After LMS Filter

The use of the finite impulse response (FIR) Weiner filter (Benesty, Chen, Huang, & Doclo, 2005; Wiener, 1964) was also attempted. This filter differs from the LMS filter in that it can be fed a known noise spectra in order to allow it to better generate an error estimate for the input signal it is attempting to filter. Although often applied to two-dimensional data, such as images, it can also be applied to audio data, and therefore to the sort of pressure data found in these experiments. To employ it, the filter was fed two pieces of data: the known noise signature from the water-only control, and the signal data from the matching water and DNA system. The result, a cross-correlation between the two signals, is output. The results were ultimately unusable due to the input and output signals being sufficiently similar, causing a failure to compute the covariance matrix.

Investigation into that similarity began with a simple spectral comparison between an example DNA-containing system and its water-only counterpart. The spectrograms were strikingly similar, showing similar power density, similar spectral banding, and similar frequency

responses for both systems.

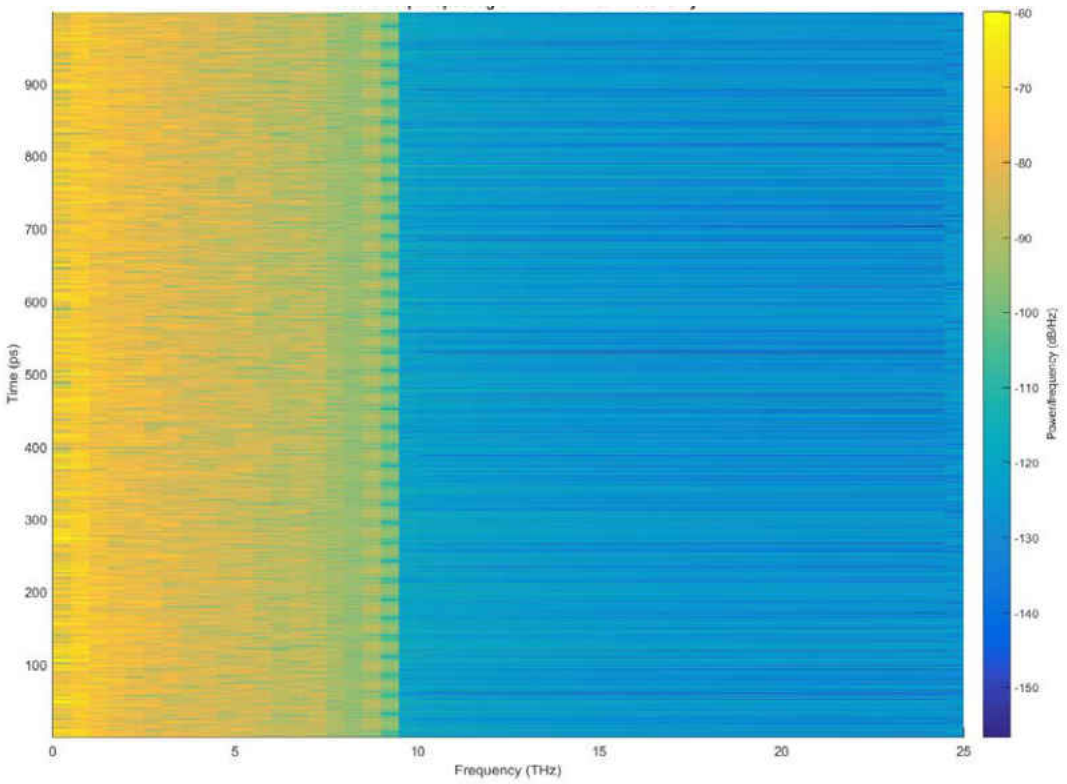


Figure 27: Pressure Output Spectrogram 10-mer Linear Water-Only

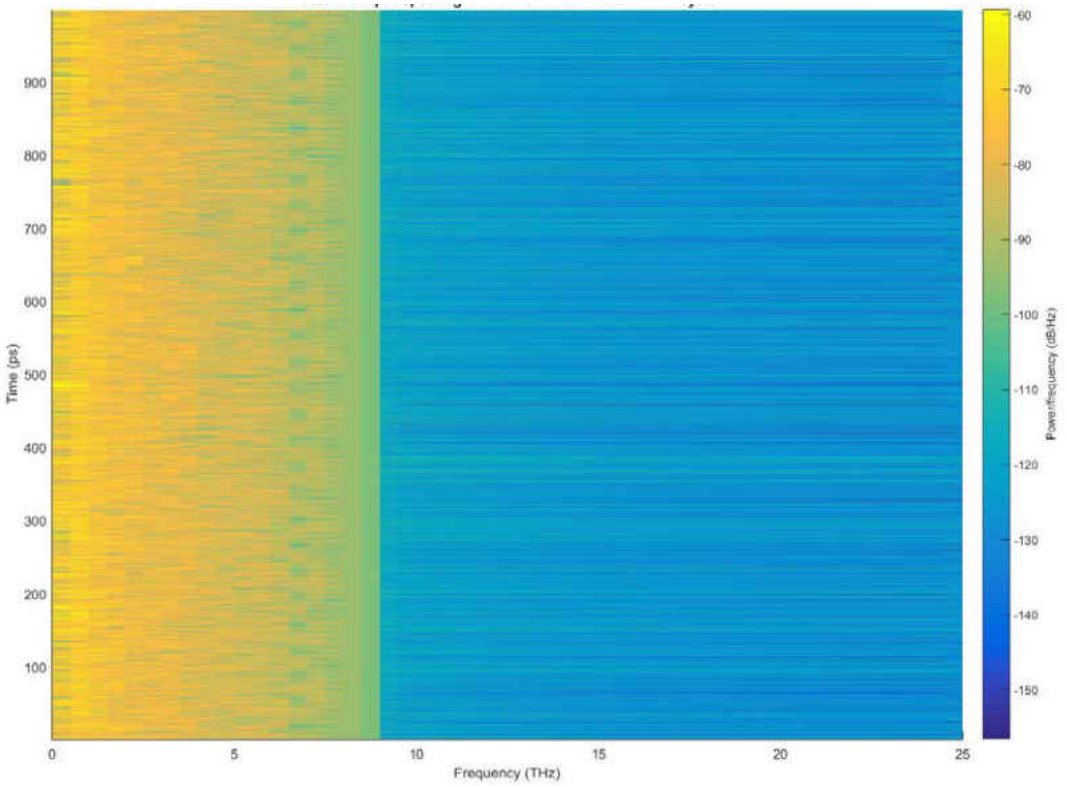


Figure 28: Pressure Output Spectrogram 10-mer Linear Water + DNA System

It was, at that point, reasonably clear that the cause for the inconclusive nature of the four experiments was not due to any issue with motion or lack of motion of the DNA molecules, but rather the intense noise coming from the solvating water itself. A total signal power estimation was carried out to determine what percentage of signal power (and therefore noise) was common between the water-only boxes and the DNA-containing water boxes. The result of that estimation is that, on average, more than 99% of the signal carried over from the water-only boxes to the DNA-containing water boxes. In one case, the water-only system had greater total power than the DNA-containing system, suggestive of other, currently unknown, possibly non-stationary, processes at work in the production of pressures of these systems.

Table 5: Comparison of Water/DNA and Water-Only Power

System Type	Water + DNA Power	Water-Only Power	Water-Only Percentage of Power
10-mer Linear	24995.5	24989.0	99.97%
12-mer Linear	24986.9	24979.4	99.97%
16-mer Linear	24986.4	24816.7	99.32%
10-mer Parallel	24995.7	24991.9	99.98%
12-mer Parallel	24986.9	24990.8	**100.02%**
16-mer Parallel	24990.8	24986.9	99.98%

From this, it became clear that another approach would be required to unravel this mystery as traditional Fourier-based filtering techniques would not provide sufficient filtering to separate the noise from the data. The vibrational properties of water are, as noted by Lock and Bakker (2002), somewhat paradoxical, for water's vibrational period increases with temperature, taking longer to stop vibrating than when it is cooler. This is the opposite behavior compared to most molecules. Because the vibrational spectra of water is not stable with temperature changes, there remains the possibility that it is also changed by other factors such as ionization and can therefore be considered to be unstable, non-stationary, or both. Because the ultimate goal is to detect defective molecules within a group of other molecules, further study is warranted.

A New Alternative: Wavelet Analysis

During the course of this research, additional study into possible alternative methods for the analysis of these data was undertaken. One promising alternative is the application of wavelet analysis to the problem. Unlike a Fourier transform, which breaks down a periodic signal into its individual frequencies using only sines and cosines, a wavelet transform works by decomposing that signal into individual small waves (hence, "wavelet") that are employed to derived from the full length of the input signal, nor is it required that they be a regular wave (such as a sine or cosine); they can be of any arbitrary shape. These shapes can be anything from

a square wave (known as the "Harr wavelet"), to a self-similar diminishing wave ("Daubechies wavelet"). This property portends three important advantages of wavelet analysis: short acting signals can be easily identified, signals that are self-similar (such as fractal signals) stand out, and signals that are below the power spectrum threshold of a Fourier analysis can be seen more easily. There is also a disadvantage: current research into wavelets has not advanced the study far enough along to have strongly statistically validated methods. Despite that limitation, they can be extremely useful for analyzing this class of problem. (Newland, 2012).

The graphical output of a wavelet is a scaleogram—the wavelet equivalent of a spectrogram. This graph is a 3-axis graph of a 2-dimensional input signal (generally amplitude vs. time) that has been decomposed into wavelets. The x-axis is time, the y-axis is scale (that is, how much of the input signal is being considered—the larger the scale, the larger the proportion of the signal is being considered for analysis), and the z-axis is the value of the resulting wavelet coefficient (or how much of the input signal that particular wavelet takes up). In the examples below, we see four wavelet scaleograms for the linear and parallel case, and we will discuss some of their features.

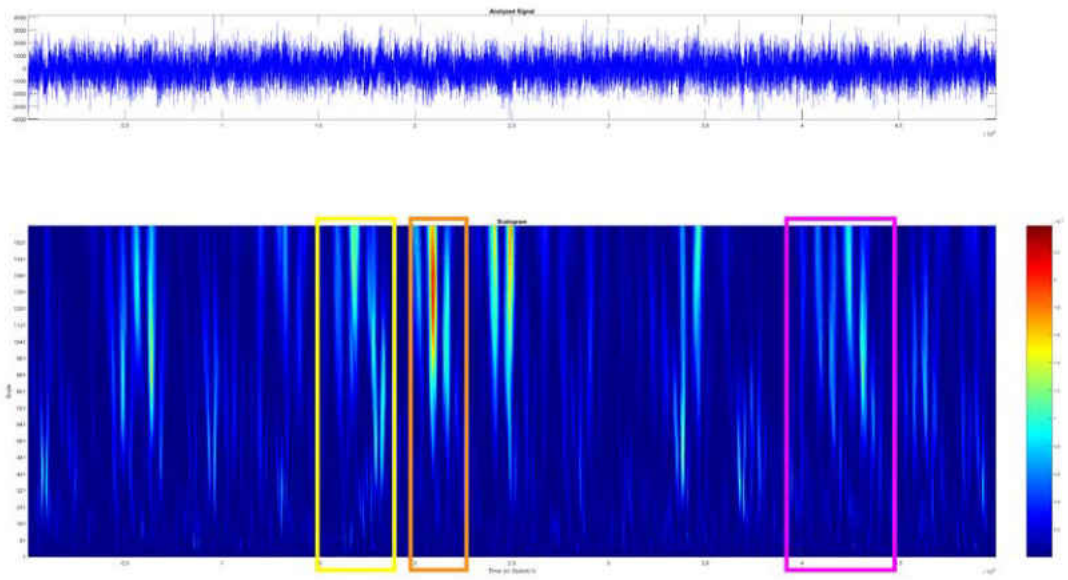


Figure 29: Linear System Configuration Scaleogram, Run 1

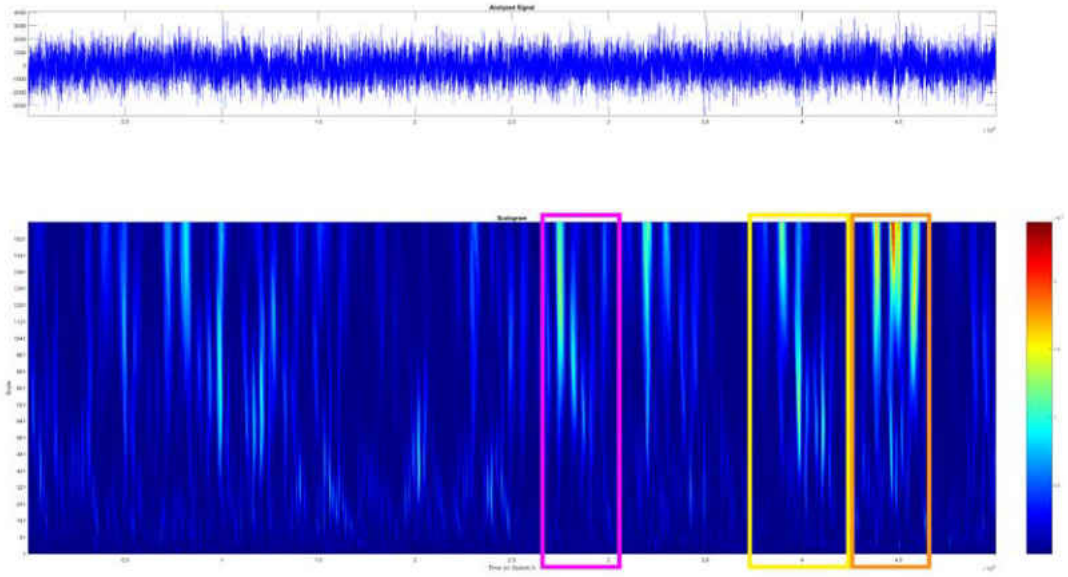


Figure 30: Linear System Configuration Scaleogram, Run 2

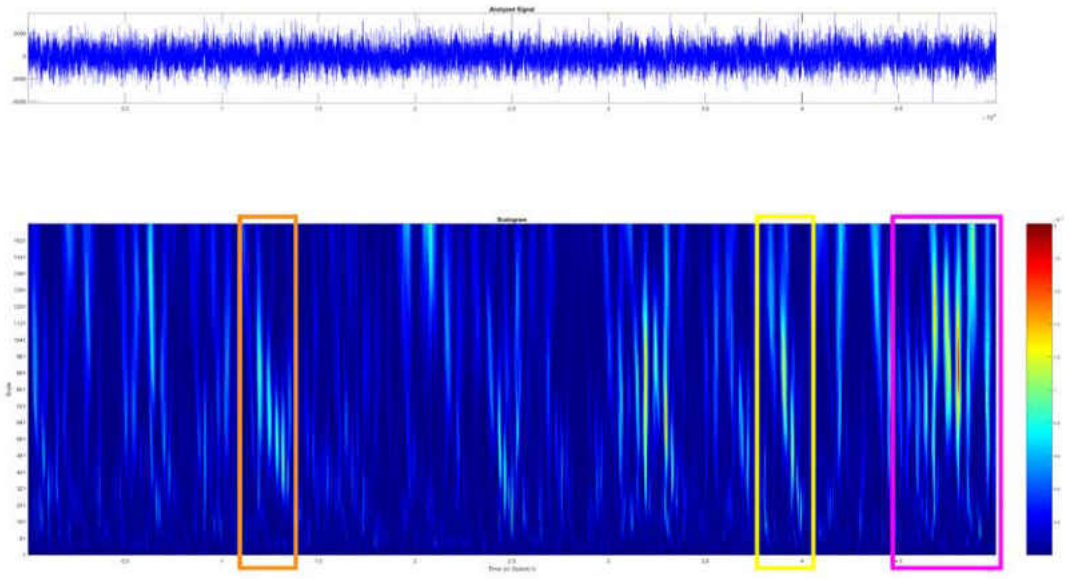


Figure 31: Parallel System Configuration Scaleogram, Run 1

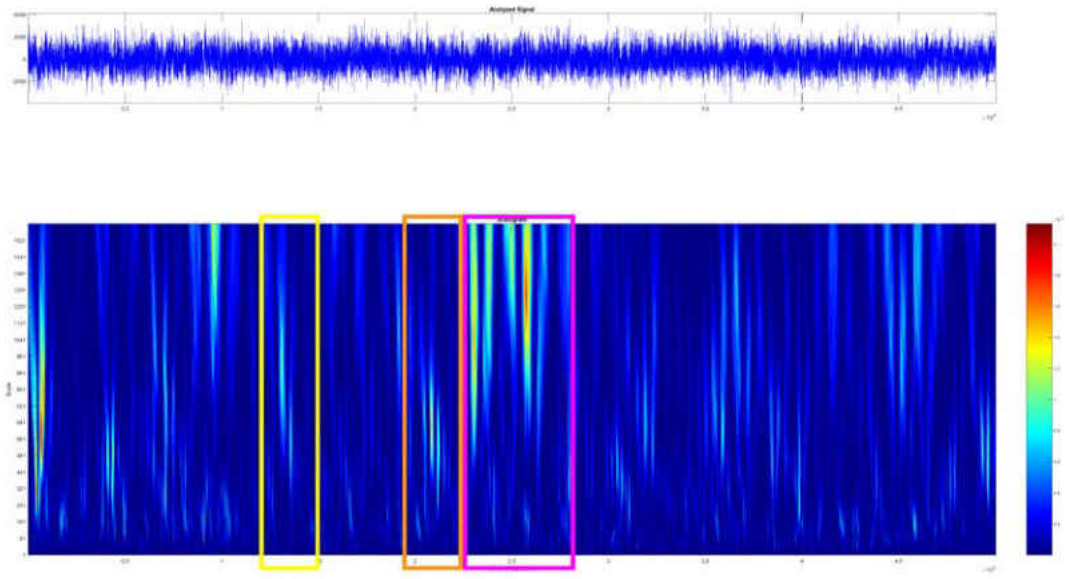


Figure 32: Parallel System Configuration Scaleogram, Run 2

These figures have been annotated to show a few common features between the scaleograms. In these figures, it is clear that similarities exist between the runs of each system, but that those

similarities are irregular, and, importantly, exist in different points in time for each run. This is possibly due to the simulation run length being too short, or it could be due to the systems being non-stationary. Stationarity is the property of a system being unchanging over time with respect to its probability distribution function, and therefore its mean and variance. Fourier analysis may not provide complete information in the case of non-stationary systems, unless it is possible to capture one entire cycle of the system. However, because we do not know when this system cycles, or whether it cycles or if the drift of its variance and mean are stochastic, Fourier transforms may not provide as much information about the system as an appropriately fitted wavelet transform could do.

CHAPTER 5: CONCLUSION

Chapter 5 Abstract

Chapter 5 begins with a summary of the introductory material relating to DNA research and the use of simulation as an emerging method of researching biological problems—the field of computational biology. The conclusions are summarized along with a brief discussion of experimental limitations and lessons learned. Finally, some parting thoughts are presented on the future direction research in this area could take in order to answer some of the questions raised in this research.

