5-2024

An Investigation of Information Structures in DNA

Joel Mohrmann
University of Nebraska-Lincoln, joel.mohrmann@outlook.com

AN INVESTIGATION OF INFORMATION STRUCTURES IN DNA

by

Joel Mohrmann

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

May, 2024

AN INVESTIGATION OF INFORMATION STRUCTURES IN DNA

Joel Mohrmann, M.S.

University of Nebraska, 2024

Advisor: Khalid Sayood

The information-containing nature of the DNA molecule has been long known and observed. One technique for quantifying the relationships existing within the information contained in DNA sequences is an entity from information theory known as the average mutual information (AMI) profile. This investigation sought to use principally the AMI profile along with a few other metrics to explore the structure of the information contained in DNA sequences.

Treating DNA sequences as an information source, several computational methods were employed to model their information structure. Maximum likelihood and maximum *a posteriori* estimators were used to predict missing bases in DNA sequences. Other novel prediction methods based upon the AMI profile and its ability to evaluate the predictability of DNA bases were also developed and tested for accuracy. The AMI profile was also adjusted to account for the triplet-code nature of DNA sequences. Additionally, machine-learning techniques such as neural networks, support vector machines, and principal component analysis were used to classify different regions of DNA sequences using the AMI profile and to compare coding versus noncoding regions.

Finally, the analysis considered the relative frequency of groups of bases (known as k-mers) in DNA sequences. Arithmetic coding was explored as a way to effect the compression of DNA sequences modeled upon the relative frequency of the appearance of k-mers. It was concluded that biological information stored in DNA

is complex, yet this investigation provided methods to elucidate some of the character of the information structure of DNA sequences.

ACKNOWLEDGMENTS

It is hard to believe that nearly six years have elapsed since the start of my work on this thesis. The university is on its third president since I started six years ago, and a new chancellor has been installed, not to mention our third athletic director and a new Husker football coach. The UNL Electrical and Computer Engineering Department is in its *third* home. It almost feels like the world is a different place than it was in the Fall of 2018. Over that span of time, many people deserve sincere gratitude for the ways that they have helped me bring this project—finally—to its completion. G. K. Chesterton said that"thanks are the highest form of thought... and gratitude is happiness doubled by wonder" (*A Short History of England*, 1917, p. 59). I am glad that I can engage in that here:

First, I would like to thank my advisor, Dr. Khalid Sayood, for helping me guide this project through its many phases and for sticking with me very patiently—longsuffering, you could say—through all the ups and downs, twists and turns of the last six years. I enjoyed our many conversations, both about this research and just about anything else, in your three different offices during this period of time (except, perhaps, the meetings where I had to park far away and walk back in the cold of winter or the ones where I hadn't made any progress since last time, and even then you simply told me to just keep pressing on). I have great respect for you, and I consider it a privilege and an honor to have worked alongside you for these years and to have been your last graduate student. Here's to enjoying a well-deserved retirement!

I would also like to thank the other members of my committee, Dr. Michael Hoffman and Dr. Hasan Otu, for agreeing to read and evaluate this thesis. I also share great respect for each of them. They have contributed significantly to my

education during this degree. In fact, I have taken a combined 21 credit hours of classes from members of my committee. Dr. Hoffman pointed out to me what is probably the greatest DNA sequence of all time, GAGACAT. Over the years, I always enjoyed stopping by his office to chat when he was around. I am thankful for / the countless bits of advice / he has given me. Dr. Otu likewise gave me valuable advice and insights throughout these years and really gave me my academic introduction to the field of bioinformatics. I also appreciated the times I got to stop by and talk to him when the opportunity presented itself.

Other professors who contributed to my graduate coursework deserve a note of thanks as well. I was encouraged to take Real Analysis I and II by Dr. Sayood, and those proved to be the most challenging mathematics courses I have ever taken. I thank Dr. Mikil Foss for the many hours spent in his office helping me understand those concepts. I also thank Dr. Glenn Ledder, a mathematics professor who devoted much time to the preparation of his lectures, making them enjoyable to listen to; they were so enjoyable, in fact, that I took as many of his courses as I could before he retired (including auditing Complex Analysis during the semester that everything shut down for COVID-19). I also want to thank Dr. Andrew Harms for his excellent instruction in estimation theory; practically all of the work in Chapter 2 came out of things learned in his class. Finally, I also thank Dr. Wei Qiao for his beneficial course on computational intelligence, which contributed significantly to the work presented in Chapter 5.

I would also like to thank those from the Sayood-Otu research group, specifically Amirsalar Mansouri, Dicle Yalcin, Poupack Baghery, and Brittany Sullivan-Reicks, with whom I took classes, studied, had interesting conversations, and shared an office in a dimly lit hallway in the very back corner of the third floor of the old Scott Engineering Center, behind a door mysteriously labeled “Occult

Information Lab," which also no longer exists. Though I often wasn't around, when I was, you all made me feel welcome and made my time enjoyable.

I would also like to thank Teresa Ryans for the assistance she has given me over the past six years, from the small things, like helping reserve the room for my thesis defense and keeping me aware of various deadlines, to the large things, like helping me get re-enrolled in this degree program after my enrollment accidentally lapsed due to an oversight on my part. I would also like to extend thanks to the UNL Department of Electrical and Computer Engineering for its financial support through the Halla Fellowship in my first year of study and the UNL Graduate College for its financial support through the Dean's Fellowship in my second year of study.

For almost ten years now, I have been employed at J.A. Woollam Co., and I continued to work there all throughout this time. I wish to thank Dr. John Woollam for hiring me as a freshman intern ten years ago, for keeping me around all this time, and for supporting my educational endeavors both organizationally and financially. Thanks for providing me a job that allowed me to be able to pursue this degree, and for giving me the most flexible schedule I could ever imagine in order to make it work. I would also like to recognize several of my colleagues at J.A. Woollam Co. who supported me on this path to my degree. I would like to thank Jeremy Van Derslice, with whom I've shared an office for all but the last two months of these past six years, and Andrew Martin. Thank you both for being there to bounce ideas off of, for encouraging me to continue on even when the road ahead didn't look promising, and for the friendship along the way. I would also like to thank Ron Synowicki, with whom I obtained my first US patent during the course of this thesis, for his friendship and the many lunches we've had over these years, sharing his extensive knowledge of scientific and ellipsometric history. I would like

to thank Tom Tiwald for helping me publish my first and, to-date, only scientific journal article, written during my time as a graduate student but which had nothing to do with my graduate research represented in this thesis. Last but not least, I would like to express gratitude to James Hilfiker and Jeff Hale for being some of the best people to work both for and with and for enabling me to keep the level and scope of my work projects conducive to the ability to devote work to this thesis. Though space probably doesn't allow me to go on, more of my colleagues could be noted as well, with whom it is a pleasure to work.

I would also like to thank Matt Romer and Zach Thompson for their friendship, who started as fellow electrical engineering majors and later both became my colleagues at J.A. Woollam Co. I have greatly enjoyed our friendship and many stimulating and fruitful conversations over these years. Matt has also been my roommate for most of the time that I've been a graduate student, one of the best roommates anyone could ask for, even though he did start his master's degree in electrical engineering at UNL after I did and yet finished before me (something, I admit, which was not *that* hard to do). I remember the night right after everything shut down for COVID-19 that we went down and cleaned out our graduate student offices and brought everything home because we didn't know what was to come. Matt, you'll always be the true winner of the door contest in my opinion. Even though my sighing is audible, Zach's ability to pull a pun out of literally anything is actually appreciated. I hope he'll find the one I've hidden in this acknowledgments section.

I would also like to thank the students and leaders of Ratio Christi. It was a joy and a challenge to help start this organization with you six years ago, and I'm glad to see it continue to flourish here at UNL. I have been grateful for all of our deep, intellectually stimulating, and profitable discussions and exchange of ideas over

these years. There are many people from Ratio Christi I could mention, but I want to specifically note those who were there from the beginning, Adam Lloyd Johnson, Wesley Farewell, and Benjamin Blowers. In different ways, I have greatly appreciated your friendship, your camaraderie, your encouragement, and your kindness, which continue to this day. I would also like to thank my friends at Heritage Bible Church, too numerous to name, for specifically *not* asking me about how my research was going all the time (except for Dan Santos and Josh Williams on April Fools' Day). I appreciated that more than you know, as well as the great encouragement and joy you have all brought to my life. I would also like to thank Susan Dittman for her support of this work and all the prayers she has offered on my behalf and for this research work—I am grateful!

I would like to thank those who prepared me to be where I am today. That includes the Jeffrey S. Raikes School of Computer Science and Management for selecting me to be a part of that program and for teaching me so much about software development over the sometimes intense four years that I spent there, which definitely equipped me to be able to write the 9,105 lines of code that I produced for this thesis. Going back even further, I want to acknowledge the contributions of all my high school teachers at Lincoln Christian School for their impact on my life, again too many to name here, but I'll at least list those whose educational instruction laid the foundations for the career path I have chosen and the work I have developed in this thesis: Mr. Sam Nelson, my physics and chemistry teacher; Mr. Randy Reinke, my biology teacher; and Mrs. Tara Free, my calculus teacher.

I would like to thank Dr. Hunter Flodman, a professor of chemical engineering here at UNL, who also happens to be my third cousin, for his advice and encouragement at a few critical points during these past few years.

Lastly, at the end of this verbose composition which has gone on long enough, it would surely not be complete if I did not state that I owe a large debt of gratitude to my family. Their love and support and encouragement over these past six years has been abundant and is greatly appreciated! I would like to thank my parents, James Mohrmann (M.A., Mathematics, UNL, 1990) and Janet Mohrmann (M.S., Agricultural Economics, UNL, 1985), who both preceded me in earning master's degrees from the University of Nebraska–Lincoln, for their unwavering love and encouragement and help in so many ways over these years, whether for letting me eat countless meals at your house so I didn't have to cook for myself, or for selling me your old house and helping me move in, or for giving me advice and guidance as I plan my life and career, I have appreciated it all. I would also like to thank my sister, formerly Jenna Mohrmann and now Jenna House, and her husband Josiah House who joined our family in the midst of my journey to this thesis. I have enjoyed your love and encouragement as well and the great times we have spent and experiences we have shared together through it all.

To all those whom I've mentioned here and for many more which the need to draw this section to a close did not allow me to name, thank you from the bottom of my heart. Finally, as the great composer Johann Sebastian Bach ended his magnificent compositions, *Soli Deo Gloria*.