Summary

We began with a brief introduction to the structure and function of the DNA molecule and the history of DNA research. This history revealed the depth and breadth of the current understanding of the molecular processes while also showing the current lack of understanding in several key areas: simulation of DNA processes, DNA's resonant properties, and how those resonant properties may or may not react to external stimuli from electromagnetic radiation. The question of vulnerability of DNA to electromagnetic energy was reviewed. This discussion went through some of the enabling works and the history of molecular dynamics simulation, as well as introducing the concept of computational biology and some of its ultimate goals: understanding interactions between biological system components, understanding microscopic phenomena, and ultimately being able to synthesize biology and anomalies in biology from computational models. Following that, a discussion of the physics behind DNA dynamics was presented in order to help bring additional context to this research.

A review of the current molecular simulation literature revealed gaps in understanding regarding some of the properties of the vibrational spectrum of DNA. There is limited

understanding of the mechanical forces created by DNA's natural vibrations, the properties of those vibrations, and whether the resonant vibrations of the individual nucleobases noted by spectroscopy translate to any meaningful pattern when the nucleobases are assembled into DNA strands. It is believed that simulation could help to answer some of these questions due to the ability of simulation software to capture the pressure data occurring inside the molecular system in a way currently unavailable *in vitro*.

A tool review was undertaken to better understand what software tooling was available to address the research questions being formulated as well as to provide a survey of the state-of-the-art and identify any gaps that might remain in the various tools' ability to provide data for this research. This review highlighted the difference between the molecular dynamics tools (notionally similar to the "front end" in a multi-tier computing system), and the underlying force fields (the "back end"). We considered three major simulation models: CHARMM, GROMOS, and Amber. CHARMM was selected based on its current support for nucleic acids, its ability to output pressure profiles on an arbitrary 3D volume, and its ability to parallelize on dissimilar hardware.

A set of four hypotheses focusing on DNA's vibrational characteristics were synthesized. These hypotheses were formulated keeping in mind experimental and data limitations, to avoid asking questions that could not readily be answered by the data likely to be generated. There were two experimental investigations each with one hypothesis: one investigation into length-dependent vibrational changes and one into sequence-dependent vibrational changes. An additional two hypotheses involving unique variances and mathematical relationships between vibrational characteristics would use the data from the two experiments for meta-analyses.

Conclusions

This research completed a four-part investigation into various aspects of DNA vibrational properties. The first experiment investigated $\mathbf{H}_{(\text{Length NULL})}$, the hypothesis that the length of the DNA strand influenced the significant frequencies of its vibrational spectra. A surprising outcome from this experiment was that the variation between each of the 20 identical-system runs was as significant as the variation occurring between each of the different lengths. Due to that result, some post-hoc tests to assess randomness were undertaken, which resulted in a failure to reject the emergent null hypothesis that the runs were random. Due to these results, $\mathbf{H}_{(\text{Length NULL})}$ failed to be rejected. The second experiment tested $\mathbf{H}_{(\text{Sequence NULL})}$, a hypothesis that asserted a relationship between sequence and the vibrational spectra. After a successful initial chi-square test to check for departure from randomness, testing using an overlay method with two-dimensional confidence intervals was used to determine which points were unlikely to be random. There was an extremely small number of points, too small to fit a model to, and so the planned F-test was abandoned. Due to the paucity of significant points, and an inability to test the resulting model, $\mathbf{H}_{(\text{Sequence NULL})}$ can only be regarded as inconclusive.

The meta-analyses, experiments 3 and 4 proceeded along slightly divergent paths to investigate uniqueness in the strand frequencies, and whether a Fourier model could be fitted, respectively. For experiment 3, the hypothesis $\mathbf{H}_{(\text{Unique NULL})}$ investigated whether we could discern uniqueness in the vibrational spectrum between runs using cross-power spectral density (CPSD) analysis. CPSD charts were generated to compare each system case. While a case could be made that we can determine the difference between linear and parallel systems based on identification of features in their respective CPSD charts, there were not any significant features in the comparison between systems, save to a few small oscillations that were so much lower in

magnitude than the noise of the system that it is impossible to say whether they are signal or noise. We therefore regarded $\mathbf{H}_{(\text{Unique NULL})}$ as inconclusive. Finally, the fourth experiment's investigation into $\mathbf{H}_{(\text{Relation NULL})}$, the hypothesis that there is a mathematical relationship between the various system configurations, used Fourier sequence fitting to test the hypothesis. An 8-term Fourier model was fit to six paired cases: one of the pair being the original sequence, the other being the doubled sequence. It was not possible to consider the results significant because the overwhelming majority of the coefficients generated were not themselves statistically significant. The only available conclusion was to fail to reject $\mathbf{H}_{(\text{Relation NULL})}$. As a post-hoc test, periodograms with confidence intervals were employed, and found similar results.

The emergent hypothesis of the water in the water box as the source of noise in the system was investigated with positive results. It was shown that the water box itself is the source of the majority of the signal of the system, which, due to its amplitude, rendered standard filtering and noise fingerprinting techniques unable to separate the water box signal from the water box + DNA signal. Further, it was shown that any signal signature from the DNA does not influence the pressures of the water box system sufficiently to be distinguished from the underlying water box noise. This inability to remove the noise, along with the nature of the noise, places the experimental results in clearer focus while pointing the way towards future research in this area.

The use of wavelet analysis as an alternative method to investigate these vibrational properties proved interesting. Although only a very preliminary look at the use of wavelets for this type of analysis, the visual output easily identified several periodicities that eluded prior analyses. The movement of those periodicities along the time axis reflect a process that may be non-stationary, or with longer, intermittent, periodicities.

In summary, due to the amount of noise, and the nature of the process underlying these vibrational signatures, this research did not yield any rejected null hypotheses, however it did yield quite a bit more information about the nature of these processes and what problems future research must overcome to continue study in this area. In that light, it can be regarded as successful.

Experimental Limitations

There were several limitations to this research, and a brief discussion of them follows in order to help the reader understand some of the design decisions that were taken. By its nature, molecular dynamics simulation is an incredibly time-consuming process. Although computational time factor scales linearly as more CPUs are added to the cluster of computers working on the problem, the computational time factor is non-polynomial for the complexity/size of the molecular system. Therefore, doubling the size of the molecular system being analyzed roughly quadruples the time required to complete the computations. For that reason, simulation runs were limited to 500,000 two-femtosecond "frames." Simulations were also run on relatively small (30,000 to 40,000 atom) systems in order to keep computation time to a reasonable length. The issue of time limitation was further felt because it was desired to run multiple replications of the same system in order to get a broad sampling, and because it was necessary to run each system simulation twice—once in online mode, and once in offline mode to generate the PME pressure terms. Finally, there were analysis limitations. The tools for multivariate Fourier analysis are not well suited to studies with a very large number of frequency parameters, such as this one, so more traditional and therefore likely less-powerful tools, derived from first-principles, had to be used. Furthermore, while automated frequency analysis tools are certainly available, they lack statistical rigor at this time to be used with an investigation such as

this one, and they have not been designed to be applied to this class of problem; this would be an excellent area for the development of new statistical analysis tools and formulas.

Lessons

Although Fourier analysis of these pressure waves proved to be less informative than hoped and at best resulting in rejecting two hypotheses and being inconclusive on two others, it is worth noting some important conclusions that can be derived from these Fourier analysis outcomes. As we saw in the results, the total power of the statistically significant portions of this system was less than half the overall power represented in the system. This result indicates two likely facts: the process that causes DNA harmonic resonance (and therefore is partially responsible for closure) is seemingly a low-intensity process, and any effect is likely to occur in the non-significant range. For these reasons, Fourier analysis is not the appropriate method to further these research questions.

One clear indication from all of the data stood out: that the effects are highest at lower frequencies than anticipated, and that at frequencies near the gigahertz and millimeter-wave ranges. This has potential industrial and commercial health and safety implications: these frequencies are common in our daily lives (e.g., from 2.4 and 5.8GHz WiFi, airport radars that function between 8 and 18GHz, new '5G' communications applications using the 73GHz band, etc.), and because the solvated DNA molecules showed resonances in those ranges, care should be taken to ensure that newer, more powerful communications devices using these bands are tested for safety around mammals. Further study, perhaps with external stimulation in those ranges would be instructive in probing the industrial hygiene implications of higher-energy RF emissions in the upper gigahertz range.

Parting Thoughts and Future Research

This research helped to answer some of the questions raised in Calloway (2011), but it is by no means definitive. Several questions remain to be sufficiently answered, including questions about the relationship between strand content/length and any signals that they may generate. Although this research did not find significant effects, it by no means was capable of covering all of the spectrum or the additional experimental possibilities that remain. In that regard, this research may be regarded as conclusive, but only for the narrow band between 10^{10} and 10^{13} hertz, and for short periods not exceeding 1 nanosecond, and for solvated DNA. It in no way speaks for frequencies above or below that band, nor does it speak to the processes' longer term movements. Furthermore, due to the noise of the solvating water, it will be necessary to develop better filtering mechanisms for dealing with this situation: a known noise signature which is the majority of the signal output of the experiment. Because we are likely looking at longer-term and lower-intensity processes, Wavelet analysis is likely to be a promising way forward from here. There also remains the possibility that the process is non-stationary, at which point, Wavelet analysis will be required to understand what relationships exist, if any, since traditional frequency analysis cannot capture non-stationarity under these experimental conditions.

Keeping these conclusions and conjectures in mind, any future research wishing to expand on this subject should, at the very minimum, be capable of addressing the following questions. What are the frequency impacts of these processes below 10^{10} Hz and above 10^{13} Hz? What is the nature of the lower-power frequency components, and how do they interact? What is the nature of these processes from the perspective of stationarity? Would Wavelet analysis be a more appropriate method of analyzing both the lower-power components and assessing

stationarity, or would more traditional tests such as Dickey-Fuller be appropriate? Regardless, there remains much to be discovered in this field, and there are rich opportunities for future study.

APPENDIX A: EXPERIMENT 2 COMPARISON OUTPUT TABLES

Table 6: Linear Significant Frequency Points

Sequence	Frequency Index	Frequency (Hz)
13	3	2E+09
16	6	5E+09
12	8	7E+09
15	11	1E+10
16	11	1E+10
14	12	1.1E+10
8	14	1.3E+10
3	26	2.5E+10
10	28	2.7E+10
16	32	3.1E+10
3	40	3.9E+10
13	43	4.2E+10
10	48	4.7E+10
19	53	5.2E+10
5	56	5.5E+10
10	57	5.6E+10
18	73	7.2E+10
12	90	8.9E+10
1	97	9.6E+10
13	99	9.8E+10
18	106	1.05E+11
5	153	1.52E+11
17	200	1.99E+11
16	224	2.23E+11
10	289	2.88E+11
17	303	3.02E+11
17	412	4.11E+11
10	505	5.04E+11
19	540	5.39E+11
10	566	5.65E+11
9	652	6.51E+11
9	658	6.57E+11
11	802	8.01E+11
7	871	8.7E+11
14	1062	1.06E+12
17	1196	1.2E+12
4	1247	1.25E+12
6	1282	1.28E+12
17	1384	1.38E+12
7	1479	1.48E+12
16	1550	1.55E+12

18	1687	1.69E+12
4	1718	1.72E+12
17	1833	1.83E+12
1	2306	2.31E+12
10	2664	2.66E+12
14	2721	2.72E+12
19	2823	2.82E+12
3	2846	2.85E+12
10	3090	3.09E+12
11	4524	4.52E+12

Table 7: Parallel Significant Frequency Points

Sequence	Frequency Index	Frequency (Hz)
1	2	1000040002
14	3	2000080003
8	5	4000160006
10	5	4000160006
10	18	17000680027
19	20	19000760030
13	23	22000880035
14	28	27001080043
1	29	28001120045
4	33	32001280051
10	44	43001720069
13	49	48001920077
7	53	52002080083
1	54	53002120085
19	63	62002480099
20	68	67002680107
1	70	69002760110
6	70	69002760110
2	74	73002920117
1	75	74002960118
9	86	85003400136
19	98	97003880155
10	103	1.02004E+11
18	110	1.09004E+11
12	115	1.14005E+11
20	119	1.18005E+11
2	129	1.28005E+11
5	145	1.44006E+11
6	151	1.50006E+11

9	153	1.52006E+11
14	159	1.58006E+11
14	172	1.71007E+11
12	190	1.89008E+11
3	200	1.99008E+11
7	203	2.02008E+11
10	217	2.16009E+11
12	231	2.30009E+11
7	267	2.66011E+11
10	269	2.68011E+11
3	277	2.76011E+11
5	284	2.83011E+11
12	298	2.97012E+11
4	304	3.03012E+11
14	305	3.04012E+11
17	305	3.04012E+11
10	309	3.08012E+11
18	326	3.25013E+11
5	339	3.38014E+11
7	354	3.53014E+11
20	363	3.62014E+11
4	365	3.64015E+11
11	366	3.65015E+11
11	411	4.10016E+11
3	412	4.11016E+11
2	506	5.0502E+11
8	580	5.79023E+11
6	583	5.82023E+11
10	607	6.06024E+11
3	623	6.22025E+11
2	724	7.23029E+11
9	731	7.30029E+11
19	734	7.33029E+11
11	933	9.32037E+11
12	946	9.45038E+11
10	1049	1.04804E+12
11	2906	2.90512E+12
5	3367	3.36613E+12
13	4133	4.13217E+12
14	5005	5.0042E+12
14	5979	5.97824E+12

Table 8: LMS Filtered Linear Significant Frequency Points

Sequence	Frequency Index	Frequency (Hz)
7	1	0
2	2	1E+09
13	2	1E+09
17	3	2E+09
13	4	3E+09
13	5	4E+09
2	6	5E+09
15	6	5E+09
15	9	8E+09
13	11	1E+10
4	12	1.1E+10
10	12	1.1E+10
13	12	1.1E+10
13	13	1.2E+10
13	15	1.4E+10
10	16	1.5E+10
13	16	1.5E+10
15	18	1.7E+10
10	23	2.2E+10
13	25	2.4E+10
1	26	2.5E+10
17	27	2.6E+10
4	31	3E+10
15	36	3.5E+10
2	41	4E+10
15	44	4.3E+10
10	69	6.8E+10
6	71	7E+10
13	76	7.5E+10
3	89	8.8E+10
15	99	9.8E+10
16	119	1.18E+11
15	135	1.34E+11
6	161	1.6E+11
3	200	1.99E+11
16	203	2.02E+11
7	229	2.28E+11
2	238	2.37E+11
15	241	2.4E+11
11	363	3.62E+11
15	449	4.48E+11

17	519	5.18E+11
3	553	5.52E+11
2	586	5.85E+11
9	723	7.22E+11
3	874	8.73E+11
7	914	9.13E+11
19	982	9.81E+11
9	1150	1.15E+12
10	1155	1.15E+12
10	1315	1.31E+12
15	1441	1.44E+12
17	1556	1.56E+12
12	1606	1.61E+12
19	2178	2.18E+12
20	2332	2.33E+12
7	2589	2.59E+12
16	2857	2.86E+12
14	3547	3.55E+12
16	3582	3.58E+12
16	3800	3.8E+12
14	4841	4.84E+12
5	5830	5.83E+12
5	5991	5.99E+12

Table 9: Parallel LMS Filtered Significant Frequency Points

Sequence	Frequency Index	Frequency (Hz)
11	1	0
1	2	1000040002
14	3	2000080003
8	5	4000160006
10	5	4000160006
10	18	17000680027
19	20	19000760030
13	23	22000880035
14	28	27001080043
1	29	28001120045
4	33	32001280051
10	44	43001720069
13	49	48001920077
7	53	52002080083
1	54	53002120085
19	63	62002480099

20	68	67002680107
1	70	69002760110
6	70	69002760110
2	74	73002920117
1	75	74002960118
9	86	85003400136
19	98	97003880155
10	103	1.02004E+11
18	110	1.09004E+11
12	115	1.14005E+11
20	119	1.18005E+11
2	129	1.28005E+11
5	145	1.44006E+11
6	151	1.50006E+11
9	153	1.52006E+11
14	159	1.58006E+11
14	172	1.71007E+11
12	190	1.89008E+11
3	200	1.99008E+11
7	203	2.02008E+11
10	217	2.16009E+11
12	231	2.30009E+11
7	267	2.66011E+11
10	269	2.68011E+11
3	277	2.76011E+11
5	284	2.83011E+11
12	298	2.97012E+11
4	304	3.03012E+11
14	305	3.04012E+11
17	305	3.04012E+11
10	309	3.08012E+11
18	326	3.25013E+11
5	339	3.38014E+11
7	354	3.53014E+11
20	363	3.62014E+11
4	365	3.64015E+11
11	366	3.65015E+11
11	411	4.10016E+11
3	412	4.11016E+11
2	506	5.0502E+11
8	580	5.79023E+11
6	583	5.82023E+11
10	607	6.06024E+11

3	623	6.22025E+11
2	724	7.23029E+11
9	731	7.30029E+11
19	734	7.33029E+11
11	933	9.32037E+11
12	946	9.45038E+11
10	1049	1.04804E+12
11	2906	2.90512E+12
5	3367	3.36613E+12
13	4133	4.13217E+12
14	5005	5.0042E+12
14	5979	5.97824E+12

APPENDIX B: EXPERIMENT 4 FOURIER FITTING EQUATIONS

10-mer Linear Run 1 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -46.08  (-53.83, -38.34)
  a1 =       4.601  (-6.39, 15.59)
  b1 =      -20.39  (-31.33, -9.449)
  a2 =      -30.89  (-41.86, -19.92)
  b2 =       -14.2  (-25.36, -3.048)
  a3 =       8.822  (-2.205, 19.85)
  b3 =        9.48  (-1.506, 20.47)
  a4 =       48.14  (36.94, 59.35)
  b4 =       16.71  (4.052, 29.37)
  a5 =       63.16  (52.15, 74.17)
  b5 =       7.184  (-7.717, 22.09)
  a6 =      -33.62  (-44.99, -22.24)
  b6 =      -15.38  (-27.86, -2.9)
  a7 =       38.32  (23.68, 52.96)
  b7 =      -43.33  (-57.05, -29.6)
  a8 =       -13.8  (-24.8, -2.807)
  b8 =      -2.332  (-13.83, 9.167)
  w =   0.0001167  (0.0001165, 0.0001168)
```

10-mer Linear Run 2 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -32.83  (-40.68, -24.97)
  a1 =      -6.776  (-17.86, 4.31)
  b1 =       5.214  (-5.92, 16.35)
  a2 =        42.4  (31.22, 53.59)
  b2 =      -22.02  (-33.52, -10.52)
  a3 =       52.44  (40.97, 63.9)
  b3 =       26.46  (14.1, 38.83)
  a4 =       36.19  (24.22, 48.15)
  b4 =        31.7  (19.5, 43.89)
  a5 =      -10.93  (-23.61, 1.752)
  b5 =       34.68  (23.44, 45.93)
  a6 =       6.202  (-5.012, 17.42)
  b6 =      -7.444  (-18.63, 3.744)
  a7 =       53.46  (39.99, 66.93)
  b7 =      -31.32  (-48.49, -14.15)
  a8 =       4.918  (-6.336, 16.17)
  b8 =      -6.636  (-17.82, 4.552)
  w =   0.0001958  (0.0001957, 0.000196)
```

12-mer Linear Run 1 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =         -23  (-30.74, -15.27)
  a1 =         2.5  (-8.612, 13.61)
  b1 =        52.5  (41.42, 63.58)
  a2 =       44.88  (33.83, 55.92)
  b2 =       27.78  (15.52, 40.05)
  a3 =       15.13  (1.886, 28.38)
  b3 =      -31.22  (-43.23, -19.21)
  a4 =      -2.385  (-16.4, 11.63)
  b4 =      -31.92  (-42.83, -21.01)
  a5 =      -7.702  (-20.35, 4.95)
  b5 =      -14.13  (-26.82, -1.45)
  a6 =      -47.58  (-59.96, -35.2)
  b6 =       27.19  (6.987, 47.4)
  a7 =      -31.32  (-43.22, -19.42)
  b7 =       4.417  (-12.4, 21.24)
  a8 =      -6.014  (-19.53, 7.505)
  b8 =       15.16  (3.649, 26.68)
  w =   1.502e-05  (1.482e-05, 1.522e-05)
```

12-mer Linear Run 2 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =  -5.421e+06  (-3.713e+07, 2.629e+07)
  a1 =   2.136e+06  (-1.549e+07, 1.976e+07)
  b1 =   9.766e+06  (-4.608e+07, 6.562e+07)
  a2 =   7.098e+06  (-3.058e+07, 4.478e+07)
  b2 =   -3.27e+06  (-2.966e+07, 2.312e+07)
  a3 =  -3.111e+06  (-2.725e+07, 2.103e+07)
  b3 =  -4.089e+06  (-2.265e+07, 1.447e+07)
  a4 =  -1.799e+06  (-7.614e+06, 4.017e+06)
  b4 =    2.14e+06  (-1.35e+07, 1.778e+07)
  a5 =    1.09e+06  (-6.207e+06, 8.386e+06)
  b5 =   5.594e+05  (-3.407e+05, 1.46e+06)
  a6 =   1.002e+05  (-6.718e+05, 8.722e+05)
  b6 =  -3.986e+05  (-2.744e+06, 1.947e+06)
  a7 =   -9.54e+04  (-5.551e+05, 3.643e+05)
  b7 =       -1440  (-3.384e+05, 3.355e+05)
  a8 =        2620  (-5.426e+04, 5.95e+04)
  b8 =   1.139e+04  (-2.721e+04, 4.999e+04)
  w =   5.843e-06  (3.859e-06, 7.827e-06)
```

16-mer Linear Run 1 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =  -4.929e+05  (-2.112e+06, 1.127e+06)
  a1 =  -1.862e+05  (-4.962e+05, 1.237e+05)
  b1 =   8.909e+05  (-2.077e+06, 3.859e+06)
  a2 =   6.534e+05  (-1.621e+06, 2.928e+06)
  b2 =   2.886e+05  (-2.01e+05, 7.782e+05)
  a3 =   2.799e+05  (-2.114e+05, 7.712e+05)
  b3 =    -3.8e+05  (-1.818e+06, 1.058e+06)
  a4 =  -1.663e+05  (-8.951e+05, 5.626e+05)
  b4 =  -1.971e+05  (-5.626e+05, 1.685e+05)
  a5 =  -1.025e+05  (-3.097e+05, 1.047e+05)
  b5 =   4.781e+04  (-2.328e+05, 3.285e+05)
  a6 =        4486  (-6.814e+04, 7.712e+04)
  b6 =    3.76e+04  (-5.005e+04, 1.253e+05)
  a7 =        8496  (-1.757e+04, 3.457e+04)
  b7 =        2723  (-5838, 1.128e+04)
  a8 =        1043  (-188.1, 2275)
  b8 =      -803.9  (-5382, 3775)
  w =   6.825e-06  (5.06e-06, 8.59e-06)
```

16-mer Linear Run 2 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      0.5306  (-6.446, 7.507)
  a1 =        58.5  (48.74, 68.25)
  b1 =      -82.85  (-92.45, -73.25)
  a2 =     -0.5812  (-10.72, 9.555)
  b2 =       22.08  (12.2, 31.96)
  a3 =      0.4687  (-10.78, 11.71)
  b3 =       46.78  (37.15, 56.41)
  a4 =      -7.139  (-17.03, 2.751)
  b4 =      -12.39  (-22.29, -2.49)
  a5 =       11.28  (-3.628, 26.19)
  b5 =       59.39  (49.75, 69.03)
  a6 =       14.37  (4.262, 24.49)
  b6 =       -7.57  (-17.47, 2.324)
  a7 =       23.48  (9.502, 37.46)
  b7 =       30.82  (20.63, 41.01)
  a8 =       76.89  (65.76, 88.02)
  b8 =       33.51  (12.55, 54.48)
  w =    1.24e-05  (1.23e-05, 1.25e-05)
```

10-mer Parallel Run 1 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -46.08  (-53.83, -38.34)
  a1 =       4.601  (-6.39, 15.59)
  b1 =      -20.39  (-31.33, -9.449)
  a2 =      -30.89  (-41.86, -19.92)
  b2 =       -14.2  (-25.36, -3.048)
  a3 =       8.822  (-2.205, 19.85)
  b3 =        9.48  (-1.506, 20.47)
  a4 =       48.14  (36.94, 59.35)
  b4 =       16.71  (4.052, 29.37)
  a5 =       63.16  (52.15, 74.17)
  b5 =       7.184  (-7.717, 22.09)
  a6 =      -33.62  (-44.99, -22.24)
  b6 =      -15.38  (-27.86, -2.9)
  a7 =       38.32  (23.68, 52.96)
  b7 =      -43.33  (-57.05, -29.6)
  a8 =       -13.8  (-24.8, -2.807)
  b8 =      -2.332  (-13.83, 9.167)
  w =   0.0001167  (0.0001165, 0.0001168)
```

10-mer Parallel Run 2 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -32.83  (-40.68, -24.97)
  a1 =      -6.776  (-17.86, 4.31)
  b1 =       5.214  (-5.92, 16.35)
  a2 =        42.4  (31.22, 53.59)
  b2 =      -22.02  (-33.52, -10.52)
  a3 =       52.44  (40.97, 63.9)
  b3 =       26.46  (14.1, 38.83)
  a4 =       36.19  (24.22, 48.15)
  b4 =        31.7  (19.5, 43.89)
  a5 =      -10.93  (-23.61, 1.752)
  b5 =       34.68  (23.44, 45.93)
  a6 =       6.202  (-5.012, 17.42)
  b6 =      -7.444  (-18.63, 3.744)
  a7 =       53.46  (39.99, 66.93)
  b7 =      -31.32  (-48.49, -14.15)
  a8 =       4.918  (-6.336, 16.17)
  b8 =      -6.636  (-17.82, 4.552)
  w =   0.0001958  (0.0001957, 0.000196)
```

12-mer Parallel Run 1 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -70.42  (-77.82, -63.02)
  a1 =       19.59  (9.253, 29.92)
  b1 =      -12.77  (-23.36, -2.174)
  a2 =       25.86  (15.31, 36.4)
  b2 =       26.45  (15.78, 37.11)
  a3 =       55.49  (44.62, 66.37)
  b3 =       35.49  (23.82, 47.16)
  a4 =       9.457  (-1.04, 19.95)
  b4 =      -11.28  (-21.83, -0.7287)
  a5 =      -8.028  (-18.5, 2.442)
  b5 =      -0.902  (-11.34, 9.534)
  a6 =       31.61  (18.54, 44.68)
  b6 =      -43.23  (-54.78, -31.68)
  a7 =       44.49  (29.98, 59)
  b7 =      -50.37  (-64.23, -36.51)
  a8 =      -40.43  (-50.82, -30.04)
  b8 =       5.099  (-9.341, 19.54)
  w =   3.216e-05  (3.204e-05, 3.228e-05)
```

12-mer Parallel Run 2 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -88.34  (-95.65, -81.03)
  a1 =       45.06  (34.58, 55.53)
  b1 =      -43.51  (-53.87, -33.16)
  a2 =       11.68  (1.193, 22.16)
  b2 =       23.81  (13.42, 34.21)
  a3 =       38.16  (27.82, 48.5)
  b3 =      -12.22  (-23.21, -1.232)
  a4 =      -22.27  (-32.64, -11.89)
  b4 =      -7.686  (-18.36, 2.983)
  a5 =       63.18  (52.84, 73.53)
  b5 =      -2.135  (-17.01, 12.74)
  a6 =       36.77  (26.27, 47.26)
  b6 =      -9.376  (-22.28, 3.524)
  a7 =       9.422  (-2.524, 21.37)
  b7 =      -24.24  (-34.81, -13.67)
  a8 =     -0.9446  (-15.6, 13.71)
  b8 =      -37.99  (-48.31, -27.67)
  w =   6.745e-05  (6.732e-05, 6.759e-05)
```

16-mer Parallel Run 1 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -89.91  (-96.39, -83.43)
  a1 =      -15.47  (-24.63, -6.304)
  b1 =      -9.988  (-19.18, -0.7935)
  a2 =      -21.77  (-30.96, -12.58)
  b2 =       8.023  (-1.277, 17.32)
  a3 =      -38.34  (-48.1, -28.58)
  b3 =      -29.59  (-39.76, -19.43)
  a4 =       3.292  (-6.384, 12.97)
  b4 =      -20.33  (-29.51, -11.15)
  a5 =      -29.16  (-38.39, -19.93)
  b5 =      -5.491  (-16.26, 5.274)
  a6 =       8.116  (-3.376, 19.61)
  b6 =       30.32  (20.95, 39.7)
  a7 =      -36.63  (-46.32, -26.93)
  b7 =      -11.69  (-25.1, 1.719)
  a8 =       11.46  (2.206, 20.72)
  b8 =      -4.695  (-14.52, 5.13)
  w =   0.0002475  (0.0002473, 0.0002476)
```

16-mer Parallel Run 2 Fitted Model:

```
General model Fourier8:
f8(x) =
          a0 + a1*cos(x*w) + b1*sin(x*w) +
          a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
          a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
          a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
          a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
  a0 =      -69.26  (-75.75, -62.78)
  a1 =       8.111  (-1.044, 17.27)
  b1 =      -1.878  (-11.08, 7.32)
  a2 =      -38.12  (-47.32, -28.93)
  b2 =      -2.955  (-12.4, 6.492)
  a3 =      -18.94  (-28.11, -9.779)
  b3 =       4.681  (-4.708, 14.07)
  a4 =       5.505  (-4.355, 15.37)
  b4 =       28.14  (18.95, 37.34)
  a5 =      -25.63  (-35.7, -15.55)
  b5 =      -27.69  (-37.68, -17.71)
  a6 =       8.069  (-1.113, 17.25)
  b6 =      0.7376  (-8.493, 9.968)
  a7 =       2.969  (-12.75, 18.68)
  b7 =       60.44  (51.25, 69.63)
  a8 =       12.57  (1.682, 23.46)
  b8 =        25.9  (16.15, 35.66)
  w =     7.2e-05  (7.187e-05, 7.212e-05)
```

APPENDIX C: CLUSTER SPECIFICATIONS

The SELL lab compute cluster consists of the following computers and CPUs for a total of nine computer workstations and 60 CPU cores:

2x Dell Precision T7500 with 2x Intel® Xeon™ X5650 2.66GHz 6-core CPUs

2x Dell Precision T3500 with Intel® Xeon™ W3565 3.20 GHz 4-core CPU

3x Dell OptiPlex 980 with Intel® Core™ i7-860 2.8GHz 4-core CPU

2x Dell XPS 730 with Intel® Core™ 2 Quad Q9650 3.0GHz 4-core CPU

APPENDIX D: MODEL CONSTRUCTION

In this appendix, a description of how the models were constructed is presented so that the initialization procedures can be the same for future research. In general, the process begins with the generation of a random DNA string, which is then converted into an appropriate format using free and open-source tools. From that point, the raw DNA strands are replicated, inserted into a water box, solvated and ionized.

1. To begin, a random strand of DNA was generated. For the purposes of this appendix, we will use the 10-mer sample, TACGCCCAAA. This random strand was generated using the tool by Maduro (2003), though one could just as easily set up a random number generator to generate a 10-digit number with values 1 through 4, assigning those values to A, C, G, or T. From there, the strand needs to be turned into a PDB (Protein Database) and a PSF (Protein Structure File). This can be accomplished with the Nucleic Acid Builder (NAB) in the AMBER suite of tools (example script in the files portion of this appendix), or with the web-based generator tool known as the "make-na server" from Stroud (2006). When using the make-na server, the parameters should be set as shown in the next figure.

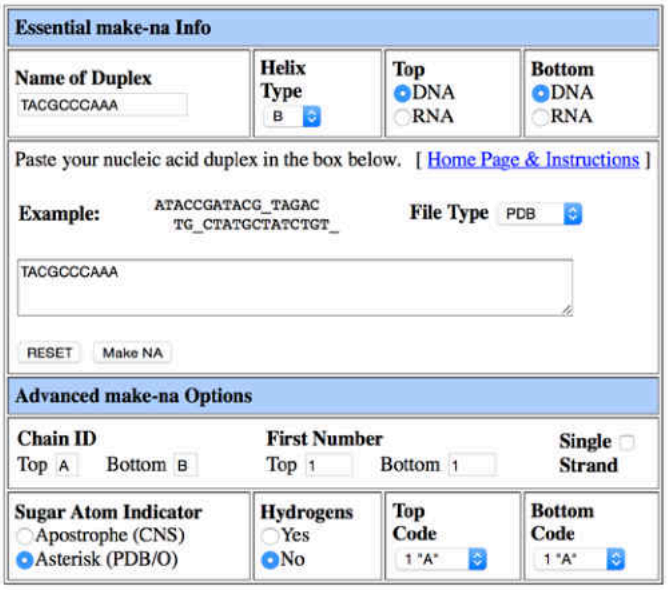


Figure 33: Options used for the Make-NA Server

2. Once the PDB is obtained, it should be opened in VMD 1.91 or later. It is first necessary to generate the PSF. AutoPSF is used by accessing the **Extensions** menu, then **Modeling**, then **Automatic PSF Builder**. First, specify the output basename (the prefix 1 and 2 in front of the file name is helpful for later step to distinguish between chains), then load the input files. Have AutoPSF guess and split chains. You may be prompted for an original PDB file, respond **No** to the prompt. You should have two segments identified. For the first chain, simply click **Create Chains** to create the PSF and matching PDB for chain 1; for the second chain, set the segment names to N3 and N4 using the **Edit Chain** button, then click **Create Chains**. Screenshots follow in the next figure to demonstrate how to setup AutoPSF.

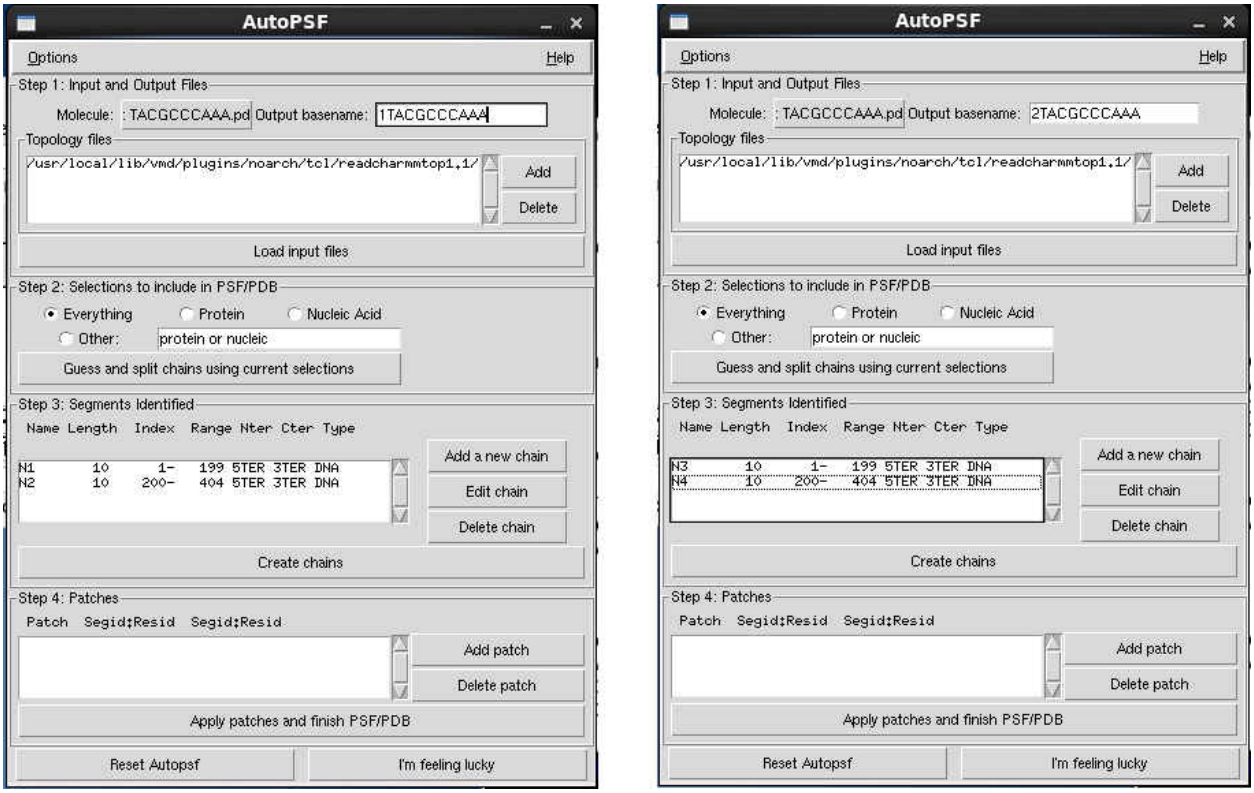


Figure 34: AutoPSF Dialog Boxes for Segments 1 and 2

3. Now that both chains have been generated, it is necessary to prepare the file for ionization and solvation. This requires two discrete steps: combining the molecules together, and rotating them.

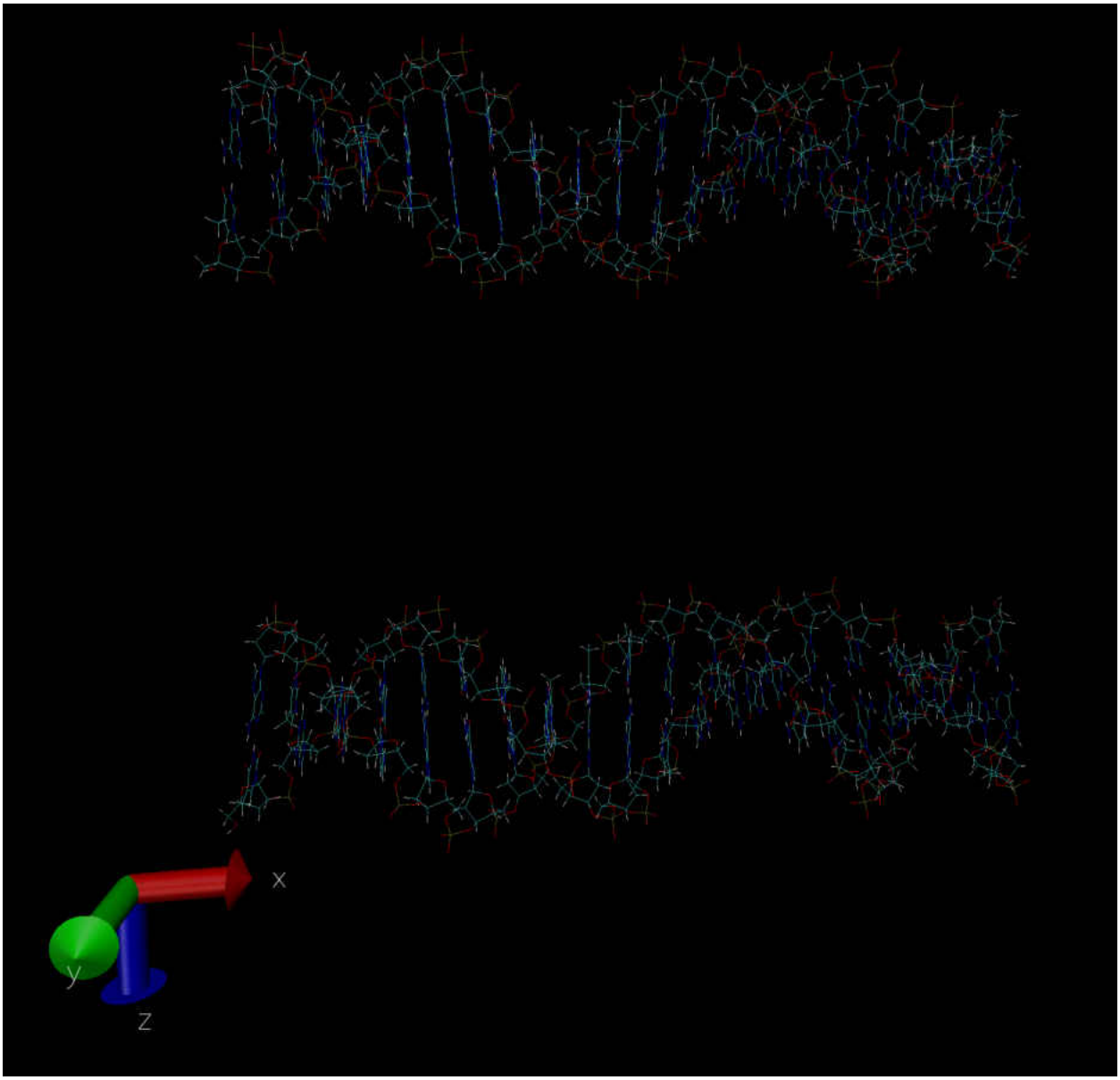
These steps are accomplished using the TCL/TK console in VMD. First, we combine the moecules (this script can be called combine_molecules.tcl):