"Magna opera Domini exquisita in omnes voluntates ejus." – James Clerk Maxwell

(*European Journal of Physics*, Volume 8, p. 235)

Table of Contents

List of Figures

List of Tables

Chapter 1

Background

1.1 Introduction

Deoxyribonucleic acid (DNA) is the molecule in the cells of all living organisms that contains the information necessary for those organisms to function and reproduce. DNA is both a code—a recipe for life—and a manufacturing platform with instructions for building and regulating the proteins that constitute much biological material. DNA can be described as a "digital" code, meaning that it is composed of discrete entities called bases (or nucleotides) which correspond to specific chemical structures in the molecule. The chemical names of these bases are adenine, cytosine, guanine, and thymine, and they are usually represented as "A," "C," "G," and "T," respectively, when DNA is transcribed as a code. Inside the cells of living organisms, DNA is organized into long strands called chromosomes, which can be up to millions of bases long. As a digital code containing the information necessary for life, DNA has a highly organized structure and specifically ordered content. The structure of the information present in the DNA molecule is not readily evident to the human eye; in other words, the language of DNA is not a language that human beings are able to read directly.

Since the discovery of DNA in the middle part of the twentieth century (often attributed to James Watson and Francis Crick), scientists have been devising methods to be able to sequence DNA's code and interpret what that code says. Much of this work has involved the use of biological "wet-lab" experimentation. The location of genes or other informationally significant segments on a DNA strand has traditionally been found by physical processes and methods that serve as indicators that an expressed gene originates from a specific location on a chromosome. In order to better understand the function of DNA in various organisms, a study of each organism's genome, the complete set of chromosomes for that particular organism, was desired. Beginning just over 30 years ago, a concerted effort was made to determine the base content, as far as possible, of all chromosomes in the genomes in a multitude of species, most notably human beings (via the Human Genome Project). The development of DNA sequencing methods has allowed databases containing the contents of these genomes to be constructed. As a result, this has made the information contained in the DNA molecule of many organisms available for further research and analysis.

Of particular interest in the study of the information content of DNA is the determination of which regions of the chromosomes are used by the cell to make proteins. Various sections of DNA strands, known as "coding regions," are used by the cell as the "blueprints" to construct various protein molecules via a transcription and translation process. However, other regions of DNA strands, known as "noncoding regions," have other various uses or else no known use. With regard to protein coding, the triplet structure of DNA is well-known, and biological "wet-lab" experimentation has yielded the ability to discover which regions of DNA strands code for proteins and which do not. While the basic structure of DNA with respect to the locations of coding and noncoding regions is known, there are

currently other aspects of the information structure of DNA that are potentially unrecognized, especially in noncoding regions where the DNA is not generally expressed through translation in the cells of the organism. Given the interest in knowing the function of various DNA regions, a method that could uncover the underlying information structure present in DNA sequences by segmenting DNA sequences into segments that correspond to biological realities without recourse to indirect methods, such as studying gene expression in cells, would be highly useful.

1.2 Information Theory

The ability to segment DNA strands into biologically significant sections by understanding the DNA's underlying information structure holds potential for new discoveries in biology and bioinformatics research. Developing a mechanism to segment DNA would require a method to determine which bases on the DNA strand are more informationally significant than other bases. For example, if the DNA strand were compared to written text in a paragraph, a method to determine that the word "the" is less informationally significant than a word like "exquisite" would be required. In short, to discover the biologically significant segments of the DNA strand, some way of knowing its information structure is needed.

Through biological research, it is known that DNA has a highly organized structure and specifically ordered content. In other words, DNA contains information. Consequently, some success has been achieved analyzing DNA using methods developed to analyze information in general, such as with the concept of average mutual information.[1] Collectively, these methods derive from the field of "information theory," which is a field of study related to the representation of information that arose from the study of how to communicate information

effectively.[2] To be able to compress information, something about its structure and specifically its redundancy must be known. Once that structure is characterized, it can be exploited to reduce the symbols used to represent the information to a representation closer to its absolute minimum. The more that is known about the structure of the information, the closer to the theoretical minimum the compression can achieve.

1.2.1 Self Information

Information theory as developed by Shannon[2] relies on the fundamental intuition that the information content of a certain entity or event is inversely related to the probability that the entity or event will appear from a given source. For example, if someone were to observe the sun rising tomorrow, that would be an event that has an extremely high probability of occurring. The fact that the sun rises tomorrow is not surprising in the least, and thus that event contains very little information—it tells that person nothing new. However, if that person were to wake up tomorrow well past the time for sunrise and realize that the sun had not risen, that would have major implications. The probability of that event happening tomorrow is extremely close to zero; thus, if that event were to happen, it would contain a large amount of information; namely, that something has gone very wrong with the solar system.

Information is thus an inverse measure of how probable or improbable it is to observe a certain effect. Some have even called this measure "surprisal," how surprising or unsurprising the occurrence of a particular event is. High probability means the event is not surprising and thus has low information; inversely, low probability means the event is surprising if it occurs and thus has high information content. Thus, letting $i(x)$ be a function measuring the information content for any event x and letting $p(x)$ be the probability density function, the information (or

"self information") for a specific event $x = X$ can be described by the following:

$$i(X) \propto \frac{1}{p(X)} \tag{1.1}$$

Mathematically, then, there are three properties that this measure of information needs to satisfy: (1) An event with a probability of 1 is completely unsurprising and thus has no information. (2) The less probable an event is, the more surprising it is and thus the more information it has. (3) If two independent events are measured separately, the total amount of information contained in their combination must be the sum of the individual informations of the individual events. It was shown by Shannon that this relationship between information and probability can thus be expressed with logarithms such that it satisfies these three properties.[2]

$$i(X) = \log_2\left(\frac{1}{p(X)}\right) = -\log_2 p(X) \tag{1.2}$$

Because of the inverse relationship in Equation (1.2), property (2) is satisfied since, as the probability increases, the information decreases, and *vice versa*. If the probability of an event is 1, $p(X) = 1$ and thus $i(X) = -\log_2(1) = 0$, satisfying property (1). Finally, for two independent events, their probability $p(XY)$ can be expressed with a product: $p(XY) = p(X)p(Y)$. Applying this to Equation (1.2) yields the following which satisfies property (3):

$$\begin{aligned} i_T(XY) &= -\log_2 p(XY) = -\log_2 \left(p(X)p(Y)\right) \\ &= -\log_2 p(X) + -\log_2 p(Y) = i(X) + i(Y) \end{aligned} \tag{1.3}$$

Thus, the information contained in two independent events is just the sum of their individual informations. A logarithm of any base can be chosen, but $\log_2$ is most often used because it quantifies information represented in bits, since a base-2 number system only has two digits, 0 and 1.

Using this definition of information, the information contained in DNA sequences can now be quantified. When calculating information values for DNA sequences, each base in the DNA sequence is treated as an event that can occur with a certain probability. For example, just considering the short DNA sequence of bases GAGACAT, the probability of the appearance of C and T, given this short snippet alone, would be $p(C) = p(T) = 1/7$, which is low. Thus, the information content of bases C and T is high, since each only appears once. Per Equation (1.2), the information of C and T is $i(C) = i(T) \approx 2.807$. On the other hand, since the base A appears three times in this sequence, it has the highest probability of occurrence ($p(A) = 3/7$) and thus its information content is lower at $i(A) \approx 1.222$.

1.2.2 Mutual Information

Often, events carry information not only about themselves but about other related events as well. When discussing independent events and deriving Equation (1.3), it was noted that the total information contained jointly in both of the events was just the sum of the information contained in each of the events. While this is true for *independent* events, it is not true for events that are not probabilistically independent. If the two events are independent, then knowing about one event does not give any information about the other event. However, if events are probabilistically dependent on each other, that means each event also contains some information about the other. This is known as *mutual information*.[3]

Before defining mutual information mathematically, a clarification needs to be made. When determining the total information contained in events X and Y, the *joint* probability is used. The joint probability is the probability that events X and Y *both* occur. Thus, when representing the associated probabilities using a Venn diagram, the joint probability represents the overlapping section, as shown in Figure 1.1. However, when determining the information based on the joint probability as in Equation (1.3), the *total* information contained in events X and Y is determined. In other words, when X and Y are considered in terms of information content using a Venn diagram, the whole Venn diagram is shaded as in Figure 1.2. This is because the total information contained in events X and Y is contingent on the state of affairs where both X and Y occur; thus, it is calculated from their joint (intersection) probability rather than their union probability. However, in the "information space," so to speak, it produces a union quantity rather than an intersection quantity.

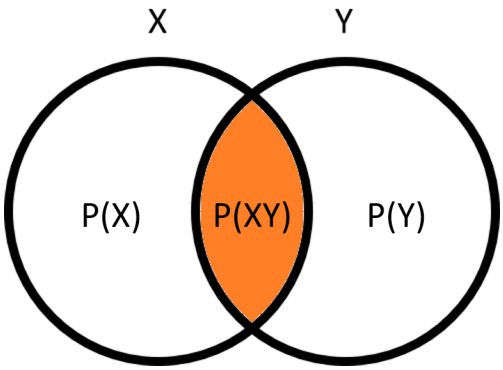

Figure 1.1: Venn-diagram representation of the joint probability of two events, in other words, the probability that both events occur.

The intersection quantity in the "information space" is mutual information, the information shared by two events, as shown in the Venn-diagram representation of Figure 1.3. In order to calculate this quantity, the self-informations of events X and

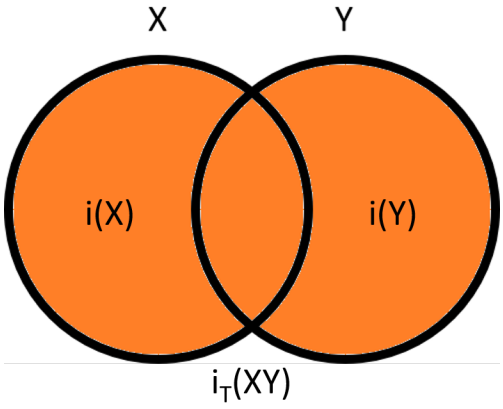

Figure 1.2: Venn-diagram representation of the total information of two events, in other words, the total information content of both events X and Y.

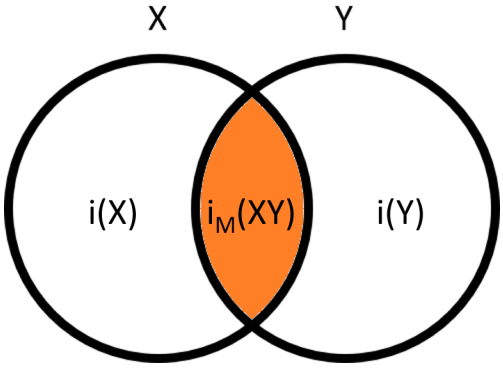

Figure 1.3: Venn-diagram representation of the mutual information of two events, in other words, the information that both events share about each other.