```
set pdb1 ./1tacgcccaaa.pdb
set psf1 ./1tacgcccaaa.psf
set pdb2 ./2tacgcccaaa.pdb
set psf2 ./2tacgcccaaa.psf

set outputPdb ./tacgcccaaa_combined.pdb
set outputPsf ./tacgcccaaa_combined.psf

package require psfgen
resetpsf

readpsf  $psf1
coordpdb $pdb1
readpsf  $psf2
coordpdb $pdb2

writepdb $outputPdb
writepsf $outputPsf
```

Second, we translate one of the moelcules 90 degrees, and then move it by a distance that will yield 12Å of separation (this script can be called rotate_parallel.tcl):

```
set pdb0 ./tacgcccaaa_combined.pdb
set psf0 ./tacgcccaaa_combined.psf

mol load psf $psf0 pdb $pdb0

set outputPdb ./tacgcccaaa_Parallel.pdb
set outputPsf ./tacgcccaaa_Parallel.psf

set sel [atomselect top "segname N1 N2 N3 N4"]
set M [transvecinv {0 0 1}]
$sel move $M
set M [transaxis x -90]
$sel move $M

set sel [atomselect top "segname N3 N4"]
$sel moveby {0 0 44.73}

set all [atomselect top all]

$all writepdb $outputPdb
$all writepsf $outputPsf

mol delete all
```

4. With the molecule moved, one should have a result that resembles this (for the parallel case):



Now, the system must be solvated (placed into water) and then ionized. Solvation is accomplished with the graphical solvation tool in VMD, available from the **Extensions** menu, under **Modeling**, and **Add Solvation Box**. The system was solvated to a boundary 12Å from the water box edge. The Solvate window is setup as shown:

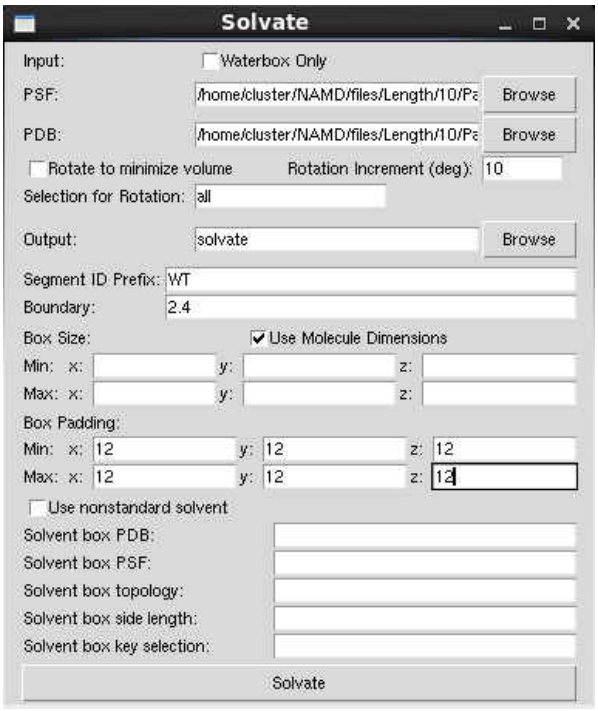


Figure 35: Solvate Options Dialog Box

The result of which will look like this (helices have been enhanced):

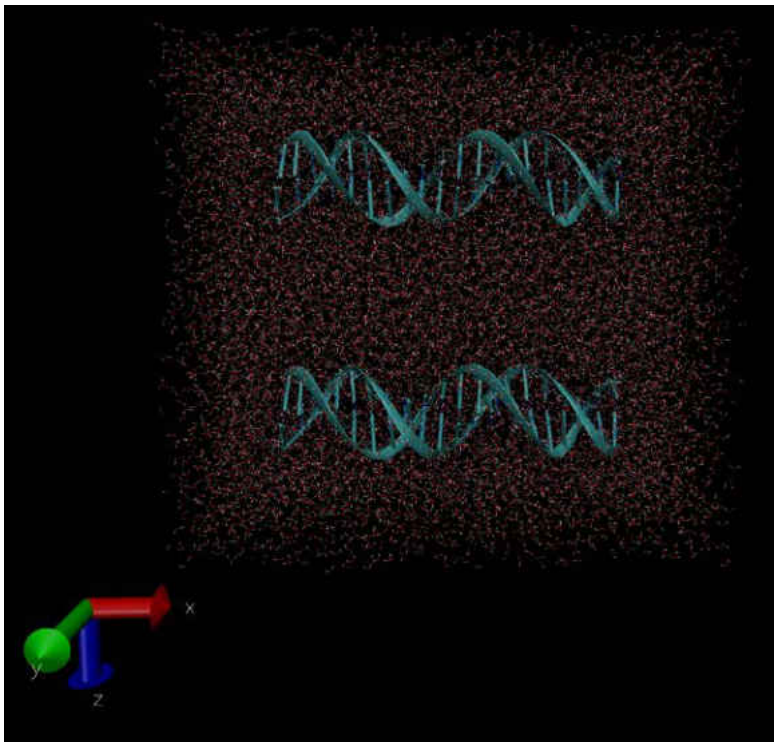


Figure 36: Solvated Parallel System

5. Finally, the system must be ionized. To do so, invoke the NAMD Autoionize tool from the **Extensions** menu, under **Modeling**, and **Add Ions**. Set the tool to neutralize charges and ionize the system to 0.5 mol/L, as shown in the next figure.

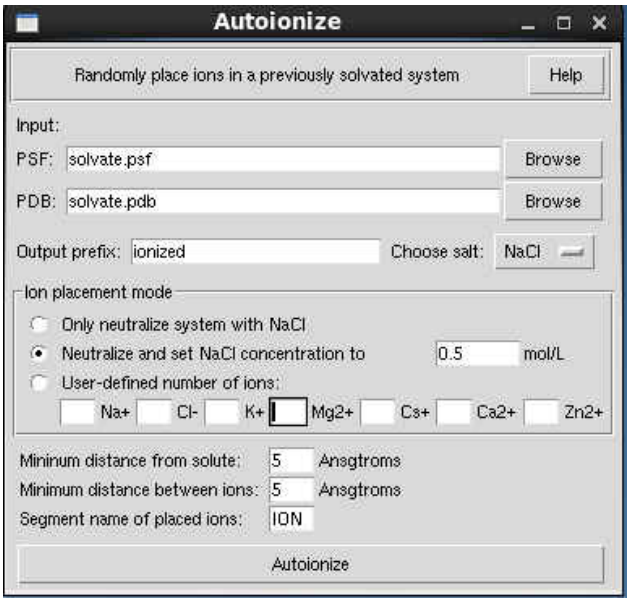


Figure 37: Autoionize Dialog Box

Once ionized, the system will be ready for input into the NAMD engine for initial startup steps, minimization, heating and equilibration, before finally being run. An ionized system will look like the following figure (helices and ions enhanced):

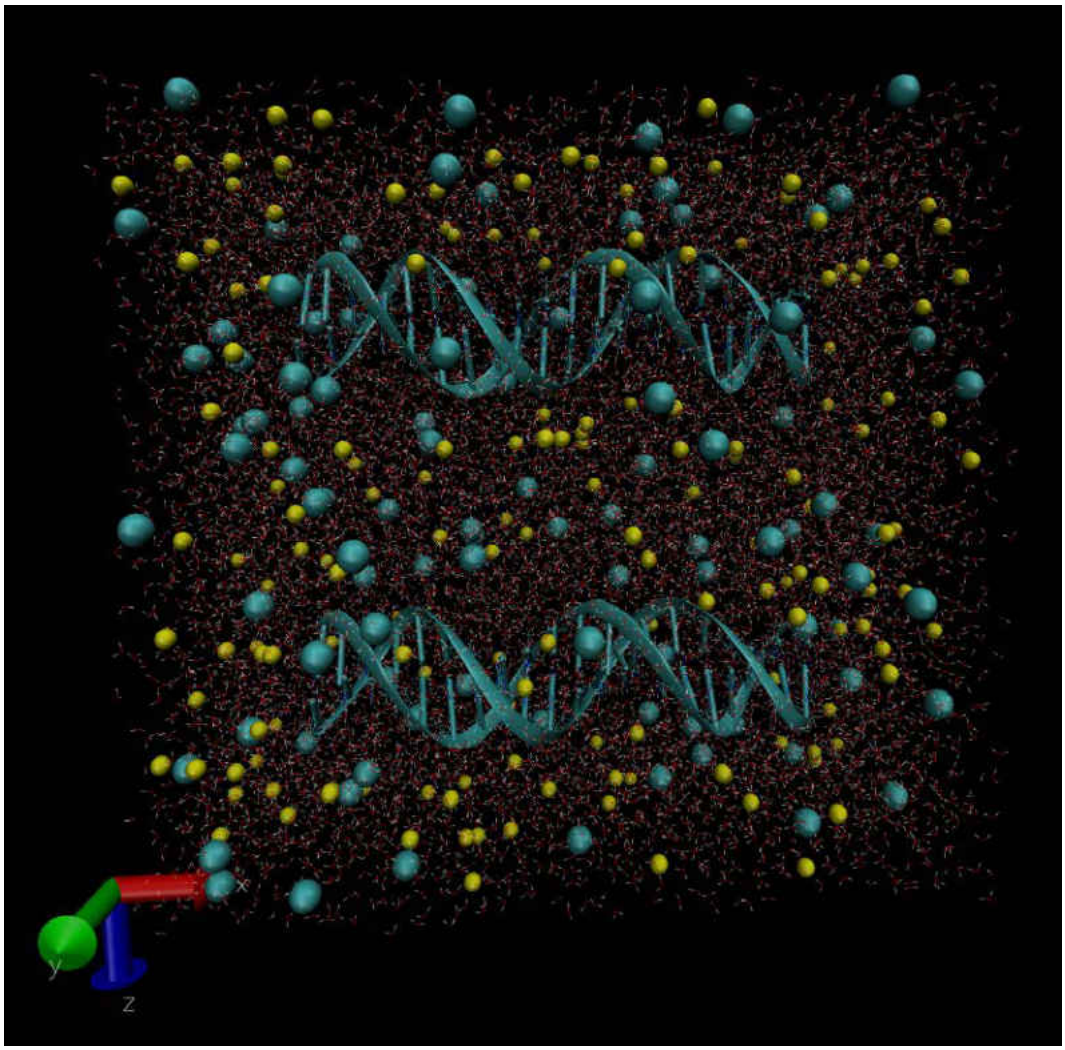


Figure 38: Solvated and Ionized System

AMBER Nucleic Acid Builder script for creating the sequences in experiment 2 (named sequences.nab):

```
// Program 1 - Average B-form DNA duplex
molecule m;

m = bdna( "gaccgaatgcggaatcatgc" );
putpdb( "1seq1_raw.pdb", m );
putpdb( "2seq1_raw.pdb", m );

m = bdna( "atcgttccactgtgtttgtc" );
putpdb( "1seq2_raw.pdb", m );
putpdb( "2seq2_raw.pdb", m );
```

```
m = bdna( "tcatgtaggacgggcgcaaa" );
putpdb( "1seq3_raw.pdb", m );
putpdb( "2seq3_raw.pdb", m );

m = bdna( "gcatacttagttcaatcttg" );
putpdb( "1seq4_raw.pdb", m );
putpdb( "2seq4_raw.pdb", m );

m = bdna( "aataccttatattattgtac" );
putpdb( "1seq5_raw.pdb", m );
putpdb( "2seq5_raw.pdb", m );

m = bdna( "acctaccggtcaccagccaa" );
putpdb( "1seq6_raw.pdb", m );
putpdb( "2seq6_raw.pdb", m );

m = bdna( "caatgtgcggacggcgttgc" );
putpdb( "1seq7_raw.pdb", m );
putpdb( "2seq7_raw.pdb", m );

m = bdna( "aactttcagggcctaatctg" );
putpdb( "1seq8_raw.pdb", m );
putpdb( "2seq8_raw.pdb", m );

m = bdna( "accgttctagataccgcact" );
putpdb( "1seq9_raw.pdb", m );
putpdb( "2seq9_raw.pdb", m );

m = bdna( "ctgggcaatacgaggtaatg" );
putpdb( "1seq10_raw.pdb", m );
putpdb( "2seq10_raw.pdb", m );

m = bdna( "ccagtcacccagtgtcgaac" );
putpdb( "1seq11_raw.pdb", m );
putpdb( "2seq11_raw.pdb", m );

m = bdna( "aacacctgacctaacggtaa" );
putpdb( "1seq12_raw.pdb", m );
putpdb( "2seq12_raw.pdb", m );

m = bdna( "gaggctcacataatggctct" );
putpdb( "1seq13_raw.pdb", m );
putpdb( "2seq13_raw.pdb", m );

m = bdna( "gccggcgtgcccagggtata" );
putpdb( "1seq14_raw.pdb", m );
putpdb( "2seq14_raw.pdb", m );

m = bdna( "ttaggtcagcatcagatgga" );
putpdb( "1seq15_raw.pdb", m );
putpdb( "2seq15_raw.pdb", m );

m = bdna( "ctgacatgaatctttacacc" );
putpdb( "1seq16_raw.pdb", m );
putpdb( "2seq16_raw.pdb", m );
```

```
m = bdna( "gaagcggaaacgggtgcgtg" );
putpdb( "1seq17_raw.pdb", m );
putpdb( "2seq17_raw.pdb", m );

m = bdna( "gactagcgaggagcaaacga" );
putpdb( "1seq18_raw.pdb", m );
putpdb( "2seq18_raw.pdb", m );

m = bdna( "aaattcctggcctgcttgat" );
putpdb( "1seq19_raw.pdb", m );
putpdb( "2seq19_raw.pdb", m );

m = bdna( "gtctcgtaatcttcttagag" );
putpdb( "1seq20_raw.pdb", m );
putpdb( "2seq20_raw.pdb", m );

exit( 0 );
```

APPENDIX E: NAMD SIMULATION PARAMETER FILES

These are sample simulation parameter files showing, generally, how each portion of the simulation process was set up.

Minimize.conf:

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Minimization step 1
# tacgcccaaa_ionized_Linear with 2 Linear molecules matching sequence ensembles


#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

structure
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.psf
coordinates
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.pdb

set temperature      0
set outputname
/home/cluster/NAMD/files/Length/10/Linear/Outputs/min/tacgcccaaa_Linear_min

firsttimestep      0

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################
# IMD settings for VMD interface
if {1} {
IMDon  on
IMDport      3001
IMDfreq      1
IMDwait      no
             }
# Input
paraTypeCharmm             on
parameters                 /home/cluster/NAMD/files/par_all27_na.prm
temperature         $temperature

# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              12.
switching           on
switchdist          10.
pairlistdist        13.5
```

```
# Integrator Parameters
timestep            1.0  # 1fs/step
nonbondedFreq       1
fullElectFrequency  2
stepspercycle       10

seed                41138351 #true random generated 8/15/2014

# Periodic Boundary Conditions
#measur center $everyone
#0.03466781973838806 0.003386907512322068 73.63472747802734

#measur minmax $everyone
#{-21.802000045776367 -22.2549991607666 -15.27400016784668} {21.799999237060547
22.285999298095703 162.52200317382813}

cellBasisVector1     43.602   0.0    0.0
cellBasisVector2     0.0    44.541   0.0
cellBasisVector3     0.0     0.0     177.796
cellOrigin           0.035   0.003  73.635

wrapAll             on
wrapNearest          yes
COMmotion            no

# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSpacing         1

# Output
outputName          $outputname
dcdfreq             100
xstFreq             100
outputEnergies      100
outputPressure      100


#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Minimization
minimize          10000
```