Y can be summed, and then the total information contained in the joint event X and Y can be subtracted. In the case of dependent events which contain redundant information about each other, the sum of self-informations will be larger than the total information. Subtracting the information in the joint event will then yield the redundant part, which is equal to the information that both events share (the overlap in the Venn diagram in Figure 1.3), i.e., the mutual information, as described by Equation (1.4).

$$i_M(XY) = i(X) + i(Y) - i_T(XY)$$
$$= -\log_2 p(X) + -\log_2 p(Y) - (-\log_2 p(XY)) = \log_2 \frac{p(XY)}{p(X)p(Y)} \quad (1.4)$$

In the case of independent events, if the result of Equation (1.3) for $i_T(XY)$ is substituted in Equation (1.4), it is found that the mutual information contained in two independent events is zero, and this makes intuitive sense. This means that, while independent events are *not* mutually exclusive with respect to probability, they are mutually exclusive with respect to their information content, as represented in Figure 1.4. Thus, knowing information about event X does not give any information about event Y, and *vice versa*. Dependent events produce an "information space" that has redundancy and thus would show overlap in the Venn-diagram representation as in Figure 1.3; in other words, probabilistically dependent events are not mutually exclusive with respect to information. Thus, some part of the information content about event X is also information about event Y, and *vice versa*.

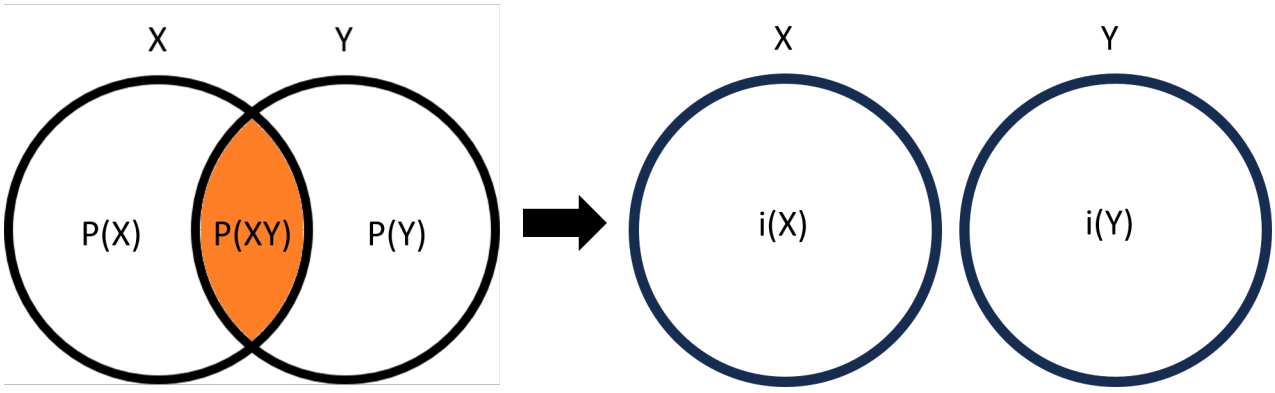

Figure 1.4: Venn-diagram representation of independent events both with respect to their probabilities and their information content.

In summary, this means that i_T, the total information, represents a union in the "information space" but is directly calculated from the joint probability, which is an

intersection in the "probability space." The mutual information shared between two events, i_M, represents the intersection relationship in the "information space" and must be calculated by finding the difference between the sum of each event's self-information and the total information contained in both events together.

Using the concept of mutual information, the relationship that is shared between bases in DNA sequences can be quantified. For example, consider a short DNA sequence of bases AGGACGAGACATG. The sequence has a length of 13 total bases and contains five A bases and five G bases, meaning that the probability of seeing an A base is estimated at $p(A) = 5/13$ and the probability of seeing a G is the same, $p(A) = 5/13$. The joint probability of seeing an A base and then a G base immediately after it can also be considered. To calculate this joint probability, it must be noted that while there are 13 individual bases, there are only 12 complete pairs of bases, and of these pairs, only two of them are an A followed by a G base (see Figure 1.5). This means that the joint probability of observing an A base next to a G base is $p_1(AG) = 2/12$. Using this, the mutual information shared between bases A and G, given that A and G appear next to each other in that order, can be calculated using Equation (1.4), which produces the result in Equation (1.5):

$$i_{M1}(AG) = \log_2 \frac{p_1(AG)}{p(A)p(G)} = \log_2 \frac{2/12}{(5/13)(5/13)} \approx 0.172 \tag{1.5}$$

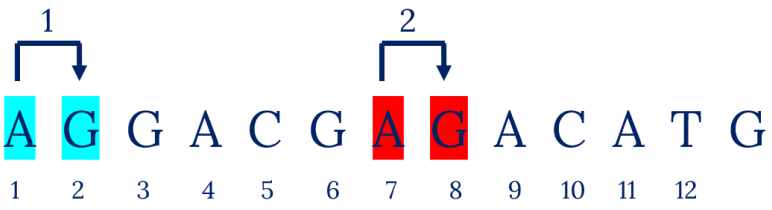

Figure 1.5: Calculation of the mutual information on a DNA sequence between bases A and G given that A and G appear next to each other in that order.

Additionally, the mutual information between bases A and G can also be calculated under a slightly modified scenario. In this case, the separation between the bases is considered, and thus, for example, the mutual information between bases A and G given that they are separated by only one base could be calculated. (In other words, bases A and G are two base positions apart). In order to do this, the joint probability of seeing a G base that occurs two base positions after an A base must be considered. To calculate this joint probability, it must be noted that while there are 13 individual bases, there are only 11 complete pairs of bases that occur one base apart, and of these pairs, three of them are an A followed by a G base (see Figure 1.6). This means that the joint probability of observing a G base one base away from an A base is $p_2(AG) = 3/11$. In this manner, the mutual information shared between bases A and G, given that A and G are one base apart, can be calculated by once again using Equation (1.4) to produce the result in Equation (1.6):

$$i_{M2}(AG) = \log_2 \frac{p_2(AG)}{p(A)p(G)} = \log_2 \frac{3/11}{(5/13)(5/13)} \approx 0.883 \tag{1.6}$$

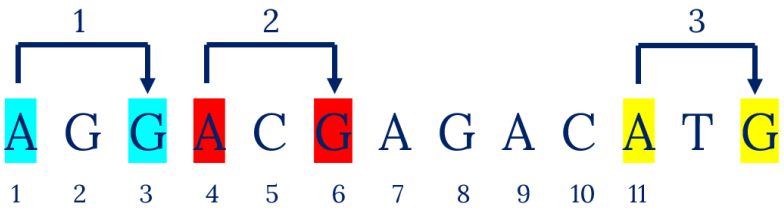

Figure 1.6: Calculation of the mutual information on a DNA sequence between bases A and G given that A and G appear one base apart from each other.

Bringing this all together, it makes sense in these examples, given the slightly lower frequency of occurrence for bases A and G next to each other (2 occurrences) compared to one base apart (3 occurrences), that the mutual information between

bases A and G is greater when they appear one base apart ($i_{M2}(AG) \approx 0.883$) than when they appear next to each other ($i_{M1}(AG) \approx 0.172$).

1.3 Average Mutual Information

A key tool for evaluating the information structure of DNA is the *average mutual information* (AMI), which is built directly upon the concept of mutual information.

1.3.1 AMI Calculation

The AMI is a weighted average of the mutual information of base pairs calculated over a whole DNA sequence (S). In order to relate the bases contained in the DNA sequence to information theory, each base is treated as an event. First, one needs to estimate the marginal probability of seeing each particular base. To calculate this marginal probability for an arbitrary base X, the total number of the times base X occurs in the DNA sequence S ($n(X)$) divided by the total number of bases in S ($n(S)$) yields this result:

$$p(X) = \frac{n(X)}{n(S)} \tag{1.7}$$

These marginal probabilities need to be calculated for each possible base in the DNA sequence: A, C, G, or T. Collectively, this set of four bases is called the DNA base alphabet, denoted by $\mathcal{B} = \{A, C, G, T\}$.

Secondly, one needs to estimate the joint probabilities for observing two bases in relation to each other. These bases can occur next to each other, or one of the bases in the pair under consideration can occur at a location k base positions downstream from the other base in question. Thus, the number of pairs of the two bases in question that occur k positions apart relative to how many total bases occur k positions apart in the whole sequence S needs to be counted. For arbitrary bases X

and Y, the number of pairs of those bases that occur k positions apart will be denoted as $n_k(X, Y)$. To obtain the total number of base pairs that occur k positions apart in the sequence, all base pairs k positions apart in the whole alphabet $\mathcal{B}$ are simply counted up as given in Equation (1.8):

$$n_k(S) = \sum_{I \in \mathcal{B}} \sum_{J \in \mathcal{B}} n_k(I, J) \tag{1.8}$$

The joint probability can then be straightforwardly estimated by dividing the number of occurrences of the base pair in question $n_k(X, Y)$ by the total number of base pairs in sequence S occurring k positions apart to obtain the joint probability for two bases X and Y k positions apart. In the following, X and Y are arbitrary bases from the alphabet $\mathcal{B}$ which is the set of bases $\{A, C, G, T\}$, $p_k(X, Y)$ is the joint probability of bases X and Y appearing k positions apart, $n_k(X, Y)$ is the number of times bases X and Y are observed k positions apart in the DNA sequence, and the denominator represents the total number of base pairs k positions apart in the DNA sequence.

$$p_k(XY) = \frac{n_k(X, Y)}{n_k(S)} = \frac{n_k(X, Y)}{\sum_{I \in \mathcal{B}} \sum_{J \in \mathcal{B}} n_k(I, J)} \tag{1.9}$$

Using the probabilities defined in Equation (1.4), the mutual information between bases X and Y can be calculated as follows:

$$i_{Mk}(XY) = \log_2 \frac{p_k(XY)}{p(X)p(Y)} \tag{1.10}$$

Finally, then, the average mutual information is a weighted average of these mutual informations for all base pairs in the DNA sequence S that occur k positions

apart. This is calculated as follows:

$$AMI_k = \sum_{X \in \mathcal{B}} \sum_{Y \in \mathcal{B}} p_k(XY) i_{Mk}(XY) = \sum_{X \in \mathcal{B}} \sum_{Y \in \mathcal{B}} p_k(XY) \log_2 \frac{p_k(XY)}{p(X)p(Y)} \tag{1.11}$$

The average mutual information represents a measure of the information which is shared between bases regardless of the specific base pair in question. In other words, it relates the information any one base likely shares, on average, with any another base given that the bases are k positions apart.

1.3.2 AMI Profile

The tools of information theory, specifically AMI, may have the potential to model the information structure of DNA sequences. In general, the more that is known about the information structure of any data source and the better the model that exists for it, the better its data can be understood and exploited by the knowledge of that information's structure. Based upon the concept of the average mutual information, a tool for analyzing DNA sequences was developed by Bauer, Schuster, and Sayood called the "AMI profile."[1] The AMI profile is simply a vector that contains the AMI for a certain set of values of k up to a certain "lag." For a DNA sequence, its AMI profile describes the differences in how much information bases share with each other as a function of their distance from each other, on average. The AMI profile is obtained by calculating the AMI between bases that are 1, 2, ... N away, forming a vector of N values describing how the bases in a DNA sequence, on average, relate to surrounding bases. An example of an AMI profile for human chromosome 19 can be seen in Figure 1.7.

The concept of the AMI profile can be used to model the information structure of DNA sequences. The AMI profile of a species' DNA can contain information

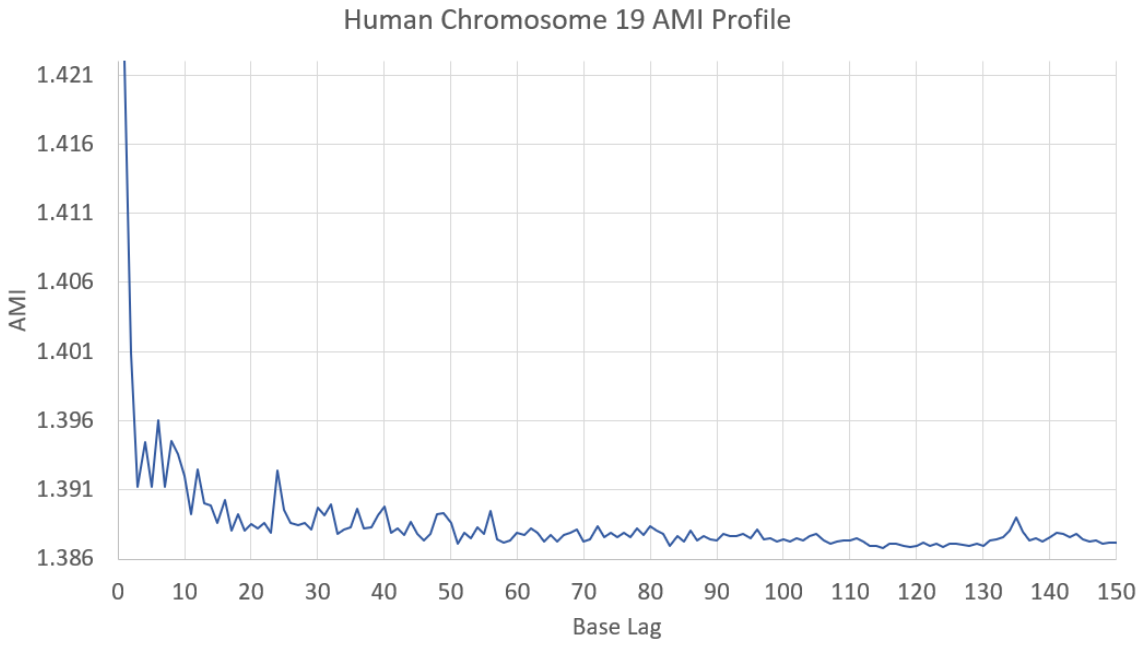

Figure 1.7: AMI profile for human chromosome 19 showing 150 base lags (k, the number of positions that separates bases on the DNA sequences for the purposes of assessing their mutual information). The actual raw value of the AMI on the Y axis is rather meaningless compared to the *relative* comparison of those values for different values of k.

about how its DNA is uniquely structured and can even act as a "species signature" for identifying from which species a DNA sequence might have originated [1]. The AMI profile, as it relates to the DNA of a specific species, has been found to have two somewhat surprising properties that make it an excellent identifier of the species from which it originates: (1) AMI profiles calculated from DNA are consistent in shape for all chromosomes from an organism. In other words, the general shape of the AMI profile does not change regardless of where in the organism's genome the DNA sequence is selected from. (2) Although AMI profiles from one organism are consistent in shape, when compared to the AMI profiles calculated from the DNA of *other* organisms, the AMI profiles have distinct shapes. This is why the AMI profile can be used as a reliable "species signature"; it is

consistent within all DNA sequences of one species, but it is also sufficiently different from the AMI profile of any other species.

The AMI profile is simply one way to quantify the information structure contained in DNA sequences, and it has been shown that the AMI profile can be reliably used to identify a species simply by the structure of its DNA information as expressed in the AMI profile.[1] Given this, it would seem feasible that the information structure of DNA is perhaps sufficiently quantifiable in such a way as to be able not only to identify a species from its AMI profile but also to recognize and interpret the significant regions of the DNA sequences themselves within each species. It is also reasonable to think that the AMI profile itself, or some adjacent or derivative concept, can be used to discover and describe that structure.

1.4 Overview

This thesis will attempt to explore the question of whether probabilistic models based on the core concepts of information theory can be used to discover the specific structure of the information contained in the DNA sequences of organisms. With the concept of the AMI profile, the relationships between the bases of DNA sequences can be analyzed and quantified with the hope of differentiating between more informationally significant regions and less informationally significant regions, perhaps between coding regions and noncoding regions, and perhaps between bases or groups of bases that form "hinge points" between differing regions of chromosomes.

This investigation has followed a somewhat indirect and meandering process of exploration which has culminated in the discovery of various insights and limitations concerning the use of information theory methods, and the AMI profile

in particular, to ascertain DNA's unique structure. The chapters of this thesis will contain the following investigations:

- In Chapter 2, an attempt is made to provide a baseline with which to compare the results obtained from using the AMI profile to predict unknown bases. Maximum likelihood and maximum *a posteriori* estimators are used to quantify how well basic estimation theory can predict unknown bases. From this result, it can be determined what threshold standard must be exceeded, and thus a metric can be produced by which to judge the success of using more sophisticated methods, such as the AMI profile.

- In Chapter 3, in order to use the AMI profile as a method of predicting unknown bases from known existing bases, an *ad hoc* method of selecting which bases to remove and which bases to keep was developed. The results of this method were an indication of how well the AMI profile could predict unknown bases in the case when the bases used to make the prediction and the bases making the prediction were chosen arbitrarily. However, a key component in compressing DNA sequences is choosing which bases can be discarded and which bases can be retained in order to get an accurate reproduction. The AMI profile was used to make this determination to see whether or not a subset of bases chosen by the AMI profile itself would yield better results than the method of choosing them arbitrarily for base prediction.

- In Chapter 4, to further explore the features of the AMI profiles calculated from DNA sequences, alternate AMI profiles calculated with respect to DNA triplets were constructed since DNA is known to operate via a triplet code. The features and anomalies of these triplet AMI profiles were analyzed and

compared amongst organisms, and special attention was made to the appearance of long tandem repeats in human DNA sequences.

- In Chapter 5, it was noted that DNA consists of regions that code for proteins known as coding regions found by "wet-lab" experimentation, the locations of which are generally well known. Various machine learning techniques such as neural networks and support vector machines were explored in combination with the AMI profile to determine whether coding regions, whose locations are already known, could be sufficiently differentiated from noncoding regions. This would indicate whether the AMI profile is helpful in differentiating between known regions.

- In Chapter 6, a method to identify differing regions of the DNA sequences on chromosomes was developed using principal component analysis based upon the AMI profile to differentiate between regions. The goal was to determine if any clustering of DNA sequences produced by the principal component analysis could be related to the information contained in the various differentiated regions.

- In Chapter 7, an arithmetic coding method was developed and applied to DNA sequences after determining the frequency of appearance of groups of bases ("k-mers") in DNA sequences. The relative frequency of the k-mers was used to test both adaptive and non-adaptive methods of arithmetic coding on several bacterial genomes to analyze whether the information about the appearance of k-mers in the DNA of an organism could be used to compress its own genetic information as well as that of other organisms.

Chapter 2

Base Prediction Using Estimation Theory

2.1 Introduction

In order to appropriately determine which sections of DNA are informationally significant, a model of its underlying information structure is required. Such a model could be constructed in a number of conceivable ways, but it is not yet known which method can produce a sufficiently accurate model. One way to test the fitness of a model for the information structure of DNA is to use it to compress and correctly recover the data contained in a DNA sequence. If compression and recovery can be performed successfully in this context, it would indicate that the model used in such a process was sufficiently accurate to describe the information structure of the DNA sequence.

To develop a compression and recovery process for DNA bases, some method of recovering DNA bases missing from a sequence that only uses the retained bases in the sequence would be needed. This chapter studies the possibility that two common estimators, the maximum likelihood (ML) estimator and the maximum *a posteriori* (MAP) estimator, could serve as accurate recovery methods for this purpose. These two estimators are based on the conditional probabilities of the surrounding bases and attempt to estimate missing bases in the DNA sequence by

using the maximum values of these associated conditional probability distributions. This chapter aims to determine whether these common estimation methods can be used to accurately predict missing bases in DNA sequences and, more broadly, whether or not these estimation methods can capture enough of the underlying information structure of DNA sequences.

The ML and MAP estimators function by determining the values of missing bases based on both the retained bases and the probabilities of bases being associated, in general, with one another. To evaluate both the ML and MAP methods of estimation for missing DNA bases, three main questions about the sufficiency of these estimators are explored. First, how many bases surrounding the missing base in question need to be included in the determination of the conditional probabilities for the predictions to be relatively accurate? In other words, to what extent will contributions from bases farther away from the missing base affect the accuracy of these estimators? Second, is it better to consider the conditional probabilities of surrounding bases individually and then aggregate the results of multiple conditional probability mass functions (PMFs) for each surrounding base, or is it better to consider the surrounding bases collectively in a single joint conditional probability? Finally, will the results match the common understanding that the MAP estimator is superior to the ML estimator? In other words, is the MAP estimator more accurate at predicting missing bases from surrounding ones as would be expected? The answers to these questions will direct the inquiry of determining the usefulness of estimation-theory methods in predicting DNA bases.

2.2 Methods

Using traditional estimation methods to estimate or predict missing bases in a DNA sequence requires some knowledge of the probability distributions that relate to each base and its position in the DNA sequence. These probabilities can be measured from the full, original DNA sequence (or a subsection thereof) by calculating the marginal probability of the appearance of each base as well as joint probabilities of bases that appear a specific number of positions away from each other, e.g., the probability that base A appears at a certain position and base T appears two positions after it. Using these probabilities calculated from the known DNA sequence, the associated discrete conditional probability distributions can then be approximated. Once these PMFs are obtained, then the ML and MAP estimators are found by simply identifying the base at which the maximum probability mass appears. Since the PMFs are based on "valueless" elements (the DNA bases A, C, G, and T have no numeric value relative to each other), selecting the base with the maximum probability in the distribution will yield the estimate.