Heat.conf:

```
#################################################################
## JOB DESCRIPTION                                             ##
#################################################################

# Heat system to to 310K
# tacgcccaaa_Linear_min as input


#################################################################
## ADJUSTABLE PARAMETERS                                       ##
#################################################################

structure
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.psf
coordinates
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.pdb
bincoordinates
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/min/tacgcccaaa_Linear_min.co
or
extendedSystem
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/min/tacgcccaaa_Linear_min.xs
c

set outputname
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/heat/tacgcccaaa_Linear_heat

#################################################################
## SIMULATION PARAMETERS                                       ##
#################################################################

# Input
paraTypeCharmm             on
parameters                 /home/cluster/NAMD/files/par_all27_na.prm

temperature  0
reassignFreq 1
reassignTemp 0
reassignIncr 1
reassignHold 310

# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              12.
switching           on
switchdist          10.
pairlistdist        13.5


# Integrator Parameters
timestep            1.0  # 1fs/step
nonbondedFreq       1
fullElectFrequency  2
stepspercycle       10
```

```
seed         41138351 #true random generated 8/15/2014

wrapAll                  on
wrapNearest         yes
COMmotion           no

# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSpacing          1

# Output
outputName          $outputname
dcdfreq             100
xstFreq             100
outputEnergies      100
outputPressure      100
outputTiming   100


#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Heat over these many steps
numsteps     500
```

Equilibrate.conf:

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Equilibrate system
# tacgcccaaa_Linear_heat as input


#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

structure
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.psf
coordinates
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.pdb
bincoordinates
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/heat/tacgcccaaa_Linear_heat.
coor
extendedSystem
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/heat/tacgcccaaa_Linear_heat.
xsc
binvelocities
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/heat/tacgcccaaa_Linear_heat.
vel
```

```
set outputname
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib
set temperature     310

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################
#Margin setting
margin              2.5
# Input
paraTypeCharmm              on
parameters                  /home/cluster/NAMD/files/par_all27_na.prm

# Constant Pressure Control (variable volume)
if {1} {
useGroupPressure      yes ;# needed for 2fs steps
useFlexibleCell       no  ;# no for water box, yes for membrane
useConstantArea       no  ;# no for water box, yes for membrane

langevinPiston        on
langevinPistonTarget  1.01325 ;#  in bar -> 1 atm
langevinPistonPeriod  100.
langevinPistonDecay   50.
langevinPistonTemp    $temperature
}

# Constant Temperature Control
langevin            on    ;# do langevin dynamics
langevinDamping     5     ;# damping coefficient (gamma) of 5/ps
langevinTemp        $temperature
langevinHydrogen    no    ;# don't couple langevin bath to hydrogens


# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              12.
switching           on
switchdist          10.
pairlistdist        13.5


# Integrator Parameters
timestep            2.0  # 1fs/step
nonbondedFreq       1
fullElectFrequency  2
stepspercycle       10

seed                41138351 #true random generated 8/15/2014

wrapAll             on
wrapNearest           yes
COMmotion             no
```

```
# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSpacing         1

# Output
outputName          $outputname
dcdfreq             100
xstFreq             100
outputEnergies      100
outputPressure      100
outputTiming   100


#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Basic equilibration
numsteps     10000
```

Linear_10_run1.conf

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################
# Simulation Run 1 with 500,000 2fs time steps
#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

structure
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.psf

coordinates
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.pdb
bincoordinates
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib.coor

extendedSystem
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib.xsc

binvelocities
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib.vel
firsttimestep 0

set outputname
/home/cluster/NAMD/files/Length/10/Linear/Outputs/full/run1/Linear_run1
set temperature 310
#############################################################
```

```
## SIMULATION PARAMETERS                          ##
#####################################################################

# IMD settings for VMD interface
if {1} {

IMDon on

IMDport 3001
IMDfreq 1
IMDwait no

}
# Input

paraTypeCharmm on
parameters /home/cluster/NAMD/files/par_all27_na.prm

# Constant Temperature Control
if {1} {
langevin     on
langevinDamping     5
langevinTemp $temperature
langevinHydrogen    no
}
# Constant Pressure Control
if {1} {
useFlexibleCell           no
useConstantArea           no
langevinPiston            on
langevinPistonTarget      1.01325
langevinPistonPeriod      200.
langevinPistonDecay 100.
langevinPistonTemp  $temperature
}
useGroupPressure no ;
# Force-Field Parameters

exclude             scaled1-4
1-4scaling          1.0
cutoff              12.
switching           on
switchdist          10.
pairlistdist        13.5

# Integrator Parameters

timestep            2.0 # 1fs/step

rigidBonds                none
nonbondedFreq             1
fullElectFrequency        2
stepspercycle             10
seed                      41138351 #true random generated 8/15/2014
```

```
wrapAll                 on
wrapNearest             yes
PME                     yes

PMEGridSpacing          1


#Pressure Profile Output

if {1} {
pressureProfile         on
pressureProfileSlabs    21
pressureProfileFreq 10
}
# Output
outputName          $outputname
dcdfreq                 10
xstFreq                 1000
outputEnergies          1000
outputPressure          1000
outputTiming        100

numsteps       499999
```

Linear_10_run1_ewald.conf

```
#############################################################
## Second pass: Ewald Pressure Calculations                ##
#############################################################

#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################
structure
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.psf

coordinates
/home/cluster/NAMD/files/Length/10/Linear/PDBs/tacgcccaaa_ionized_Linear.pdb

bincoordinates
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib.coor

extendedSystem
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib.xsc

binvelocities
       /home/cluster/NAMD/files/Length/10/Linear/Outputs/equilib/tacgcccaaa_Linear_eq
uilib.vel
set outputname
/home/cluster/NAMD/files/Length/10/Linear/Outputs/full/run1/Linear_run1_Ewald
set temperature 310
```

```
#############################################################
## SIMULATION PARAMETERS ##
#############################################################
# IMD settings for VMD interface
# Input
paraTypeCharmm on
parameters /home/cluster/NAMD/files/par_all27_na.prm
# Constant Temperature Control no
if {1} {
langevin     on
langevinDamping     5
langevinTemp $temperature
langevinHydrogen    no
}
# Constant Pressure Control (variable volume) no pressure influence wanted
if {1} {
#useGroupPressure   yes ;# needed for 2fs steps
useFlexibleCell           no
useConstantArea           no
langevinPiston            on
langevinPistonTarget      1.01325
langevinPistonPeriod      200.
langevinPistonDecay 100.
langevinPistonTemp  $temperature
}
useGroupPressure no ;# needed for 2fs steps # Force-Field Parameters
exclude                   scaled1-4
1-4scaling          1.0
cutoff              12.
switching           on
switchdist          10.
pairlistdist        13.5
# Integrator Parameters
timestep            2.0 # 1fs/step
rigidBonds          none # all needed for 2fs steps
nonbondedFreq             1
fullElectFrequency        2
stepspercycle       10
seed                41138351 #true random generated 8/15/2014
wrapAll                   on
wrapNearest         yes
# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSpacing            1
outputName          $outputname

#Pressure Profile Output
if {1} {
pressureProfile           on
pressureProfileSlabs      21
pressureProfileFreq 10
pressureProfileEwald  on
pressureProfileEwaldX  20
pressureProfileEwaldY  20
pressureProfileEwaldZ  20
```

```
}

set ts 0
firstTimestep $ts

coorfile open dcd
/home/cluster/NAMD/files/Length/10/Linear/Outputs/full/run1/Linear_run1.dcd
while { [coorfile read] != -1 } {
  firstTimestep $ts
  run 0
  incr ts 10
}
coorfile close
```

APPENDIX F: PRESSUREPARSER TOOL

The PressureParser tool is a .NET Framework 4.5 tool written in Visual Basic.NET for the automatically summing pressure output files with the PME pressure output files into an Excel spreadsheet. The tool will automatically adjust for the number of slices in a data set, can batch process entire directories, automatically matching runs and outputting summed Microsoft Excel files for easy importing into analysis software. The tool has an Inputs tab for a single run, three tabs for viewing the tables (of pressures, PME, and summed data, respectively), and finally a Batch tab for automatic batch processing. It requires a 64-bit version of Microsoft Windows, and a 64-bit version of Microsoft Office to run.

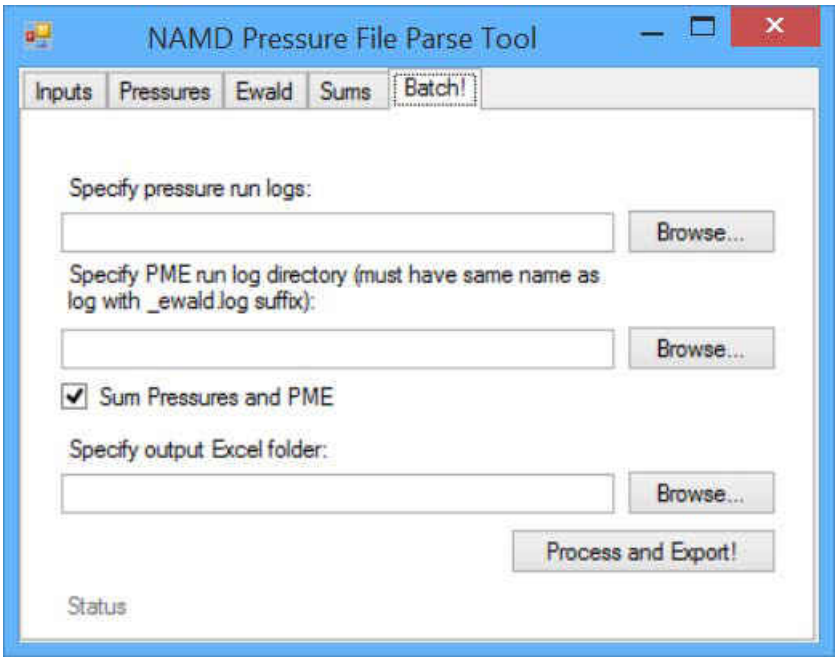


Figure 39: PressureParser tool screen shot showing batch interface

The source code for the tool is below, and the tool is divided into three code modules: the main module, the Parser.vb logic module, and the GlobalVariables module.

MainForm.vb

(This is the main code module for the PressureParser tool)

```
Imports Microsoft.Office.Interop
Imports System.IO

Public Class PressureForm
    Dim LogFileArray() As String
    Dim PMELogFileArray() As String

    Public Function GetFileName(ByVal filepath As String) As String

        'This Function Gets the name of a file without the path or extension.

        'Input:
        '   filepath - Full path/filename of file.
        'Return:
        '   GetFileName - Name of file without the extension.

        'Get indices of characters directly before and after filename
        Dim slashindex As Integer = filepath.LastIndexOf("\")
        Dim dotindex As Integer = filepath.LastIndexOf(".")

        GetFileName = filepath.Substring(slashindex + 1, dotindex - slashindex -
1)
    End Function

    Private Sub BrowsePressureButton_Click(sender As Object, e As EventArgs)
Handles BrowsePressureButton.Click
        Dim fld As New OpenFileDialog
        fld.Filter = "Log files (*.log)|*.log|All files (*.*)|*.*"
        If fld.ShowDialog() = Windows.Forms.DialogResult.OK Then
            TextBoxPressure.Text = fld.FileName
            If TextBoxOutput.Text = "" Then
                Dim tempFilename As String
                tempFilename = fld.FileName
                tempFilename = Microsoft.VisualBasic.Left(tempFilename,
tempFilename.Length - 4) & ".xlsx"
                TextBoxOutput.Text = tempFilename
            End If
        End If
    End Sub

    Private Sub BrowsePMEButton_Click(sender As Object, e As EventArgs) Handles
BrowsePMEButton.Click
        Dim fld As New OpenFileDialog
        fld.Filter = "Log files (*.log)|*.log|All files (*.*)|*.*"
        If fld.ShowDialog() = Windows.Forms.DialogResult.OK Then
            TextBoxPME.Text = fld.FileName
        End If

        If TextBoxPME.Text <> "" Then
            CheckBoxSum.Checked = True
        End If
    End Sub
```

```
            Private Sub BrowseOutputButton_Click(sender As Object, e As EventArgs) Handles
BrowseOutputButton.Click
                Dim fld As New SaveFileDialog
                fld.Filter = "Excel XML files (*.xlsx)|*.xlsx|All files (*.*)|*.*"
                If fld.ShowDialog() = Windows.Forms.DialogResult.OK Then
                    TextBoxOutput.Text = fld.FileName
                End If
            End Sub

            Private Sub ProcessButton_Click(sender As Object, e As EventArgs) Handles
ProcessButton.Click
                Dim numberOfSlices As Integer

                If CheckBoxSum.Checked Then
                    If TextBoxPressure.Text = "" Or TextBoxPME.Text = "" Then
                        MsgBox("You must specify both a pressure log and a PME log to sum
them.")
                        Exit Sub
                    End If
                End If

                'Clear the table
                GlobalVariables.PressureTable.Clear()

                'Get the number of slices

                'MsgBox("Number of slices: " +
Parser.GetSlicesFromFile(TextBoxPressure.Text).ToString)
                numberOfSlices = Parser.GetSlicesFromFile(TextBoxPressure.Text)
                If numberOfSlices = "-1" Then
                    MsgBox("Failed to detect number of slices from file.  Is this a NAMD
log?")
                    Exit Sub
                Else
                    'Initialize the tables
                    InitTables(numberOfSlices)
                End If

                'Parse the pressure file
                Parser.ParsePressureFile(TextBoxPressure.Text,
GlobalVariables.PressureTable)

                'If there's a PME file, parse it
                If TextBoxPME.Text = "" Then
                    'Do nothing, don't parse
                Else
                    'Clear the table
                    GlobalVariables.EwaldTable.Clear()

                    'Parse the PME file
                    Parser.ParsePressureFile(TextBoxPME.Text, GlobalVariables.EwaldTable)
                    ExportPMEButton.Enabled = True
                End If

                If CheckBoxSum.Checked = True Then
                    'Sum the two
                    For r = 0 To (GlobalVariables.EwaldTable.Rows.Count - 1)
                        Dim newrow As DataRow = GlobalVariables.SummedTable.NewRow()
```

```
                    newrow(0) = GlobalVariables.EwaldTable.Rows(r).Item(0)
                    For m = 1 To (numberOfSlices * 3)
                        newrow(m) = GlobalVariables.EwaldTable.Rows(r).Item(m) +
GlobalVariables.PressureTable.Rows(r).Item(m)
                    Next m
                    GlobalVariables.SummedTable.Rows.Add(newrow)
                Next r
            End If

            If CheckBoxSum.Checked = True Then
                'Save out the Summed sheet
                Dim tempExcelfilename As String
                tempExcelfilename = TextBoxOutput.Text
                ExportToExcel(GlobalVariables.SummedTable, tempExcelfilename)
                MsgBox("Excel sheet with sums saved.")
            Else
                'Save out the pressure sheet
                Dim tempExcelfilename As String
                tempExcelfilename = TextBoxOutput.Text
                ExportToExcel(GlobalVariables.PressureTable, tempExcelfilename)
            End If

            ExportButton.Enabled = True
        End Sub

        Private Function InitTables(numSlices As Integer)
            'Initialize the tables we're going to load the data into

            'First, do some cleanup
            GlobalVariables.PressureTable.Dispose()
            GlobalVariables.EwaldTable.Dispose()
            GlobalVariables.SummedTable.Dispose()

            GlobalVariables.PressureTable.Columns.Add("Timestep",
Type.GetType("System.Int32"))
            For i = 1 To numSlices
                GlobalVariables.PressureTable.Columns.Add("Pressure_" & i & "_X",
Type.GetType("System.Decimal"))
                GlobalVariables.PressureTable.Columns.Add("Pressure_" & i & "_Y",
Type.GetType("System.Decimal"))
                GlobalVariables.PressureTable.Columns.Add("Pressure_" & i & "_Z",
Type.GetType("System.Decimal"))
            Next i
            DataGridView1.DataSource = GlobalVariables.PressureTable

            GlobalVariables.EwaldTable.Columns.Add("Timestep",
Type.GetType("System.Int32"))
            For i = 1 To numSlices
                GlobalVariables.EwaldTable.Columns.Add("Pressure_" & i & "_X",
Type.GetType("System.Decimal"))
                GlobalVariables.EwaldTable.Columns.Add("Pressure_" & i & "_Y",
Type.GetType("System.Decimal"))
                GlobalVariables.EwaldTable.Columns.Add("Pressure_" & i & "_Z",
Type.GetType("System.Decimal"))
            Next i
            DataGridView2.DataSource = GlobalVariables.EwaldTable
```

```
                GlobalVariables.SummedTable.Columns.Add("Timestep",
Type.GetType("System.Int32"))
                For i = 1 To numSlices
                    GlobalVariables.SummedTable.Columns.Add("Pressure_" & i & "_X",
Type.GetType("System.Decimal"))
                    GlobalVariables.SummedTable.Columns.Add("Pressure_" & i & "_Y",
Type.GetType("System.Decimal"))
                    GlobalVariables.SummedTable.Columns.Add("Pressure_" & i & "_Z",
Type.GetType("System.Decimal"))
                Next i
                DataGridView3.DataSource = GlobalVariables.SummedTable

            End Function

            Private Sub ExportToExcel(ByVal dtTemp As DataTable, ByVal filepath As String)
                Dim strFileName As String = filepath
                If System.IO.File.Exists(strFileName) Then
                    If (MessageBox.Show("Do you want to replace from the existing file?",
"Export to Excel", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2) = System.Windows.Forms.DialogResult.Yes) Then
                        System.IO.File.Delete(strFileName)
                    Else
                        Return
                    End If

                End If
                Dim _excel As New Excel.Application
                Dim wBook As Excel.Workbook
                Dim wSheet As Excel.Worksheet

                wBook = _excel.Workbooks.Add()
                wSheet = wBook.ActiveSheet()

                Dim dt As System.Data.DataTable = dtTemp
                Dim dc As System.Data.DataColumn
                Dim dr As System.Data.DataRow
                Dim colIndex As Integer = 0
                Dim rowIndex As Integer = 0
                Dim arr As Object(,) = New Object(dt.Rows.Count + 1, dt.Columns.Count - 1)

                'Column names
                dr = dt.Rows(0)
                For Each dc In dt.Columns
                    arr(0, colIndex) = dc.ColumnName
                    colIndex = colIndex + 1
                Next

                'Data, copied to an array which will we shove into a range with a single
operation--must faster than cell-by-cell
                For r As Integer = 0 To dt.Rows.Count - 1
                    Dim dra As DataRow = dt.Rows(r)
                    For c As Integer = 0 To dt.Columns.Count - 1
                        'r+1 because 0 is occupied by the row headers
                        arr(r + 1, c) = dra(c)
                    Next
                Next

                Dim c2 As Excel.Range = wSheet.Cells(dt.Rows.Count + 1, dt.Columns.Count)
```

```
            Dim range As Excel.Range = wSheet.Range("A1", c2)

            range.Value = arr

            wSheet.Columns.AutoFit()
            wBook.SaveAs(strFileName)

            ReleaseObject(wSheet)
            wBook.Close(False)
            ReleaseObject(wBook)
            _excel.Quit()
            ReleaseObject(_excel)
            GC.Collect()

            'MessageBox.Show("File Export Successfully!")
        End Sub

        Private Sub ReleaseObject(ByVal o As Object)
            Try
                While (System.Runtime.InteropServices.Marshal.ReleaseComObject(o) > 0)
                End While
            Catch
            Finally
                o = Nothing
            End Try
        End Sub

        Private Sub ExportButton_Click(sender As Object, e As EventArgs) Handles
ExportButton.Click
            ExportToExcel(GlobalVariables.PressureTable, TextBoxOutput.Text.ToString)
        End Sub

        Private Sub ExportPMEButton_Click(sender As Object, e As EventArgs) Handles
ExportPMEButton.Click
            ExportToExcel(GlobalVariables.EwaldTable, TextBoxOutput.Text.ToString &
"_ewald.xlsx")
        End Sub

        Private Sub ButtonSelectBatchLog_Click(sender As Object, e As EventArgs)
Handles ButtonSelectBatchLog.Click
            Dim fld As New OpenFileDialog
            Dim i As Integer

            fld.Multiselect = True
            fld.Filter = "Log files (*.log)|*.log|All files (*.*)|*.*"
            If fld.ShowDialog() = Windows.Forms.DialogResult.OK Then
                ReDim LogFileArray(fld.FileNames.Count - 1)
                i = 0
                For Each file In fld.FileNames
                    TextBoxBatchPressure.Text = TextBoxBatchPressure.Text + "|" +
file.ToString

                    LogFileArray(i) = file.ToString
                    i = i + 1
                Next file
            End If
        End Sub
```

```
            Private Sub ButtonSelectBatchPME_Click(sender As Object, e As EventArgs)
Handles ButtonSelectBatchPME.Click
                Dim fld As New FolderBrowserDialog
                If fld.ShowDialog() = Windows.Forms.DialogResult.OK Then
                    TextBoxBatchPME.Text = fld.SelectedPath
                    CheckBoxBatchSum.Checked = True
                End If
            End Sub

            Private Sub ButtonSelectBatchOutput_Click(sender As Object, e As EventArgs)
Handles ButtonSelectBatchOutput.Click
                Dim fld As New FolderBrowserDialog
                If fld.ShowDialog() = Windows.Forms.DialogResult.OK Then
                    TextBoxExcelOutput.Text = fld.SelectedPath
                    CheckBoxBatchSum.Checked = True
                End If
            End Sub

            Private Sub ButtonBatchProcess_Click(sender As Object, e As EventArgs) Handles
ButtonBatchProcess.Click
                Dim tempFilename As String
                Dim numberOfSlices As Integer

                If CheckBoxBatchSum.Checked = True Then
                    If TextBoxBatchPME.Text = "" Then
                        MsgBox("You must select a PME directory.")
                        Exit Sub
                    End If

                    If TextBoxExcelOutput.Text = "" Then
                        MsgBox("You must select an Excel output directory.")
                        Exit Sub
                    End If

                    ReDim PMELogFileArray(LogFileArray.Length - 1)

                    For i = 0 To (LogFileArray.Length - 1)
                        If LogFileArray(i).Contains(".log") = True Then
                            'Add the PME log file names to that array
                            tempFilename = LogFileArray(i)
                            PMELogFileArray(i) = TextBoxBatchPME.Text + "\" +
GetFileName(tempFilename) + "_ewald.log"
                        Else
                            MsgBox("File: " + LogFileArray(i).ToString + " is not a .log
file.  Please change your log file selection and try again.")
                            Exit Sub
                        End If
                    Next i

                    'Check to see if the Corresponding PME files exist
                    For j = 0 To (PMELogFileArray.Length - 1)
                        If File.Exists(PMELogFileArray(j)) Then
                            ' File exists, do nothing
                        Else
                            MsgBox("PME file: " + PMELogFileArray(j) + " does not exist.")
                            Exit Sub
                        End If
                    Next j
```

```
            End If

            'OK, file list built and existence of PME files is confirmed, let's start
loading and calculating


            'First, get the number of slices from the first file
            numberOfSlices = Parser.GetSlicesFromFile(LogFileArray(0))
            If numberOfSlices = "-1" Then
                MsgBox("Failed to detect number of slices from file.  Is this a NAMD
log?")
                Exit Sub
            Else
                'Initialize the tables
                InitTables(numberOfSlices)
            End If

            For k = 0 To (LogFileArray.Length - 1)
                'For k = 0 To 0
                'Clear the table
                GlobalVariables.PressureTable.Clear()

                'Parse the pressure file
                StatusLabel.Text = "Parsing pressure file #" + (k + 1).ToString
                StatusLabel.Refresh()
                Parser.ParsePressureFile(LogFileArray(k),
GlobalVariables.PressureTable)

                'If there's a PME file, parse it, then sum it
                If CheckBoxBatchSum.Checked = True Then
                    'Clear the tables
                    GlobalVariables.EwaldTable.Clear()
                    GlobalVariables.SummedTable.Clear()

                    'Parse the PME file
                    StatusLabel.Text = "Parsing PME file #" + (k + 1).ToString
                    StatusLabel.Refresh()
                    Parser.ParsePressureFile(PMELogFileArray(k),
GlobalVariables.EwaldTable)
                    ExportPMEButton.Enabled = True

                    'Sum the two
                    StatusLabel.Text = "Summing"
                    StatusLabel.Refresh()
                    For r = 0 To (GlobalVariables.EwaldTable.Rows.Count - 1)
                        Dim newrow As DataRow = GlobalVariables.SummedTable.NewRow()
                        newrow(0) = GlobalVariables.EwaldTable.Rows(r).Item(0)
                        For m = 1 To (numberOfSlices * 3)
                            newrow(m) = GlobalVariables.EwaldTable.Rows(r).Item(m) +
GlobalVariables.PressureTable.Rows(r).Item(m)
                        Next m
                        GlobalVariables.SummedTable.Rows.Add(newrow)
                    Next r

                End If

                'Save it out
                If CheckBoxBatchSum.Checked = True Then
```

```
                    'Save out the Summed sheet
                    StatusLabel.Text = "Saving Excel file #" + (k + 1).ToString
                    StatusLabel.Refresh()
                    Dim tempExcelfilename As String
                    tempExcelfilename = TextBoxExcelOutput.Text + "\" +
GetFileName(LogFileArray(k).ToString) + "_summed.xlsx"
                    ExportToExcel(GlobalVariables.SummedTable, tempExcelfilename)
                Else
                    'Save out the pressure sheet
                    StatusLabel.Text = "Completed saving file #" + (k + 1).ToString
                    StatusLabel.Refresh()
                    Dim tempExcelfilename As String
                    tempExcelfilename = TextBoxExcelOutput.Text + "\" +
GetFileName(LogFileArray(k).ToString) + ".xlsx"
                    ExportToExcel(GlobalVariables.PressureTable, tempExcelfilename)
                End If
            Next k
            StatusLabel.Text = "Export Complete"
        End Sub
    End Class
```