The probabilities calculated from the known DNA sequence (before bases are removed) are obtained as follows. As with the determination of AMI in Equation (1.7), the marginal probabilities for each base are calculated by considering the total number of times a base X appears in the sequence and dividing that number by the total number of bases in the sequence. The joint probabilities for each pair of specific bases appearing a certain distance from each other are calculated according to Equation (1.9). Based on these empirical marginal and joint probability calculations, both the likelihood and posterior conditional PMFs are approximated using Bayes' Theorem according to Equation (2.1). In this equation, $p_k(X|Y)$ is the conditional probability of base X appearing given that base Y appears k positions

away, $p_k(XY)$ is the joint probability of bases X and Y appearing k positions apart, and $p(Y)$ is the marginal probability of base Y.

$$p_k(X|Y) = \frac{p_k(XY)}{p(Y)} \tag{2.1}$$

Once the conditional probabilities are calculated for each base of interest, the ML and MAP estimates can be found by finding the base with the maximum probability mass in the corresponding conditional PMF. As described in [4], the ML estimator relies on the likelihood conditional probability distribution, $P(Y|X)$, which is the probability that some measurements Y would occur given an associated state X. In contrast, the MAP estimator relies on the posterior conditional probability distribution, $P(X|Y)$, which is the probability that the state X would exist given the measurements Y occurred. For bases in a DNA sequence, the missing base is treated as the "state," which is unknown and needs to be estimated, and the surrounding known bases are treated as the "measurements" since they are known and are the basis for the estimation.

Specifically, the likelihood conditional probability used in the ML estimator is the probability of the occurrence of the surrounding bases given the unknown base. For example, in the short sequence GAG_CAT, the likelihood conditional probability is the probability that G occurs three positions away from the unknown base given that the unknown base is A. This is considered for all four possible base values (A, C, G, and T) for the unknown base to form the discrete likelihood conditional PMF. The posterior conditional probability used in the MAP estimator is the probability of the unknown base given the surrounding bases. For example, in the same short sequence GAG_CAT as before, the posterior conditional probability is the probability that the unknown base is A given that G is three positions away.

Unknown Base	ML PMF_3	MAP PMF_3
A	$p_3(G\|A)$	$p_3(A\|G)$
C	$p_3(G\|C)$	$p_3(C\|G)$
G	$p_3(G\|G)$	$p_3(G\|G)$
T	$p_3(G\|T)$	$p_3(T\|G)$

Table 2.1: Summary of the construction of the PMFs for both the ML and MAP estimators for an unknown base, the state, given that the base G, the measurement, is three positions away.

This is also considered for all four possible base values for the *unknown* base to form the discrete posterior conditional PMF. A summary of the construction of these PMFs for the ML and MAP estimators according to the example used can be seen in Table 2.1.

Since there is more than one known base surrounding any given unknown base, the ML estimate and MAP estimate can be computed in one of two ways. First, each surrounding base can be considered an individual estimator, and thus each base would produce an estimate for the unknown base from its specific, position-dependent conditional PMF. To determine the final estimate for that position, each surrounding base casts one "vote" for its prediction, and the winner with the most votes is selected as the prediction representing all the surrounding bases. This will be called the "individual" method of estimation. (As a tie-breaker in case of two estimates receiving equal votes, the base with the higher marginal probability is selected.) Second, all surrounding bases can be considered together to be one estimator. In this case, the appearance of each surrounding base is considered a probabilistically independent event, and so the conditional PMFs for each surrounding base are multiplied together to obtain a joint conditional PMF for all surrounding bases. For example, with the short GAG_CAT sequence from before, this joint conditional PMF for the ML estimator represents the probability that

GAG and CAT surrounds the unknown base given that it is A. For the MAP estimator with the same sequence, this joint conditional PMF represents the probability that the unknown base is A given that GAG and CAT surround it. The ML and MAP estimates are then determined to be the base with the maximum probability mass in their respective joint conditional PMFs. This will be called the "collective" method of estimation.

To test the accuracy of the ML and MAP estimators in both the individual and collective implementations, they were applied to several human chromosomes. Sequences of 500 bases in length, taken from each of these chromosomes, were used for each test; thus, the probabilities were calculated specifically from these 500-base-long sequences, not for the chromosome as a whole, as this can better encapsulate local information and provide for better estimates. The ML and MAP estimators were tested to predict each base in the sequence, as each base was removed one-by-one and a specific number of surrounding bases (called the "window length") was used to predict the one missing base. The window length was measured reflectively about the unknown base, so a window length of 10 means that 10 bases before and 10 bases after the unknown base (if the ends of the sequence permitted) were included in the determination of the ML and MAP estimators. Both the individual and collective implementations of the ML and MAP estimators were tested over varying window lengths, from 1 to 100 bases on either side of the unknown base. Once all bases in the sequence had been predicted, the "predicted sequence" was compared with the original, actual sequence, and an accuracy value (from 0 to 1) was determined. Finally, to obtain results that were representative of the chromosome as a whole, the accuracy values for each type of ML or MAP estimator were averaged over 25 different 500-base sequences from different locations on the chromosome.

2.3 Estimator Prediction Performance

The ML and MAP estimators were tested on DNA sequences from several chromosomes in the human genome (obtained from the GRCh38.p13 Primary Assembly in the RefSeq database of NCBI). These chromosomes were somewhat arbitrarily selected: chromosome 9 (NC_000009.12), chromosome 15 (NC_000015.10), the X chromosome (NC_000023.11), and the mitochondrial DNA sequence (NC_012920.1). An additional "randomly generated" DNA sequence was created by selecting from each base (A, C, G, or T) at random with equal probability. This sequence was thus designed to have no underlying information structure since the order of its bases was random. All ML and MAP estimators were also tested on this randomly generated sequence as a control in order to demonstrate that the ML and MAP estimators were capturing some underlying structure in the DNA data. The results on the randomly generated sequence can be seen in Figure 2.1, and the results for the real human chromosomes can be seen in Figures 2.2 through 2.5. These results show that both the ML (with the exception of the mitochondrial DNA) and the MAP estimators outperform random guessing of bases, which should result in an average accuracy of 25%.

Figure 2.1 demonstrates that, when applied to a random DNA sequence that contains no real information and no structure, both the ML and MAP estimators only attain at or slightly over a 25% accuracy, which is expected, indicating that there are probably no measurable biases or other anomalies associated with the estimation process itself. By comparison with the results from the human chromosomes, it also shows that the real DNA has underlying information structure that the ML and MAP estimators are capturing when they have average accuracy values greater than 25% for the real human chromosomes.

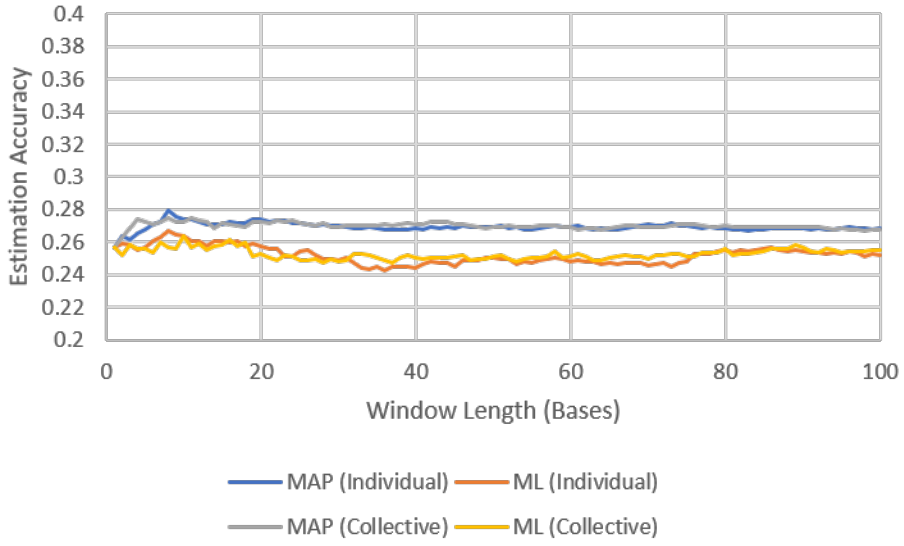

Figure 2.1: Average accuracy for the individual and collective methods of both the ML and MAP estimators over varied window lengths for randomly-generated DNA-like sequences.

The results for the chromosomes in Figures 2.2 through 2.5 show similar trends for both estimators. It can be seen that a window length of about 20-30 bases on either side of the unknown base is sufficient to obtain the highest accuracy possible for each estimator. For the ML estimator, regardless of the chromosome studied, the collective method, where the maximum of the joint conditional probability of all bases in the window was used, performed better than the individual method by about 1-4% accuracy. For the MAP estimator, however, the results for the individual and collective methods were almost identical, and neither produced more accurate results on any chromosome tested. It can be speculated that this is due to the fact that, with the MAP estimator, the bases surrounding the unknown base are treated as given in its definition of the conditional probability; thus, the estimation of the unknown base varies little for either the individual or collective methods. Considered another way, the fact that the surrounding bases are already known when using the MAP estimator precludes the mechanism by which the maximum

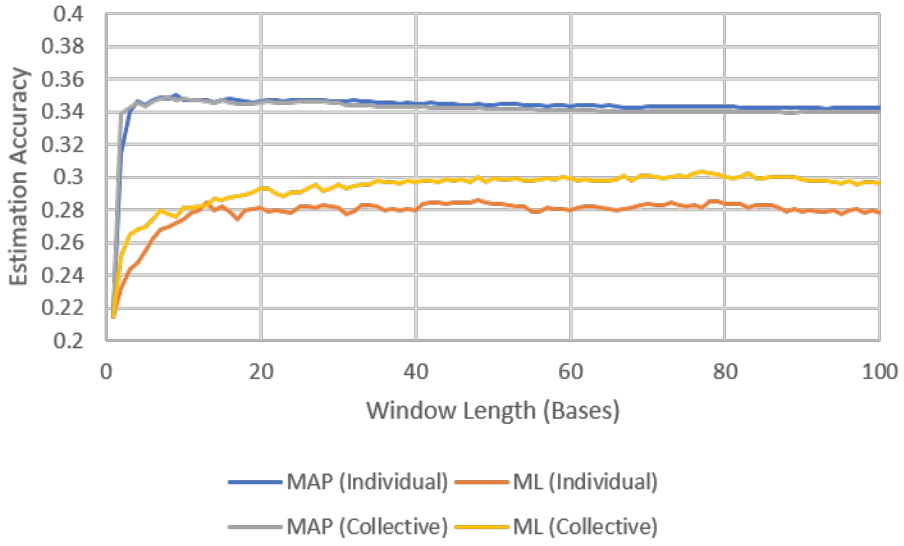

Figure 2.2: Average accuracy for the individual and collective methods of both the ML and MAP estimators over varied window lengths for human chromosome 9.

probability and thus the estimate is calculated from varying widely since that mechanism uses the surrounding bases to make its determination. For the ML estimator, however, since the surrounding bases are probabilistically considered based on their conditional probability given the unknown base, the method for calculating the maximum conditional probability can greatly affect the prediction of the unknown base. In other words, the collective method intuitively makes more sense when considering the probability that the surrounding bases, taken together, appear given the unknown base rather than considering how each individual surrounding base would appear by itself, to the exclusion of the others, given the unknown base.