Parser.vb

(This is the parser module)

```
    Imports System.IO

    Public Class Parser
        Public Shared Function GetSlicesFromFile(filename As String) As Integer
            'Gets the number of slices from the log file, returns -1 if the number
cannot be determined
            Try
                Using sr As New StreamReader(filename)
                    Dim line As String
                    Do While sr.Peek <> -1
                        line = sr.ReadLine

                        If line.Contains("Info:       NUMBER OF SLABS:") Then
                            Dim words As String() = line.Split(New Char() {" "})

                            Dim word As String
                            Dim numSlices As Integer

                            For Each word In words
                                If word = "Info:" Then
                                    'Do nothing, it's the beginning of the line
                                ElseIf word = "NUMBER" Then
                                    'Do nothing, it's the beginning of the line
                                ElseIf word = "OF" Then
                                    'Do nothing, it's the beginning of the line
                                ElseIf word = "SLABS:" Then
                                    'Do nothing, it's the beginning of the line
                                ElseIf word = "" Then
                                    'Do nothing, it's the end of the line
                                Else
```

```
                                    'Do something, it's the number
                                    numSlices = Integer.Parse(word)
                                    Return numSlices
                                End If
                            Next

                        End If
                    Loop
                End Using
            Catch ex As Exception
                Dim ExceptionString As String
                ExceptionString = ex.Message.ToString
                MsgBox("The file could not be read: " & ExceptionString)
            End Try

            'Couldn't read a number of slices, so return -1
            Return -1
        End Function

        Public Shared Function ParsePressureFile(filename As String, table As
DataTable)
            'Open the pressure log for a streamreader and then search for the lines
that begin with PRESSUREPROFILE:
            'also must remember these are UNIX terminated strings (LF), not Windows
(CRLF)
            'Looking for a space-delimited block of 1 + (number of slices)*3 values (1
being position, other being a 3x3 array of pressures from slab 1 to X as x,y,z tuples)

            Try
                Using sr As New StreamReader(filename)
                    Dim line As String
                    Do While sr.Peek <> -1
                        line = sr.ReadLine

                        If line.Contains("PRESSUREPROFILE:") Then
                            Dim words As String() = line.Split(New Char() {" "})

                            Dim word As String
                            Dim position As Integer
                            Dim row As DataRow = table.NewRow()
                            position = 0 'Tables are 0-centered
                            For Each word In words
                                If word = "PRESSUREPROFILE:" Then
                                    'Do nothing, it's the beginning of the line
                                ElseIf word = "" Then
                                    'Do nothing, it's the end of the line
                                Else
                                    'Do something, it's valid
                                    row(position) = Decimal.Parse(word,
Globalization.NumberStyles.AllowDecimalPoint +
Globalization.NumberStyles.AllowLeadingSign + Globalization.NumberStyles.AllowExponent)
                                    position = position + 1
                                End If

                            Next
                            table.Rows.Add(row)
                            'Exit Sub ' Let's just do this one for sanity checking
                        End If
```

```
                Loop
            End Using
        Catch ex As Exception
            Dim ExceptionString As String
            ExceptionString = ex.Message.ToString
            MsgBox("The file could not be read: " & ExceptionString)
        End Try

    End Function
End Class
```

GlobalVariables.vb

```
Public Class GlobalVariables
    Public Shared PressureTable As New DataTable("Pressures")
    Public Shared EwaldTable As New DataTable("Ewald Pressures")
    Public Shared SummedTable As New DataTable("Sum of Pressures")
End Class
```

APPENDIX G: EXCEL PRESSURE MATCHING TOOL

The Excel pressure matching tool is used to find common frequencies between the 20 simulation runs. The statistically significant frequencies are compiled in columns “A” through “T”. When executed, the macro runs through the data and outputs the common frequencies to column “Z” in detection order.

Macro:

```
Public Sub Main()

Dim thisRange As Range, col1 As Range, col2 As Range, SigFreq As Integer, count As
Integer
Dim offsetCol As Integer, SigFreqArray()
SigFreq = 0
Set thisRange = Range(Range("A2"), Range("A" & Rows.count).End(xlUp)) 'Scan the
sheet
    For Each col1 In thisRange 'First column to compare
        For offsetCol = 0 To 19 'Scan through the columns
            For Each col2 In thisRange.Offset(, offsetCol)
                If col1 = col2 Then 'Record a match by incrementing the counter
                    count = count + 1
                    Exit For
                End If
            Next col2
        Next offsetCol

 If count = 19 Then 'Frequency matched all columns, add it to the list
    ReDim Preserve SigFreqArray(SigFreq)
        SigFreqArray(SigFreq) = col1
        SigFreq = SigFreq + 1
 End If
 count = 0

Next col1

'All done, log to Z1
Range("Z1").Resize(SigFreq).Value = Application.Transpose(SigFreqArray)

End Sub
```

APPENDIX H: MATLAB PROCESSOR FILES

ProcessLengths.m:

```
%Read in the data file

%data_file_template =
'tacgcccaaa_Linear_10_21slab_run#_summed.xlsx'; %file name with
the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaact_Linear_12_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaactagcc_Linear_16_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number

%data_file_template =
'tacgcccaaa_Parallel_10_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaact_Parallel_12_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaact_Parallel_16_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number

%data_file_template =
'control_Parallel_10_21slab_run#_summed.xlsx'; %file name with
the literal "run#" that will be replaced by the current run
number
data_file_template =
'control_Linear_10_21slab_run#_summed.xlsx'; %file name with the
literal "run#" that will be replaced by the current run number

strand_length = 10;
%strand_length = 12;
%strand_length = 16;
strand_length_string = num2str(strand_length);

type = 'Linear';
%type = 'Parallel';

%Synthesize path according to length and type
```

```
%winpath =
strcat('..\',strand_length_string,'\',type,'\Excel\');
%macpath =
strcat('../',strand_length_string,'/',type,'/Excel/');

winpath = '..\..\Control\10\Linear\Excel\';
macpath = '../../Control/10/Linear/Excel/';

numfiles = 1; %number of data files
%runnum = numfiles;%replace this with loop

warning('off','MATLAB:xlswrite:AddSheet'); %suppress specious
warnings on creating Excel sheets

for runnum=1:numfiles

    data_file = strrep(data_file_template, 'run#',
strcat('run',num2str(runnum)));

    if ismac
        path = macpath;
    else
        path = winpath;
    end

    zdata1 = xlsread(strcat(path,data_file),1,'A:A'); %read in
the step
    zdata1(:,2) = xlsread(strcat(path,data_file),1,'AH:AH');
%read in the pressure in the X/Y-axis for the run from slice 11
    pressure1=zdata1(:,2);
    times1=zdata1(:,1);

    %Begin processing/declaring constants
    Fs = (0.5e+15)/10; % (1/2)e+15 simulation steps per second /
10 (every 10 steps output)
    T = 1/Fs;   % Take sampling frequency, convert to time
    L= length(pressure1);   % will be 50000 measurements in the
Excel sheet
    t=times1*T;             % create time time vector from
t=(0:L-1)*T (steps to time translation);
    Zworking = fft(pressure1);
    Zworking(1)=[];
    f = Fs/2*linspace(0,1,L/2+1); %generates a linearly spaced
vector for frequency
    f(1)=[];
    L=length(Zworking);
    alpha=0.001; %99.9% CI
```

```
    alpha_str=num2str(alpha);

    eval(['Z' num2str(runnum) '= Zworking']);

    SampleTime = T*L; %time a single sample covers

    %Get mean and standard deviation of the raw coefficients,
then standardize them
    mean_fft_coef=mean(Zworking);
    Sz1=std(Zworking);
    fft_coef_std=(Zworking-mean_fft_coef)/Sz1;

    %We need to square the FFT coefficients to get power, this
will give us
    %coefficients with a Chi Square distribution that can be
subjected
    %to Hartley's Fmax test
    poweroutput = abs(fft_coef_std).^2;
    TotalPower=sum(poweroutput)-poweroutput(1); %Subtract DC
offset from total power
    Pos_TotalPower=TotalPower/2;        %Divide by 2 to get
power of only [positive] side, recalling that the result of FFTs
are symmetrical

    eval(['power' num2str(runnum) '= poweroutput']);

    %Calculate the Standardized Positive FFT Coefficients
    y=1;
    Pos_fft_coef_std=zeros(floor(L/2),1); %preallocate array
    while (y<floor(L/2)+1)
        Pos_fft_coef_std(y)=fft_coef_std(y); y=y+1;
    end

    eval(['Pos_fft_coef_std' num2str(runnum) '=
Pos_fft_coef_std']);

    freq = (0:(floor(L/2-1)))/(SampleTime); %find the
corresponding frequency in Hz This assumes shifted coeficients
    freq_no_offset = (1:(floor(L/2-1)))/(SampleTime); %find the
corresponding frequency in Hz

    power_zeroed_const=poweroutput;
    power_zeroed_const(1)=0; %zero out the constant term of the
power array, but leave it in, the offsets are needed to match up
to frequencies
    power_no_const=poweroutput(2:floor(L/2));%possibly an error?
    power_pos=poweroutput(1:floor(L/2));
```

```
    power_pos_no_const=poweroutput(2:floor(L/2));
    power_pos_zeroed_const=power_zeroed_const(1:floor(L/2));

    %to start Hartley's test, we have to come up with the
quantities Pk, R, and
    %the sum of residual for each variable (the sum of Pj, for
all variables
    %except j=k, and the constant term)
    %we can then do Hartley's test for each harmonic

    residuals=zeros((floor(L/2)),1); %preallocate array
    for y=1:(floor(L/2))
        residuals(y)=(Pos_TotalPower-power_pos_zeroed_const(y));
    end

    R=(L-3)/2; %total amount of relative power in these
residuals is the sum of these # of random variables
    Hstat=zeros((floor(L/2)),1); %preallocate array
    for i=1:(floor(L/2)) %Compute H-statistic
        Hstat(i)=power_pos_zeroed_const(i)/((1/R)*residuals(i));
    end

    Htestresults=zeros((floor(L/2)),1); %preallocate the array
for storing H test results
    j=1;
    while (j<(L/2))
        Htestresult = Hartley(Hstat(j),2,(L-3),alpha);
        if (Htestresult==1)
            %Got a positive (reject) value, record it; no need
to record hits, the
            %array is pre-zeroed
            Htestresults(j)=1;
        end
    j=j+1;
    end

    %Now we have our H-test results, let's put togeher an
ordered list with
    %just the significant ones

    k=1;
    z=1;
    Significant_Coefficients = 0; %zero the array
    while (k<(L/2))
        if (Htestresults(k)==0)
            %A good result, grab the resulting coefficients and
save them
```

```
Significant_Coefficients(z,1)=power_pos_zeroed_const(k);
%coefficient
            Significant_Coefficients(z,2)=freq(k); %frequency
            z=z+1;
        end
    k=k+1;
    end

    %sort the list, greatest power to least

Significant_Coefficients_Sorted=sortrows(Significant_Coefficient
s,-1);

    eval(['Significant_Coefficients' num2str(runnum) '=
Significant_Coefficients']);

    %Now, let's save out that ordered list.
    data_file_output = strrep(data_file_template, 'run#',
strcat('outputsignificantrun',num2str(runnum)));


xlswrite(data_file_output,Significant_Coefficients_Sorted,1);

    figure_file_output1 = strrep(data_file_template, 'run#',
strcat('figureALL',num2str(runnum)));
    figure_file_output2 = strrep(data_file_template, 'run#',
strcat('figureSIG',num2str(runnum)));
    figure_file_output1 = strrep(figure_file_output1, '.xlsx',
'');
    figure_file_output2 = strrep(figure_file_output2, '.xlsx',
'');

    figure;
    semilogx(freq_no_offset,power_pos_no_const,'*b');
    graph_title1=['+ Power All Coefficients '];
    graph_title2=['Data File:',strrep(data_file,'_','\_'),'
Alpha:',alpha_str,' Confidence Level'];
    title({graph_title1;graph_title2});
    saveas(gcf, strcat('./Figures/',figure_file_output1),'fig');
    saveas(gcf, strcat('./Figures/',figure_file_output1),'png');
    close(gcf);

    figure;

semilogx(Significant_Coefficients(:,2),Significant_Coefficients(
:,1),'*b');
```

```
    graph_title1=['+ Power Significant Coefficients '];
    graph_title2=['Data File:',strrep(data_file,'_','\_'),'
Alpha:',alpha_str,' Confidence Level'];
    title({graph_title1;graph_title2});
    saveas(gcf, strcat('./Figures/',figure_file_output2),'fig');
    saveas(gcf, strcat('./Figures/',figure_file_output2),'png');
    close(gcf);

    %error('End of code execution');
    %---through here

end


% Wavelet generation code, to run manually
% figure;
% title('Continuous Transform, absolute coefficients, Linear DNA
10 mer.')
% cw1 = cwt(pressure1,1:1600,'sym2','plot');
% ylabel('Scale')
% [cw1,sc] = cwt(pressure1,1:1600,'sym2','scal');
% title('Scalogram')
% ylabel('Scale')

% Example CPSD code
% cpsd(pressure12,pressure16,[],[],[],Fs)
```