In all cases studied, the MAP estimator always outperformed the ML estimator, which was expected. The MAP estimator achieved greater than 32% accuracy on all tests, and its greatest performance was on the X chromosome, which averaged about a 36.5% accuracy. In comparison, the ML estimator only reached a maximum of

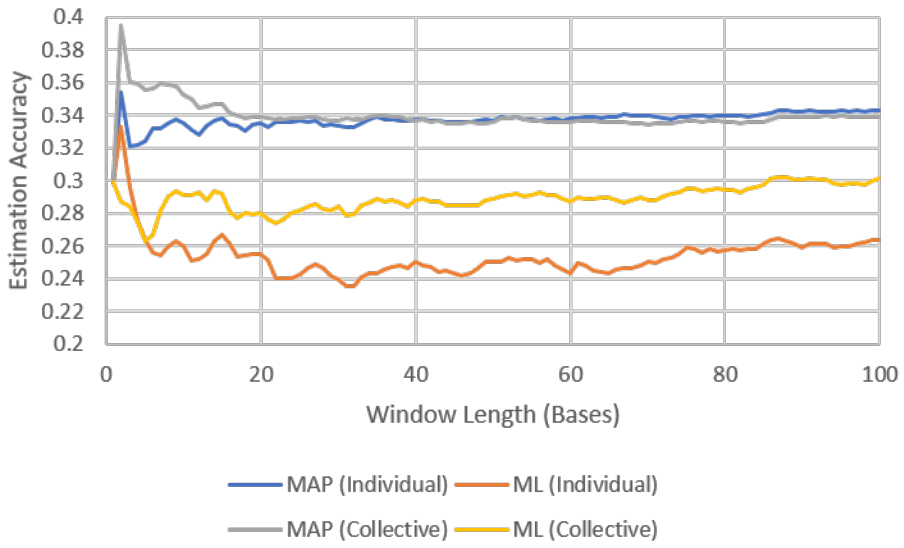

Figure 2.3: Average accuracy for the individual and collective methods of both the ML and MAP estimators over varied window lengths for human chromosome 15.

32% for the X chromosome, and sometimes averaged no better than random guessing at 25% accuracy. Therefore, since the highest observed accuracy of the ML estimator was the lowest accuracy observed for the MAP estimator, the MAP estimator can be considered superior to the ML estimator in this application as in many others. However, accuracies of no greater than 36.5% on average (with some individual, unaveraged test results reaching as high as 48%) for the MAP estimator do not provide accurate enough recovery for a compression and recovery scheme to be successful by itself using only the MAP estimator. It can be said that while the results of both the ML and MAP estimators are insufficient for full compression and recovery, their best accuracy can serve as a threshold for more sophisticated methods of prediction to be measured against if they are to be deemed more successful than simple estimators.

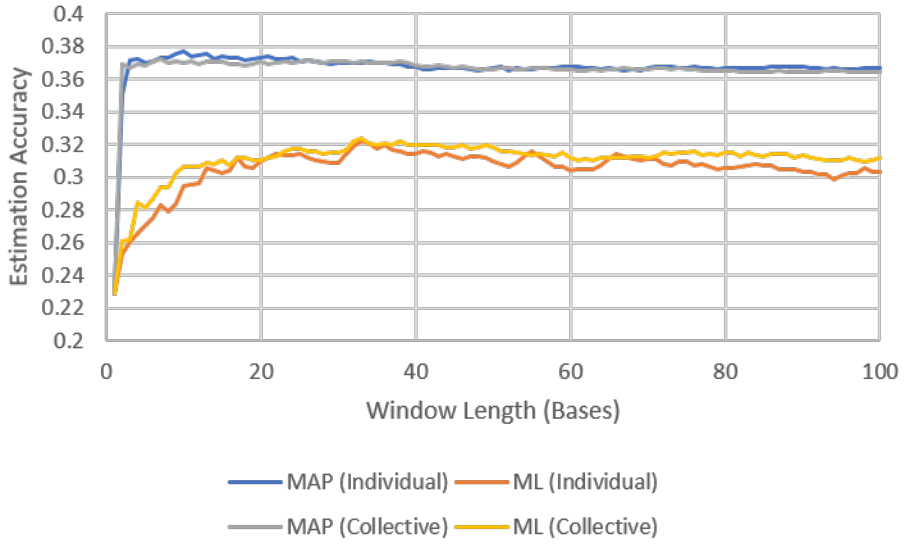

Figure 2.4: Average accuracy for the individual and collective methods of both the ML and MAP estimators over varied window lengths for the human X chromosome.

2.4 Future Work

Some future work in this area could potentially include incorporating the MAP estimator method with other approaches such as those discussed later which use the AMI profile in order to achieve improved accuracy. More future work could entail a deeper study of some of the actual predictions and the response of these estimator methods to high marginal probability, as some results produced predictions of the same base for the entire sequence if the marginal probability of one base was exceedingly high for a part of the sequence. There is also a chance that these estimation methods could inform the first part of the information-structure model, the choice of bases to keep in a compression set, rather than just affecting the recovery phase, which could lead to further study. Finally, conditional probabilities that take into account the order of the bases could be studied in addition to the directionless ones addressed in this study. In other words, this chapter only examined the probability that specific bases occurred a certain distance apart, e.g.,

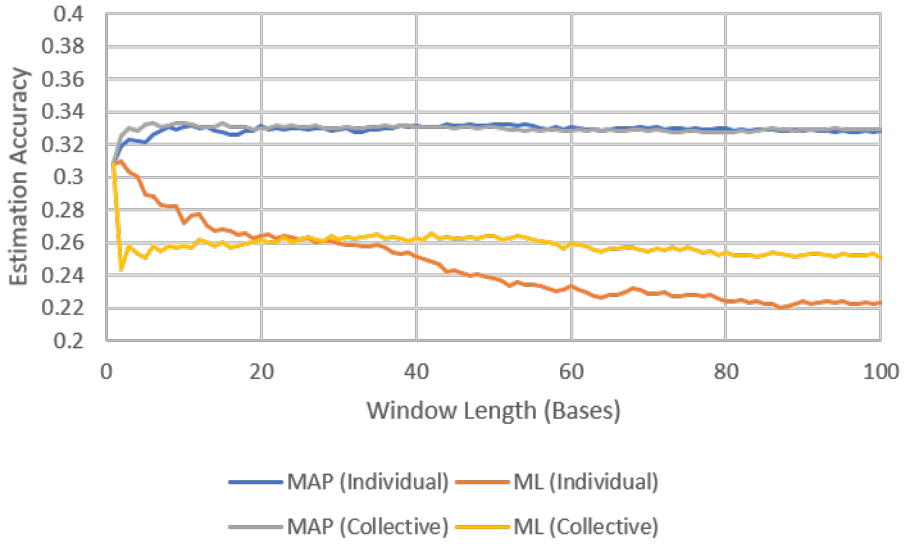

Figure 2.5: Average accuracy for the individual and collective methods of both the ML and MAP estimators over varied window lengths for human mitochondrial DNA.

whether an A was three spaces away from a G, but it did not examine whether the additional information about where bases occur affected the results, e.g., whether an A appeared either three before a G or three after a G.

2.5 Conclusion

The purpose of this study of ML and MAP estimators was to determine if these common methods of estimation could predict missing DNA bases accurately such that they could form the basis of a model for compression of DNA sequences. If an accurate model for compression and recovery could be found, then knowledge of the information structure of DNA could be obtained as a byproduct. It was found that both the ML and MAP estimators performed better than random, meaning they captured some probability-based information structure in the DNA sequences, and it was also found that the MAP estimator always outperformed the ML estimator. However, although the ML and MAP estimators tested do expose some of the

information structure within the DNA sequences, their accuracy was not sufficient to be used as the sole recovery method for missing DNA bases. At minimum, since ML and MAP estimators represent very basic models of prediction, these results could be used as a baseline to evaluate the results of other, more complex models in justifying their increased complexity.

Chapter 3

Base Prediction Using the AMI Profile

3.1 Introduction

Inspired by a compression method for images called "inpainting" developed by Galić, Weickert, et al. and described in [5], which was able to use partial differential equations to predict regions of missing pixels in an image based on retained pixels, a prediction mechanism for missing DNA bases was developed that used the AMI profile in place of partial differential equations. This method was tested to determine whether the AMI profile of a DNA sequence alone could predict missing bases in that same sequence based on certain bases which were retained. If this could be done, then DNA could be compressed and recovered accurately. In order to perform recoverable compression of anything, its underlying structure has to be understood. If the information structure of DNA was being quantified by the AMI profile in a manner suitable for compression, then that knowledge of the structure could possibly be exploited to discover the locations of informationally significant regions of the DNA sequence.

When conceptualizing this approach to predict missing bases of a DNA sequence based on strategically retained bases, the problem of selecting which bases to retain and which to discard is evident. The methods employed by [5] were specifically

tailored to the problem of image compression, such as retaining pixels in regions around definite features and points of high contrast within the image and discarding pixels in regions without much variation. In the realm of DNA, however, the ideal method for selecting which bases to retain and which bases to discard was not inherently evident. Thus, two methods were attempted: one method simply selected groups of bases to retain and to leave out arbitrarily. The other method attempted to quantify which bases would be the best at predicting other bases in order to retain the best predictors and discard the worst predictors.

3.2 *Ad Hoc* Prediction Method

Without any method readily suggested by the AMI itself for selecting bases to discard and bases to retain, an "*ad hoc*" method was employed which arbitrarily selected bases to be retained and bases to be discarded based on regular increments of bases. These two increments of bases were referred to as "groups" and "gaps," which consisted of a certain number of bases in each that repeated regularly for the length of the DNA sequence in question. These two arbitrary parameters, the group length and the gap length (in terms of the number of bases in each), were fixed at the beginning of a prediction test. The group length determined how many bases in a row would be retained on the DNA sequence before a gap was encountered. Consequently, the gap length determined how many bases in a row would be left out on the DNA sequence between two groups which were retained. In the end, this resulted in a DNA sequence with alternating groups and gaps of fixed lengths sequentially alternating as one moves along the DNA sequence. All the bases in the group regions were retained and thus became part of the compression set. An algorithm was then employed to predict the values of the bases in the gap regions

from the bases remaining in the group regions using the joint probabilities and the AMI profile.

3.2.1 Process and Methods

In order to perform base prediction testing, first a DNA sequence was selected for analysis. The probability of each base's occurrence in the sequence was determined for each base according to Equation (1.7). Using this, the joint probabilities of the occurrence of each base type with respect to every other base type k positions away were determined according to Equation (1.9). Finally, using these probabilities, the AMI profile was calculated for this sequence according to Equation (1.11). In performing these calculations, all bases in the sequence were known. (In other words, no bases had yet been excluded for the purposes of prediction.)

Once the AMI profile had been determined, the group length (X) and gap length (Y) were fixed. The retained bases on the DNA sequence would be constructed as follows: starting at the beginning of the sequence, X number of bases (a group) would be retained, and then the next Y bases (a gap) would be omitted. Then another X would be retained, another Y omitted, and so on for the whole sequence. The retained bases formed the set upon which recovery of the missing bases would be attempted using the AMI profile and the joint probabilities between all bases.

Several methods for determining the missing bases in each gap were tested. To begin, each empty base position in the gap would be considered individually. For a particular empty base position, both the bases in the group immediately *before* the base position in question and the bases in the group immediately *after* the base position in question would be used to determine the missing base. The joint probabilities of each base with respect to another base k positions away were used to make guesses about what the base in the gap position in question should be.

Every base in both the precedent and subsequent groups made a guess based on its own joint probability with the empty base position in question by distance k (which would be different for each base in each group). Finally, once each base in both groups had made its "guess," then these guesses would be analyzed to determine the final prediction for that base position. If the joint probabilities of two bases with respect to the base in question k positions away were equal, the marginal probabilities of each single base would be used as a tie breaker. (In other words, the base with the highest appearance in the sequence was chosen as the prediction.)

For example, consider the sequence fragment GAG_ _ _ _ _CAT, which contains a group length of three bases and a gap length of five bases. In this whole sequence, there are three bases retained before a gap and three bases retained after it. To begin making predictions, the first empty base position in the gap of five bases would be considered. The first base in the preceding group would make a guess at the empty base. Since the group in this example is size 3 and the first base position in the gap is being considered, the joint probability of the base G with all other base types three positions away would be considered. The base out of the set A, C, G, T which had the highest joint probability with base G at three positions away ($k = 3$) would be chosen as the "guess" from base G. Next, the base A from the preceding group would be considered. The joint probability of base A with a base that is now two positions away from the base position in question would be used to make its guess. Thus, the base out of the set A, C, G, T which had the highest joint probability with base A at two positions away ($k = 2$) would be chosen as the "guess" from base A. The same process would be followed for the second base G in the preceding group with $k = 1$.

A similar backward-looking process would then be followed for each base in the succeeding group for the same base position in question (the first position in the gap

in this example). First, the joint probabilities of base C with another base that is 5 positions away ($k = 5$) would be considered, and the base out of the set {A, C, G, T} which had the highest joint probability with base C at 5 positions away would be chosen as its "guess" for the empty base position in question. The same process would be followed for bases A and T in the succeeding group.

Once all of the bases in the groups both before and after the base position in question have made their guesses, these guesses are then compared and a single prediction for that base position is offered. The contribution of each bases' guess is weighted by the AMI profile value for how far away that base is from the empty base position in question. Two prediction modes were tested: (1) The "weighted vote" prediction mode had each base in the surrounding set of retained bases "cast a vote" for the base that it guessed, with each "vote" weighted by the AMI profile for the position of the base relative to the empty base position in question. The base receiving the highest score as a sum of the weighted votes was made the prediction for that empty base position. (2) The "highest weight" prediction mode simply made the guess of the base with the highest AMI weight the prediction for the empty base position in question.

The *ad hoc* prediction method would use this process to predict a base for every empty base position on the sequence. Once all bases were predicted, each empty base position was compared to the known bases from those positions, and an accuracy value was determined. Many tests were performed combining various values for the group and gap lengths.

3.2.2 Prediction Results

This *ad hoc* method yielded mixed results when tested on various DNA sequences. Assuming that guessing bases randomly would be met with a 25% success rate, the

ad hoc method performed better than random guessing, meaning that it was detecting "something" in the data, but the accuracy of the predictions rarely exceeded 50% for any sequence no matter what combination of group and gap values were tested. Prediction accuracy rates of 30-35% were very common. As a control, is was observed that, if instead of using an actual human DNA sequence, a randomly generated string of the letters A, C, G, and T was used, the prediction levels never exceeded 25-27%. Thus, there was some structure within the information being exploited by the prediction methods tested, because accuracy levels were above that for trying to predict a sequence with no structure. However, not enough information was obtained to determine exactly what that structure was, how to improve it, or how to use it effectively and reliably (in other words, employ this method of prediction method repeatedly with confidence in the results).

To analyze the accuracy of these prediction methods, human chromosome 9 (NC_000009.12 *Homo sapiens*, GRch38.p13) and chromosome 15 (NC_000015.10 *Homo sapiens*, GRCh38.p13) were arbitrarily chosen to be analyzed. On each chromosome, 20 different sequences consisting of 1,000 bases each were chosen, and the accuracy results obtained from predicting bases on each of these sequences were averaged together to get a final average accuracy. For chromosome 9, in order to avoid the telomere region where bases in the human genome assembly are often undetermined, the starting offset was set to 1,000,000 bases from the start of the chromosome. Each 1,000-base sequence was then selected at 1,000,000-base intervals, thus selecting sequences from position 1,000,000, 2,000,000, 3,000,000, etc. For chromosome 15, since close to 17,000,000 bases at the start of the chromosome are currently undetermined, the starting position for the selection of its 1,000-base regions was at position 18,000,000; starting there, regions were selected similarly to those for chromosome 9.

The results on chromosome 9 when using prediction mode (1), where each vote is weighted by its AMI profile value, are given in Figure 3.1. The highest average accuracy for any region on chromosome 9 using this base-prediction method was 33% for a group length of 6 and a gap length of 1. This is almost a trivial case, since 6 bases are retained for every one base discarded, which makes the achievable compression very low, even if close to 100% accuracy could be obtained. Intuitively, it can be seen that slightly greater accuracies result when there are fewer gap positions and more bases retained. Beyond that, however, it turns out that the group and gap length combination has an almost negligible effect on the average accuracy that can be obtained.

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.310479	0.316168	0.311487	0.314303	0.313234	0.315559	0.307256	0.317634	0.312046	0.314066	0.313041	0.316883	0.314316	0.311994	0.308942
2	0.324401	0.314044	0.312106	0.311078	0.313287	0.313558	0.316645	0.317636	0.3116	0.312738	0.3134	0.311343	0.310161	0.312075	0.313785
3	0.315737	0.30908	0.314671	0.317483	0.315873	0.318899	0.318388	0.31353	0.314418	0.317143	0.315341	0.318346	0.316422	0.317797	0.309821
4	0.304229	0.321407	0.319231	0.317063	0.320714	0.316172	0.315463	0.319271	0.317893	0.316736	0.317843	0.31918	0.31545	0.311033	0.316101
5	0.323054	0.320455	0.320899	0.321652	0.321386	0.314286	0.317347	0.317857	0.318981	0.317015	0.318182	0.316737	0.309409	0.314016	0.316667
6	0.330769	0.317063	0.327679	0.32203	0.312967	0.320139	0.320315	0.32066	0.31932	0.312381	0.316564	0.309747	0.318868	0.317787	0.317014
7	0.310317	0.322321	0.314356	0.316346	0.321548	0.322619	0.316567	0.312966	0.312434	0.317288	0.311282	0.317217	0.315008	0.316071	0.316667
8	0.323214	0.317822	0.32033	0.319643	0.320909	0.316088	0.312473	0.315972	0.318267	0.309821	0.316724	0.316912	0.315385	0.315916	0.316439
9	0.313861	0.317857	0.32123	0.321916	0.31875	0.314925	0.315986	0.320127	0.310913	0.318396	0.317558	0.31684	0.31689	0.317776	0.311349
10	0.326374	0.325	0.316017	0.319965	0.317761	0.32037	0.319976	0.313728	0.318029	0.320882	0.317708	0.319022	0.318444	0.311565	0.312683
11	0.32381	0.320455	0.31713	0.316418	0.316825	0.31709	0.31824	0.317453	0.318627	0.318021	0.318379	0.316951	0.312454	0.313937	0.308291
12	0.303247	0.316319	0.31194	0.32004	0.319831	0.318304	0.313073	0.315074	0.319444	0.3175	0.319938	0.312103	0.313039	0.305678	0.317895
13	0.326389	0.31194	0.32381	0.316102	0.319821	0.315094	0.315266	0.320573	0.316787	0.319318	0.313636	0.311382	0.306016	0.318797	0.315093
14	0.314925	0.322222	0.320339	0.317187	0.31	0.314706	0.31756	0.315897	0.315909	0.314643	0.310754	0.308868	0.316194	0.315476	0.311524
15	0.324603	0.319492	0.31875	0.304009	0.318431	0.319618	0.315373	0.319176	0.314815	0.310122	0.305944	0.317982	0.317201	0.311633	0.313529

Figure 3.1: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 9 using the weighted-voting prediction mode.

The results on chromosome 9 when using prediction mode (2), where only the guess with the highest AMI-profile value is chosen as the prediction, are given in Figure 3.2. These outcomes are extremely similar to those for prediction mode (1) with only a slightly lower maximum average accuracy of 32.6% for all regions tested. Once again, the intuitive result that regions with fewer gaps and more bases retained have higher prediction accuracies. However, no substantial improvement or deterioration in accuracy is observed depending on which prediction mode was

employed. These accuracy values for this prediction algorithm, while performing better than random guessing, which should be expected to achieve a 25% accuracy, fare hardly better than the results for the best estimators studied in Chapter 2.

Group Length (Bases)	Gap Length (Bases)														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.316467	0.316168	0.314077	0.314303	0.313772	0.315559	0.308957	0.317634	0.312156	0.314066	0.312771	0.316883	0.314316	0.311994	0.309471
2	0.312725	0.313247	0.311028	0.310853	0.313497	0.312037	0.317857	0.315903	0.312149	0.310536	0.314876	0.313194	0.313375	0.30924	0.311695
3	0.315339	0.30597	0.315369	0.316259	0.311508	0.316964	0.315842	0.311538	0.311177	0.315455	0.314457	0.314366	0.310317	0.31362	0.307798
4	0.302985	0.316766	0.31352	0.315079	0.3175	0.314191	0.312009	0.313393	0.314791	0.314028	0.313161	0.312831	0.312321	0.310395	0.313522
5	0.318563	0.320629	0.314286	0.315067	0.31802	0.309707	0.315476	0.317208	0.317978	0.314179	0.313997	0.314831	0.310234	0.31314	0.310588
6	0.325524	0.307143	0.320685	0.320297	0.306154	0.316071	0.317347	0.318837	0.315423	0.311429	0.315871	0.307292	0.312119	0.312885	0.311667
7	0.306746	0.314955	0.312376	0.313187	0.314405	0.320022	0.31875	0.316884	0.310935	0.315339	0.306494	0.313836	0.312066	0.312054	0.306957
8	0.314732	0.311634	0.308608	0.310268	0.317013	0.317014	0.310448	0.313393	0.315913	0.306429	0.314237	0.311928	0.309936	0.309472	0.310909
9	0.308911	0.301374	0.315278	0.317532	0.313472	0.307338	0.308844	0.316525	0.311409	0.314528	0.312032	0.311806	0.31112	0.306981	0.301508
10	0.314835	0.320536	0.316667	0.320486	0.314925	0.313624	0.311743	0.312835	0.311216	0.315294	0.311837	0.311685	0.307605	0.306037	0.303252
11	0.323214	0.310065	0.316435	0.318843	0.312063	0.314266	0.3125	0.309552	0.313943	0.313333	0.308202	0.307292	0.304762	0.305836	0.306325
12	0.298701	0.314931	0.31791	0.319246	0.315424	0.312946	0.309299	0.31152	0.314931	0.308696	0.307955	0.307738	0.308443	0.305128	0.314474
13	0.304861	0.316045	0.321693	0.312288	0.309821	0.311006	0.314426	0.315365	0.306522	0.30875	0.308225	0.311789	0.306607	0.315508	0.305
14	0.320896	0.317857	0.310452	0.309821	0.309057	0.312418	0.314881	0.303261	0.31452	0.305119	0.307761	0.309402	0.315081	0.307639	0.306476
15	0.320635	0.309322	0.319345	0.308726	0.317843	0.321181	0.301708	0.312358	0.302778	0.307561	0.308741	0.315241	0.309722	0.308061	0.310098

Figure 3.2: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 9 using the highest-weight prediction mode.