ProcessSequences.m:

```
%Read in the data file

data_file_template = 'seq_Parallel_21slab_seq#_summed.xlsx';
%file name with the literal "run#" that will be replaced by the
current run number
%data_file_template = 'seq_Linear_21slab_seq#_summed.xlsx';
%file name with the literal "run#" that will be replaced by the
current run number

type = 'Parallel';
%type = 'Linear';

%Synthesize path according to length and type
winpath = strcat('..\',type,'\Excel\');
macpath = strcat('../',type,'/Excel/');
```

```
numfiles = 20; %number of sequences
%runnum = numfiles;%replace this with loop

warning('off','MATLAB:xlswrite:AddSheet'); %suppress specious
warnings on creating Excel sheets

for seqnum=1:numfiles

    data_file = strrep(data_file_template, 'seq#',
strcat('seq',num2str(seqnum)));

    if ismac
        path = macpath;
    else
        path = winpath;
    end

    zdata1 = xlsread(strcat(path,data_file),1,'A:A'); %read in
the step
    zdata1(:,2) = xlsread(strcat(path,data_file),1,'AH:AH');
%read in the pressure in the X/Y-axis for the run from slice 11
    pressure1=zdata1(:,2);
    times1=zdata1(:,1);

    %Begin processing/declaring constants
    Fs = (0.5e+15)/10; % (1/2)e+15 simulation steps per second /
10 (every 10 steps output)
    T = 1/Fs;   % Take sampling frequency, convert to time
    L= length(pressure1);   % will be 50000 measurements in the
Excel sheet
    t=times1*T;             % create time time vector from
t=(0:L-1)*T (steps to time translation);
    Zworking = fft(pressure1);
    Zworking(1)=[];
    f = Fs/2*linspace(0,1,L/2+1); %generates a linearly spaced
vector for frequency
    f(1)=[];
    L=length(Zworking);
    alpha=0.001; %99.9% CI per proposal
    alpha_str=num2str(alpha);

    eval(['Z' num2str(seqnum) '= Zworking']);

    SampleTime = T*L; %time a single sample covers
```

```
    %Get mean and standard deviation of the raw coefficients,
then standardize them
    mean_fft_coef=mean(Zworking);
    Sz1=std(Zworking);
    fft_coef_std=(Zworking-mean_fft_coef)/Sz1;

    eval(['fft_coef_std' num2str(seqnum) '= fft_coef_std']);

    %We need to square the FFT coefficients to get power, this
will give us
    %coefficients with a Chi Square distribution that can be
subjected
    %to Hartley's Fmax test
    %poweroutput = abs(fft_coef_std(1:floor(L/2))).^2;
    poweroutput = abs(fft_coef_std).^2;
    TotalPower=sum(poweroutput)-poweroutput(1); %Subtract DC
offset from total power
    Pos_TotalPower=TotalPower/2;        %Divide by 2 to get
power of only [positive] side, recalling that the result of FFTs
are symmetrical

    eval(['power' num2str(seqnum) '= poweroutput']);

    %Calculate the Standardized Positive FFT Coefficients
    y=1;
    Pos_fft_coef_std=zeros(floor(L/2),1); %preallocate array
    while (y<floor(L/2)+1)
        Pos_fft_coef_std(y)=fft_coef_std(y); y=y+1;
    end

    eval(['Pos_fft_coef_std' num2str(seqnum) '=
Pos_fft_coef_std']);

    freq = (0:(floor(L/2-1)))/(SampleTime); %find the
corresponding frequency in Hz This assumes shifted coeficients
    freq_no_offset = (1:(floor(L/2-1)))/(SampleTime); %find the
corresponding frequency in Hz

    power_zeroed_const=poweroutput;
    power_zeroed_const(1)=0; %zero out the constant term of the
power array, but leave it in, the offsets are needed to match up
to frequencies
    power_no_const=poweroutput(2:floor(L/2));%possibly an error?
    power_pos=poweroutput(1:floor(L/2));
    power_pos_no_const=poweroutput(2:floor(L/2));
    power_pos_zeroed_const=power_zeroed_const(1:floor(L/2));
```

```
    %to start Hartley's test, we have to come up with the
quantities Pk, R, and
    %the sum of residual for each variable (the sum of Pj, for
all variables
    %except j=k, and the constant term)
    %we can then do Hartley's test for each harmonic

    residuals=zeros((floor(L/2)),1); %preallocate array
    for y=1:(floor(L/2))
        residuals(y)=(Pos_TotalPower-power_pos_zeroed_const(y));
    end

    R=(L-3)/2; %total amount of relative power in these
residuals is the sum of these # of random variables
    Hstat=zeros((floor(L/2)),1); %zero out array
    for i=1:(floor(L/2)) %Compute H-statistic
        Hstat(i)=power_pos_zeroed_const(i)/((1/R)*residuals(i));
    end

    Htestresults=zeros((floor(L/2)),1); %preallocate the array
for storing H test results
    j=1;
    while (j<(L/2))
        Htestresult = Hartley(Hstat(j),2,(L-3),alpha);
        if (Htestresult==1)
            %Got a positive (reject) value, record it; no need
to record hits, the
            %array is pre-zeroed
            Htestresults(j)=1;
        end
    j=j+1;
    end

    %Now we have our H-test results, let's put togeher an
ordered list with
    %just the significant ones

    k=1;
    z=1;
    Significant_Coefficients = 0; %zero the array
    while (k<(L/2))
        if (Htestresults(k)==0)
            %A good result, grab the resulting coefficients and
save them

Significant_Coefficients(z,1)=power_pos_zeroed_const(k);
%coefficient
```

```
            Significant_Coefficients(z,2)=freq(k); %frequency
            z=z+1;
        end
    k=k+1;
    end

    %sort the list, greatest power to least

Significant_Coefficients_Sorted=sortrows(Significant_Coefficient
s,-1);

    eval(['Significant_Coefficients' num2str(seqnum) '=
Significant_Coefficients']);

%     %Now, let's save out that ordered list.
    data_file_output = strrep(data_file_template, 'seq#',
strcat('outputsignificantseq',num2str(seqnum)));


xlswrite(data_file_output,Significant_Coefficients_Sorted,1);

    figure_file_output1 = strrep(data_file_template, 'seq#',
strcat('figureALL',num2str(seqnum)));
    figure_file_output2 = strrep(data_file_template, 'seq#',
strcat('figureSIG',num2str(seqnum)));
    figure_file_output1 = strrep(figure_file_output1, '.xlsx',
'');
    figure_file_output2 = strrep(figure_file_output2, '.xlsx',
'');

    figure;
    semilogx(freq_no_offset,power_pos_no_const,'*b');
    graph_title1=['Pos side of Pwr Spectrum ALL Coefficients '];
    graph_title2=['Data File:',strrep(data_file,'_','\_'),'
Alpha:',alpha_str,' Confidence Level'];
    title({graph_title1;graph_title2});
    saveas(gcf, strcat('./Figures/',figure_file_output1),'fig');
    saveas(gcf, strcat('./Figures/',figure_file_output1),'png');
    close(gcf);

    figure;

semilogx(Significant_Coefficients(:,2),Significant_Coefficients(
:,1),'*b');
    graph_title1=['Pos side of Pwr Spectrum Significant
Coefficients '];
```

```
    graph_title2=['Data File:',strrep(data_file,'_','\_'),'
Alpha:',alpha_str,' Confidence Level'];
    title({graph_title1;graph_title2});
    saveas(gcf, strcat('./Figures/',figure_file_output2),'fig');
    saveas(gcf, strcat('./Figures/',figure_file_output2),'png');
    close(gcf);

end

% Compute the CI -- a circle surrounding the vector (see Thibos)
prob = 1 - alpha;
df1 = 2;
df2 = (L-3);
F_table = finv(prob,df1,df2);
rho = sqrt((F_table/R)*(residuals(1)));

%Prepare Complete FFT coefficients for testing
for seqnum=1:numfiles

eval(strcat('Pos_fft_coef_std_combined(:,',num2str(seqnum),') =
Pos_fft_coef_std',num2str(seqnum),';'));
end
Pos_fft_coef_std_combined_norm =
arrayfun(@norm,Pos_fft_coef_std_combined);
Pos_fft_coef_std_combined_real =
arrayfun(@real,Pos_fft_coef_std_combined);
Pos_fft_coef_std_combined_complex =
arrayfun(@imag,Pos_fft_coef_std_combined);

Pos_fft_coef_std_combined_norm =
rot90(Pos_fft_coef_std_combined_norm);
Pos_fft_coef_std_combined_real =
rot90(Pos_fft_coef_std_combined_real);
Pos_fft_coef_std_combined_complex =
rot90(Pos_fft_coef_std_combined_complex);


%Process according to CI and check that no Pdist is inside the
CI
overlaptest = zeros(floor(L/2),1);
outputnum = 1;
for freqnum=1:(floor(L/2))

    for rownum=1:numfiles
        basecoef =
Pos_fft_coef_std_combined_real(rownum,freqnum);
        countup = 0;
```

```
        for pagenum=1:numfiles
            eudist = abs(basecoef -
Pos_fft_coef_std_combined_real(pagenum,freqnum));
            if (eudist <= rho) %distance is less than the
critical
                if (eudist > 0) %not the basecoef - basecoef
case
                    countup = countup + 1;
                end
            end

        end
        if (countup == 0)
            outText = sprintf('Found significant point at
sequence %d, frequency %d',rownum,freqnum);
            disp(outText);
            sigpoints(outputnum,1) = rownum;
            sigpoints(outputnum,2) = freqnum;
            sigpoints(outputnum,3) = freq(freqnum);
            outputnum = outputnum + 1;
        end
    end

end
```

StatsLoader.m:

```
%Concatenate the fourier coefficients of Zdata column 2's into a
big
%matrix, to support tests for randomness and generate graphs

% Loader
% Loads Excel sheets into workspace zdata1-20

%Read in the data file

%data_file_template =
'tacgcccaaa_Linear_10_21slab_run#_summed.xlsx'; %file name with
the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaact_Linear_12_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaactagcc_Linear_16_21slab_run#_summed.xlsx'; %file name
```

```
with the literal "run#" that will be replaced by the current run
number

%data_file_template =
'tacgcccaaa_Parallel_10_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaact_Parallel_12_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number
%data_file_template =
'tacgcccaaact_Parallel_16_21slab_run#_summed.xlsx'; %file name
with the literal "run#" that will be replaced by the current run
number

data_file_template =
'control_Parallel_10_21slab_run#_summed.xlsx'; %file name with
the literal "run#" that will be replaced by the current run
number

strand_length = 10;
%strand_length = 12;
%strand_length = 16;
strand_length_string = num2str(strand_length);

%type = 'Linear';
type = 'Parallel';

%Synthesize path according to length and type
%winpath =
strcat('..\',strand_length_string,'\',type,'\Excel\');
%macpath =
strcat('../',strand_length_string,'/',type,'/Excel/');

winpath = '..\..\Control\10\Parallel\Excel\';
macpath = '../../Control/10/Parallel/Excel/';

numfiles = 1; %number of data files

warning('off','MATLAB:xlswrite:AddSheet'); %suppress specious
warnings on creating Excel sheets

for runnum=1:numfiles

    data_file = strrep(data_file_template, 'run#',
strcat('run',num2str(runnum)));
```

```
    if ismac
        path = macpath;
    else
        path = winpath;
    end

    zdatatmp = xlsread(strcat(path,data_file),1,'A:A'); %read in
the step
    zdatatmp(:,2) = xlsread(strcat(path,data_file),1,'AH:AH');
%read in the pressure in the X/Y-axis for the run from slice 11
    eval(['zdata' num2str(runnum) '= zdatatmp']);
    eval(['zdata' num2str(runnum) '(:,2) = zdatatmp(:,2)']);
end

ZRunsResult = zeros(length(ZTest),1);
for runstestnum=1:length(ZTest)
    ZRunsResult(runstestnum,1) =
runstest(ZTest(:,runstestnum),'ud','Alpha',0.001);
end

sum(ZRunsResult(:)==1)

%
%     pressure1=zdata1(:,2);
%     times1=zdata1(:,1);

numfiles = 20; %number of data files

%zdatasum = zeros(length(zdata1),2);

alpha=0.001; %99.9% CI per proposal
alpha_str=num2str(alpha);
isMeanInited = 0; %initialize the variable, it'll grow later

for runnum=1:numfiles

%     zdatasum = zdatasum + eval(['zdata' num2str(runnum)]);


eval(strcat('pressure',num2str(runnum),'=zdata',num2str(runnum),
'(:,2);'));

eval(strcat('times',num2str(runnum),'=zdata',num2str(runnum),'(:
,1);'));

    %Begin processing/declaring constants
```

```
    Fs = (0.5e+15)/10; % (1/2)e+15 simulation steps per second /
10 (every 10 steps output)
    T = 1/Fs;   % Take sampling frequency, convert to time
    eval(strcat('L',num2str(runnum),'=
length(pressure',num2str(runnum),');'));   % will be 50000
measurements in the Excel sheet
    eval(strcat('t=times',num2str(runnum),'*T;'));              %
create time time vector from t=(0:L-1)*T (steps to time
translation);
    eval(strcat('Z',num2str(runnum),' =
fft(pressure',num2str(runnum),');'));
    eval(strcat('Z',num2str(runnum),'(1)=[];'));
    eval(strcat('f',num2str(runnum),' =
Fs/2*linspace(0,1,L',num2str(runnum),'/2+1);')); %generates a
linearly spaced vector for frequency
    eval(strcat('f',num2str(runnum),'(1)=[];'));

eval(strcat('L',num2str(runnum),'=length(Z',num2str(runnum),');'
)); %get length of Fourier transform
    if (isMeanInited == 0) %initialize the ZMean variable if it
hasn't been before
        eval(strcat('ZMean =
zeros(length(Z',num2str(runnum),'),1);'));
        eval(strcat('powerAvg =
zeros(length(Z',num2str(runnum),')/2,1);'));
        isMeanInited = 1;
    end
    eval(strcat('ZMean = ZMean + Z',num2str(runnum),';'));
%     eval(strcat('ZHotel(:,',num2str(runnum),') =
Z',num2str(runnum),';'));
    eval(strcat('power',num2str(runnum),' =
fft(pressure',num2str(runnum),')/(L',num2str(runnum),');'));
    eval(strcat('power',num2str(runnum),' =
(power',num2str(runnum),'(1:L',num2str(runnum),'/2)).^2;'));
    eval(strcat('ZHotel(:,',num2str(runnum),') =
power',num2str(runnum),';'));
    eval(strcat('powerAvg = powerAvg +
power',num2str(runnum),';'));
end

%create a combined power output to facilitate runs tests
for runnum=1:numfiles
    eval(strcat('powercombined(',num2str(runnum),',:) =
power',num2str(runnum),';'));
end
```

Exp3.m:

```
% Experiment 3 processing file; must run StatsLoader.m to load
all pressures
% Change numeral of pressure1 for each of the runs

[pxx,f,pxxc] = periodogram(pressure1,[],[],Fs,'ConfidenceLevel',
0.999);
plot(f,10*log10(pxx))
hold on
plot(f,10*log10(pxxc),'r-.')
xlabel('Hz')
ylabel('dB')
```

Exp4.m:

```
% Experiment 4 processing file; must run StatsLoader.m to load
all pressures

f8 = fit(times1,pressure1,'fourier8')
plot(f8,times1,pressure1);

f8 = fit(times2,pressure2,'fourier8')
figure
plot(f8,times2,pressure2);
```

LIST OF REFERENCES

Alder, B. J., & Wainwright, T. E. (1957). Phase Transition for a Hard Sphere System. *The Journal of Chemical Physics, 27*(5), 1208-1209. Retrieved from http://dx.doi.org/10.1063/1.1743957

Alexandrov, B. S., Gelev, V., Bishop, A. R., Usheva, A., & Rasmussen, K. Ø. (2010). DNA breathing dynamics in the presence of a terahertz field. *Physics Letters A, 374*(10), 1214-1217. doi:10.1016/j.physleta.2009.12.077

Alexandrov, B. S., Rasmussen, K. Ø., Bishop, A. R., Usheva, A., Alexandrov, L. B., Chong, S., . . . Rodriguez, G. (2011). Non-thermal effects of terahertz radiation on gene expression in mouse stem cells. *Biomed. Opt. Express, 2*(9), 2679-2689. Retrieved from http://www.opticsinfobase.org/boe/abstract.cfm?URI=boe-2-9-2679

Becker, O. M., MacKerell, A. D., Roux, B., & Watanabe, M. (Eds.). (2001). *Computational Chemistry and Biophysics* (1 ed.). New York: Marcel Dekker.

Benesty, J., Chen, J., Huang, Y., & Doclo, S. (2005). Study of the Wiener Filter for Noise Reduction *Speech Enhancement* (pp. 9-41): Springer Berlin Heidelberg.

Bhandarkar, M., Brunner, R., Chipot, C., Dalke, A., Dixit, S., Grayson, P., . . . Zhu, F. (2012). *NAMD User's Guide Version 2.9*

Blackford, J. U., Salomon, R. M., & Waller, N. G. (2009). Detecting Change in Biological Rhythms: A Multivariate Permutation Test Approach to Fourier-Transformed Data. *Chronobiology International, 26*(2), 258-281. doi:10.1080/07420520902772221

Bock, J., Fukuyo, Y., Kang, S., Phipps, M. L., Alexandrov, L. B., Rasmussen, K. Ø., . . . Usheva, A. (2010). Mammalian Stem Cells Reprogramming in Response to Terahertz Radiation. *PLoS ONE, 5*(12), e15806. doi:10.1371/journal.pone.0015806

Bradley, J. V. (1968). *Distribution-free statistical tests*: Prentice-Hall.

Brand, J. C. D. (1995). *Lines of Light: The Sources of Dispersive Spectroscopy, 1800-1930*: Gordon and Breach.

Brooks, B. R., Brooks, C. L., 3rd, Mackerell, A. D., Jr., Nilsson, L., Petrella, R. J., Roux, B., . . . Karplus, M. (2009). CHARMM: the biomolecular simulation program. *J Comput Chem, 30*(10), 1545-1614. doi:10.1002/jcc.21287

Bryce, R. A., & Hillier, I. H. (2013). Quantum Chemical Approaches: Semiempirical Molecular Orbital and Hybrid Quantum Mechanical/Molecular Mechanical Techniques. *Curr Pharm Des*. doi:Advance online publication. 10.2174/13816128113199990601

Calloway, R., Proctor, M., Boyer, V., & Napier, S. (2014). A computational study of dsDNA pairs and vibrational resonance in separating water. *Systems and Synthetic Biology, 8*(4), 329-335. doi:10.1007/s11693-014-9157-3

Calloway, R. J. (2011). *Homologous pairing through DNA driven harmonics-- a simulation investigation.* (Doctor of Philosophy in Modeling and Simulation), University of Central Florida, Orlando, FL.

Chechetkin, V. R., & Turygin, A. Y. (1995). Size-dependence of three-periodicity and long-range correlations in DNA sequences. *Physics Letters A, 199*(1–2), 75-80. doi:http://dx.doi.org/10.1016/0375-9601(95)00047-7

Christen, M., Hunenberger, P. H., Bakowies, D., Baron, R., Burgi, R., Geerke, D. P., . . . van Gunsteren, W. F. (2005). The GROMOS software for biomolecular simulation: GROMOS05. *J Comput Chem, 26*(16), 1719-1751. doi:10.1002/jcc.20303

Church, G. M., Gao, Y., & Kosuri, S. (2012). Next-Generation Digital Information Storage in DNA. *Science, 337*(6102), 1628. doi:10.1126/science.1226355

Cornell, W. D., Cieplak, P., Bayly, C. I., Gould, I. R., Merz, K. M., Ferguson, D. M., . . . Kollman, P. A. (1995). A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *Journal of the American Chemical Society, 117*(19), 5179-5197. doi:10.1021/ja00124a002

Crick, F. H. C. (1958). On Protein Synthesis. *The Symposia of the Society for Experimental Biology, 12*, 138-163.

Crick, F. H. C. (1970). Central Dogma of Molecular Biology. *Nature, 227*(5258), 561-563. Retrieved from http://dx.doi.org/10.1038/227561a0

Crick, F. H. C., Barnett, L., Brenner, S., & Watts-Tobin, R. J. (1961). General Nature of the Genetic Code for Proteins. *Nature, 192*(4809), 1227-1232. Retrieved from http://dx.doi.org/10.1038/1921227a0

Duan, Y., Wu, C., Chowdhury, S., Lee, M. C., Xiong, G., Zhang, W., . . . Kollman, P. (2003). A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *Journal of Computational Chemistry, 24*(16), 1999-2012. doi:10.1002/jcc.10349

Earl, D. J., & Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics (Incorporating Faraday Transactions), 7*, 3910. Retrieved from http://adsabs.harvard.edu/abs/2005PCCP....7.3910E

Energy, U. S. D. o. (2013). Potential Benefits of Human Genome Project Research. Retrieved from http://web.ornl.gov/sci/techresources/Human_Genome/project/benefits.shtml

Englander, S. W., Kallenbach, N. R., Heeger, A. J., Krumhansl, J. A., & Litwin, S. (1980). Nature of the open state in long polynucleotide double helices: possibility of soliton excitations. *Proc Natl Acad Sci U S A, 77*(12), 7222-7226.

Fink, H.-W., & Schonenberger, C. (1999). Electrical conduction through DNA molecules. *Nature, 398*(6726), 407-410. Retrieved from http://dx.doi.org/10.1038/18855

Foloppe, N., & MacKerell, J. A. D. (2000). All-atom empirical force field for nucleic acids: I. Parameter optimization based on small molecule and condensed phase macromolecular target data. *Journal of Computational Chemistry, 21*(2), 86-104. doi:10.1002/(SICI)1096-987X(20000130)21:2<86::AID-JCC2>3.0.CO;2-G

Guvench, O., & MacKerell, A., Jr. (2008). Comparison of Protein Force Fields for Molecular Dynamics Simulations. In A. Kukol (Ed.), *Molecular Modeling of Proteins* (Vol. 443, pp. 63-88): Humana Press.

Hendrickson, J. B. (1961). Molecular Geometry. I. Machine Computation of the Common Rings. *Journal of the American Chemical Society, 83*(22), 4537-4547. doi:10.1021/ja01483a011

Hill, T. L. (1946). On Steric Effects. *The Journal of Chemical Physics, 14*(7), 465-465. Retrieved from http://dx.doi.org/10.1063/1.1724172

Isgro, T., Phillips, J. C., Sotomayor, M., Villa, E., Yu, H., Tanner, D., & Liu, Y. (2012). NAMD Tutorial. Retrieved from http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-html/

Juncker, A. S., Jensen, L. J., Pierleoni, A., Bernsel, A., Tress, M. L., Bork, P., . . . Brunak, S. (2009). Sequence-based feature prediction and annotation of proteins. *Genome Biology, 10*(2).

Kaukinen, U., Venalainen, T., Lonnberg, H., & Perakyla, M. (2003). The base sequence dependent flexibility of linear single-stranded oligoribonucleotides correlates with the reactivity of the phosphodiester bond. *Org Biomol Chem, 1*(14), 2439-2447. doi:10.1039/B302751A

Kitano, H. (2002). Computational systems biology. *Nature, 420*(6912), 206-210. Retrieved from http://dx.doi.org/10.1038/nature01254

Klepeis, J. L., Lindorff-Larsen, K., Dror, R. O., & Shaw, D. E. (2009). Long-timescale molecular dynamics simulations of protein structure and function. *Current Opinion in Structural Biology, 19*(2), 120-127. doi:http://dx.doi.org/10.1016/j.sbi.2009.03.004

Korenstein-Ilan, A., Barbul, A., Hasin, P., Eliran, A., Gover, A., & Korenstein, R. (2008). Terahertz radiation increases genomic instability in human lymphocytes. *Radiat Res, 170*(2), 224-234. doi:10.1667/rr0944.1

Lange, O. F., van der Spoel, D., & de Groot, B. L. (2010). Scrutinizing molecular mechanics force fields on the submicrosecond timescale with NMR data. *Biophys J, 99*(2), 647-655. doi:10.1016/j.bpj.2010.04.062

Leach, A. (2001). *Molecular Modelling: Principles and Applications* (2 ed.). Harlow, England: Prentice Hall.

Lindorff-Larsen, K., Trbovic, N., Maragakis, P., Piana, S., & Shaw, D. E. (2012). Structure and Dynamics of an Unfolded Protein Examined by Molecular Dynamics Simulation. *Journal of the American Chemical Society, 134*(8), 3787-3791. doi:10.1021/ja209931w

Lock, A. J., & Bakker, H. J. (2002). Temperature dependence of vibrational relaxation in liquid H2O. *The Journal of Chemical Physics, 117*(4), 1708-1713. doi:doi:http://dx.doi.org/10.1063/1.1485966

MacKerell, A. D., Jr., Banavali, N., & Foloppe, N. (2000). Development and current status of the CHARMM force field for nucleic acids. *Biopolymers, 56*(4), 257-265. doi:10.1002/1097-0282(2000)56:4<257::aid-bip10029>3.0.co;2-w

Maduro, M. (2003). Random DNA Sequence Generator. Retrieved from http://www.faculty.ucr.edu/~mmaduro/random.htm

Manchester, K. L. (1995). Louis Pasteur (1822–1895) — chance and the prepared mind. *Trends in Biotechnology, 13*(12), 511-515. doi:http://dx.doi.org/10.1016/S0167-7799(00)89014-9

Matsumoto, A., & Olson, W. K. (2002). Sequence-Dependent Motions of DNA: A Normal Mode Analysis at the Base-Pair Level. *Biophys. J., 83*(1), 22-41. doi:http://dx.doi.org/10.1016/S0006-3495(02)75147-3

Meggers, E., Michel-Beyerle, M. E., & Giese, B. (1998). Sequence Dependent Long Range Hole Transport in DNA. *Journal of the American Chemical Society, 120*(49), 12950-12955. doi:10.1021/ja983092p

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics, 21*(6), 1087-1092. Retrieved from http://dx.doi.org/10.1063/1.1699114

Miyamoto, K.-i., Ishibashi, K.-i., Hiroi, K., Kimura, Y., Ishii, H., & Niwano, M. (2005). Label-free detection and classification of DNA by surface vibration spectroscopy in conjugation with electrophoresis. *Applied Physics Letters, 86*(5), 053902-053903. Retrieved from http://dx.doi.org/10.1063/1.1853529

Mongan, J., Case, D. A., & McCammon, J. A. (2004). Constant pH molecular dynamics in generalized Born implicit solvent. *J Comput Chem, 25*(16), 2038-2048. doi:10.1002/jcc.20139

Morse, P. M. (1929). Diatomic Molecules According to the Wave Mechanics. II. Vibrational Levels. *Physical Review, 34*(1), 57-64. Retrieved from http://link.aps.org/doi/10.1103/PhysRev.34.57

Muller, H. J. (1927). ARTIFICIAL TRANSMUTATION OF THE GENE. *Science, 66*(1699), 84-87. doi:10.1126/science.66.1699.84

Nagel, M., Bolivar, P. H., Brucherseifer, M., Kurz, H., Bosserhoff, A., & Buttner, R. (2002). Integrated THz technology for label-free genetic diagnostics. *Applied Physics Letters, 80*(1), 154-156. Retrieved from http://dx.doi.org/10.1063/1.1428619

Newland, D. E. (2012). *An Introduction to Random Vibrations, Spectral & Wavelet Analysis: Third Edition* (3rd ed.). Mineola, NY: Dover Publications.

NIH. (2000). NIH Working Definition of Bioinformatics and Computational Biology. Retrieved from http://www.bisti.nih.gov/docs/compubiodef.pdf

Noble, D. (2002). The rise of computational biology. *Nat Rev Mol Cell Biol, 3*(6), 459-463. Retrieved from http://dx.doi.org/10.1038/nrm810

Norizawa, K., Herrmann, M., Tabata, H., & Kawai, T. (2005, 19-23 Sept. 2005). *THz time-domain spectroscopy and vibration analysis of DNA-related base molecules.* Paper presented at the Infrared and Millimeter Waves and 13th International Conference on Terahertz Electronics, 2005. IRMMW-THz 2005. The Joint 30th International Conference on.

Olson, W. K., Gorin, A. A., Lu, X.-J., Hock, L. M., & Zhurkin, V. B. (1998). DNA sequence-dependent deformability deduced from protein–DNA crystal complexes. *Proceedings of the National Academy of Sciences, 95*(19), 11163-11168. doi:10.1073/pnas.95.19.11163

Ouzounis, C. A. (2012). Rise and Demise of Bioinformatics? Promise and Progress. *PLoS Comput Biol, 8*(4), e1002487. doi:10.1371/journal.pcbi.1002487

Peligrad, M., & Wu, W. B. (2010). Central limit theorem for Fourier transforms of stationary processes. 2009-2022. doi:10.1214/10-AOP530

Pevzner, P. (2000). *Computational molecular biology: an algorithmic approach*. Cambridge, Mass: MIT Press.

Peyrard, M. (2004). *Nonlinear dynamics and statistical physics of DNA* (Vol. 17). Bristol, ROYAUME-UNI: Institute of Physics.

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., . . . Schulten, K. (2005). Scalable molecular dynamics with NAMD. *J Comput Chem, 26*(16), 1781-1802. doi:10.1002/jcc.20289

Pisana, S., Lazzeri, M., Casiraghi, C., Novoselov, K. S., Geim, A. K., Ferrari, A. C., & Mauri, F. (2007). Breakdown of the adiabatic Born-Oppenheimer approximation in graphene. *Nat Mater, 6*(3), 198-201. doi:http://www.nature.com/nmat/journal/v6/n3/suppinfo/nmat1846_S1.html

Plazanet, M., Fukushima, N., & Johnson, M. R. (2002). Modelling molecular vibrations in extended hydrogen-bonded networks – crystalline bases of RNA and DNA and the nucleosides. *Chemical Physics, 280*(1–2), 53-70. doi:http://dx.doi.org/10.1016/S0301-0104(02)00441-X

Porcar, M., Danchin, A., de Lorenzo, V., dos Santos, V., Krasnogor, N., Rasmussen, S., & Moya, A. (2011). The ten grand challenges of synthetic life. *Systems and Synthetic Biology, 5*(1-2), 1-9. doi:10.1007/s11693-011-9084-5

Pronk, S., Páll, S., Schulz, R., Larsson, P., Bjelkmar, P., Apostolov, R., . . . Lindahl, E. (2013). GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics, 29*(7), 845-854. doi:10.1093/bioinformatics/btt055

Quake, S. R., Babcock, H., & Chu, S. (1997). The dynamics of partially extended single molecules of DNA. *Nature, 388*(6638), 151-154.

Ramsay, J. O., Graves, S., & Hooker, G. (2009). *Functional data analysis with R and MATLAB Use R!* Retrieved from http://dx.doi.org/10.1007/978-0-387-98185-7 Retrieved from http://dx.doi.org/10.1007/978-0-387-98185-7

Rief, M., Clausen-Schaumann, H., & Gaub, H. E. (1999). Sequence-dependent mechanics of single DNA molecules. *Nat Struct Mol Biol, 6*(4), 346-349. Retrieved from http://dx.doi.org/10.1038/7582

Salomon-Ferrer, R., Case, D. A., & Walker, R. C. (2013). An overview of the Amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science, 3*(2), 198-210. doi:10.1002/wcms.1121

Santamaria, R., Charro, E., Zacarías, A., & Castro, M. (1999). Vibrational spectra of nucleic acid bases and their Watson–Crick pair complexes. *J. Comp. Chem., 20*(5), 511-530. doi:10.1002/(SICI)1096-987X(19990415)20:5<511::AID-JCC4>3.0.CO;2-8

Schlecht, M. F. (1997). *Molecular Modeling on the PC*. New York: Wiley-VCH.

Shen, Y. C., Upadhya, P. C., Linfield, E. H., & Davies, A. G. (2003). Temperature-dependent low-frequency vibrational spectra of purine and adenine. *Applied Physics Letters, 82*(14), 2350-2352. Retrieved from http://dx.doi.org/10.1063/1.1565680

Shoemaker, B. A., & Panchenko, A. R. (2007). Deciphering Protein–Protein Interactions. Part II. Computational Methods to Predict Protein and Domain Interaction Partners. *PLoS Comput Biol, 3*(4), e43. doi:10.1371/journal.pcbi.0030043

Shumway, R. H., & Stoffer, D. S. (2011). *Time Series Analysis and Its Applications: With R Examples* (Third Edition ed.). New York: Springer.

Smith, J. C., Baudry, J., Hery, S., Lamy, A., Micu, A., Souaille, M. (1996). Harmonic and Anharmonic Dynamics in Proteins and Molecular Crystals. *Nonlinear Phys.*, 575-581.

Stewart, J. J. P. (2009). Application of the PM6 method to modeling proteins. *Journal of Molecular Modeling, 15*(7), 765-805. doi:10.1007/s00894-008-0420-y

Stroud, J. (2006). Automated Nucleic Acid Builder. *Make-Na.* Retrieved from http://structure.usc.edu/make-na/server.html

Swendsen, R. H., & Wang, J.-S. (1986). Replica Monte Carlo Simulation of Spin-Glasses. *Physical Review Letters, 57*(21), 2607-2609. Retrieved from http://link.aps.org/doi/10.1103/PhysRevLett.57.2607

Tamaoki, M., Yamauchi, Y., & Nakai, H. (2005). Short-time Fourier transform analysis of ab initio molecular dynamics simulation: collision reaction between CN and C4H6. *J Comput Chem, 26*(5), 436-442. doi:10.1002/jcc.20183

Tatiana, G., Dwight, W., Thomas, W. C., Tatyana, K., Boris, G., & Jeffrey, H. (2006). Terahertz Fourier transform characterization of biological materials in a liquid phase. *Journal of Physics D: Applied Physics, 39*(15), 3405. Retrieved from http://stacks.iop.org/0022-3727/39/i=15/a=028

Ten, G. N., Burova, T. G., & Baranov, V. I. (2009). Calculation and analysis of vibrational spectra of adenine–thymine, guanine–cytosine, and adenine–uracil complementary pairs in the condensed state. *Journal of Applied Spectroscopy, 76*(1), 73-81. doi:10.1007/s10812-009-9149-3

Thibos, L. N. (2003). *Fourier Analysis for Beginners* Retrieved from http://www.opt.indiana.edu/VSG/Library/FourierBook/title.html Retrieved from http://www.opt.indiana.edu/VSG/Library/FourierBook/title.html

Vreven, T., Byun, K. S., Komáromi, I., Dapprich, S., Montgomery, J. A., Morokuma, K., & Frisch, M. J. (2006). Combining Quantum Mechanics Methods with Molecular Mechanics Methods in ONIOM. *Journal of Chemical Theory and Computation, 2*(3), 815-826. doi:10.1021/ct050289g

Welch, P. D. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *Audio and Electroacoustics, IEEE Transactions on, 15*(2), 70-73. doi:10.1109/TAU.1967.1161901

Widrow, B., & Stearns, S. D. (1985). *Adaptive signal processing*: Prentice-Hall, Inc.

Wiener, N. (1964). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*: The MIT Press.

Williams, H. S., & Williams, E. H. (1904). *A History of Science: in Five Volumes. Volume IV: Modern Development of the Chemical and Biological Sciences* (Vol. 4). New York: Harper.

Woolard, D. L., Koscica, T., Rhodes, D. L., Cui, H. L., Pastore, R. A., Jensen, J. O., . . . Nuss, M. C. (1997). Millimeter wave-induced vibrational modes in DNA as a possible alternative to animal tests to probe for carcinogenic mutations. *J Appl Toxicol, 17*(4), 243-246.

Worning, P., Jensen, L. J., Nelson, K. E., Brunak, S., & Ussery, D. W. (2000). Structural analysis of DNA sequence: evidence for lateral gene transfer in Thermotoga maritima. *Nucleic Acids Research, 28*(3), 706-709. doi:10.1093/nar/28.3.706

Wu, Y., Genton, M. G., & Stefanski, L. A. (2006). A multivariate two-sample mean test for small sample size and missing data. *Biometrics, 62*(3), 877-885. doi:10.1111/j.1541-0420.2006.00533.x

Zaret, M. M., & Snyder, W. Z. (1977). Cataracts and avionic radiations. *Br J Ophthalmol, 61*(6), 380-384.