It can be noted, however, that these accuracies are averages over several regions on chromosome 9. If an optimistic lens is used, the maximum accuracy for any trial in any particular region can be noted to see if the performance of any trial shows promise as a potential compression technique. The maximum accuracy for any trial conducted in any particular region for each group- and gap-length combination can be seen in Figure 3.3. (Since the weighted-voting prediction mode had slightly better accuracy, only its results are shown.) It can be seen that, while these accuracy values are generally higher, representing somewhere between 40% and 50% accuracy, a 50.5% accuracy is still the *highest* accuracy that could be obtained for this *ad hoc* prediction method *anywhere* in the trials on human chromosome 9, which is not significant enough to form the basis of a compression technique.

Trials on human chromosome 15 displayed highly similar results to those of chromosome 9. The average accuracies for all trials performed on chromosome 15 using prediction mode (1) can be seen in Figure 3.4, while the average accuracies

MAX	Gap Length (Bases)														
Group Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.407186	0.399701	0.410359	0.390547	0.392814	0.399767	0.409297	0.399554	0.391639	0.398901	0.401515	0.399351	0.400641	0.394456	0.405291
2	0.398204	0.420319	0.38806	0.390719	0.408392	0.417989	0.397959	0.397277	0.399267	0.396429	0.403778	0.405093	0.398393	0.408163	0.40226
3	0.418327	0.400498	0.391218	0.396853	0.41746	0.415179	0.403112	0.402473	0.395503	0.403896	0.39899	0.409204	0.412698	0.414044	0.404762
4	0.40796	0.404192	0.393939	0.430556	0.414286	0.414191	0.411303	0.395833	0.399711	0.404167	0.412483	0.410053	0.430248	0.408163	0.411321
5	0.407186	0.398601	0.441799	0.40625	0.421782	0.413919	0.409864	0.405844	0.395062	0.422388	0.402597	0.423729	0.40522	0.413747	0.419608
6	0.433566	0.452381	0.410714	0.428218	0.408791	0.412698	0.41744	0.407986	0.427861	0.403175	0.42527	0.395833	0.415094	0.415966	0.409722
7	0.412698	0.392857	0.419142	0.425824	0.409524	0.411255	0.39881	0.408582	0.402116	0.433898	0.397727	0.416667	0.422323	0.418155	0.427536
8	0.419643	0.440594	0.446886	0.419643	0.418182	0.398148	0.422175	0.414683	0.425612	0.398214	0.418525	0.433007	0.413462	0.427019	0.410606
9	0.445545	0.461538	0.420635	0.435065	0.413889	0.435323	0.426304	0.423729	0.39881	0.422642	0.43672	0.414931	0.419732	0.423701	0.436508
10	0.505495	0.428571	0.411255	0.40625	0.432836	0.417989	0.421308	0.404018	0.421384	0.433333	0.414773	0.429348	0.412587	0.430272	0.413008
11	0.440476	0.428571	0.398148	0.429104	0.396825	0.420904	0.40051	0.408019	0.43573	0.420833	0.442688	0.410985	0.432234	0.41115	0.411966
12	0.415584	0.430556	0.41791	0.412698	0.416949	0.416667	0.412399	0.438725	0.423611	0.45	0.411157	0.434524	0.412758	0.410256	0.419298
13	0.472222	0.395522	0.433862	0.419492	0.403571	0.415094	0.420168	0.421875	0.458937	0.418182	0.430736	0.408537	0.416174	0.411654	0.438889
14	0.447761	0.436508	0.429379	0.415179	0.422642	0.421569	0.428571	0.467391	0.411616	0.433333	0.40133	0.418803	0.421053	0.430556	0.420952
15	0.396825	0.440678	0.416667	0.424528	0.431373	0.423611	0.475155	0.420455	0.449735	0.395122	0.426573	0.429825	0.425214	0.420408	0.421569

Figure 3.3: Heat map showing the maximum accuracy obtained predicting missing bases with varying group and gap lengths for any of the 20 regions on human chromosome 9 using the weighted-voting prediction mode.

using prediction mode (2) are shown in Figure 3.5. Once again, the average accuracies all fall within a close range of about 28-33% regardless of the group and gap lengths employed. However, for chromosome 15, prediction mode (2) performed slightly better than prediction mode (1), just about reversing the results obtained from chromosome 9. As can be seen in Figure 3.6, the maximum accuracies observed for any trial on any region of chromosome 15 were still almost always below 50%, as observed for chromosome 9, with the exception of some values that exceeded 50%, the maximum being 56.9% for a group length of 13 and a gap length of 1. Again, it can be reiterated that this "best" result is an edge case—a trivial situation in which only one base can be discarded for every 13 bases retained, which simply will not form a feasible foundation for any type of consequential compression technique.

3.2.3 Prediction Results with Single-Base AMI Profiles

Due to the relatively low accuracy results obtained for the *ad hoc* base-prediction method explored so far, it was assumed that more knowledge of the information structure of a DNA sequence could yield higher prediction accuracies. The only active components of the prediction algorithm quantifying the relatedness of the

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.287126	0.299251	0.29595	0.299565	0.290539	0.294231	0.292347	0.293862	0.291419	0.293516	0.290097	0.289394	0.290972	0.290032	0.294286
2	0.312275	0.308167	0.297098	0.297754	0.297063	0.297156	0.298023	0.28979	0.299451	0.296905	0.292857	0.295023	0.292021	0.293934	0.294746
3	0.315737	0.302488	0.29501	0.300175	0.299048	0.29747	0.292999	0.298008	0.299868	0.292208	0.295013	0.29347	0.295604	0.294492	0.296607
4	0.303483	0.297904	0.29965	0.300794	0.298482	0.297772	0.298509	0.297247	0.29329	0.297083	0.29654	0.294643	0.293546	0.29662	0.290692
5	0.293413	0.311538	0.298545	0.301339	0.297822	0.297436	0.294728	0.292776	0.29838	0.295299	0.292641	0.296257	0.299451	0.29407	0.293268
6	0.308741	0.300992	0.304167	0.296658	0.296703	0.299107	0.296568	0.296181	0.29602	0.292937	0.296995	0.300372	0.294702	0.295308	0.293958
7	0.316667	0.310491	0.295215	0.304808	0.301786	0.297078	0.294444	0.296175	0.291711	0.297373	0.301299	0.295047	0.2954	0.29561	0.30087
8	0.308036	0.30198	0.304396	0.292708	0.295844	0.296644	0.297548	0.291567	0.295386	0.297857	0.298027	0.294444	0.296154	0.301553	0.295
9	0.302475	0.310165	0.294841	0.298864	0.299722	0.298756	0.292404	0.298623	0.296528	0.301509	0.294831	0.297135	0.302508	0.300406	0.296111
10	0.325824	0.294345	0.301515	0.299826	0.300597	0.290873	0.296489	0.294531	0.30304	0.296961	0.297727	0.30317	0.301049	0.296088	0.298455
11	0.292262	0.310065	0.3	0.298321	0.290952	0.297599	0.290689	0.301297	0.296732	0.298125	0.302372	0.300568	0.297436	0.298868	0.300684
12	0.308442	0.303125	0.299005	0.294246	0.300339	0.291815	0.302695	0.297672	0.300231	0.3025	0.29876	0.299802	0.295685	0.299084	0.295877
13	0.296528	0.300373	0.295238	0.301695	0.292857	0.301572	0.295098	0.297396	0.306522	0.298636	0.295563	0.297154	0.302268	0.295395	0.296944
14	0.28806	0.30119	0.300282	0.292411	0.301132	0.303922	0.295387	0.305707	0.298106	0.293333	0.296452	0.301709	0.299393	0.297718	0.298952
15	0.300794	0.304661	0.296131	0.307311	0.303922	0.29566	0.302329	0.299858	0.294312	0.298537	0.302681	0.29989	0.297756	0.298673	0.291863

Figure 3.4: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 15 using the weighted-voting prediction mode.

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.30519	0.299251	0.300465	0.299565	0.290898	0.294231	0.292744	0.293862	0.291584	0.293516	0.290855	0.289394	0.290545	0.290032	0.294709
2	0.308383	0.306175	0.293698	0.296332	0.28958	0.29623	0.293495	0.285087	0.296093	0.292381	0.288843	0.288947	0.290758	0.295635	0.289379
3	0.319323	0.296269	0.298703	0.296503	0.297143	0.291295	0.28826	0.29217	0.29246	0.28526	0.287311	0.293595	0.293407	0.289407	0.293393
4	0.304726	0.292814	0.300583	0.299405	0.291071	0.293152	0.290659	0.287723	0.286724	0.290417	0.293962	0.28955	0.289961	0.289413	0.290818
5	0.291317	0.307517	0.298545	0.297879	0.289307	0.29185	0.285629	0.289042	0.291512	0.291269	0.287085	0.291314	0.290179	0.288814	0.289608
6	0.306643	0.302778	0.30506	0.296163	0.296264	0.289782	0.288683	0.294271	0.291128	0.286429	0.290524	0.290625	0.290203	0.289776	0.292986
7	0.317857	0.3	0.295875	0.296429	0.295	0.291017	0.295139	0.289552	0.285979	0.288644	0.287662	0.291745	0.289291	0.293824	0.289565
8	0.311161	0.296535	0.30348	0.297321	0.295584	0.295023	0.289872	0.284127	0.289736	0.289286	0.292882	0.290441	0.294792	0.294953	0.289545
9	0.30297	0.314835	0.298214	0.300974	0.295417	0.294154	0.279365	0.290148	0.2875	0.295755	0.292246	0.299219	0.295987	0.290341	0.29246
10	0.336264	0.3	0.305411	0.301563	0.296418	0.281217	0.290678	0.28404	0.289727	0.294804	0.296023	0.297101	0.292745	0.293452	0.290732
11	0.301786	0.307143	0.296528	0.300746	0.280476	0.292373	0.28125	0.287382	0.291721	0.29625	0.293182	0.291477	0.2913	0.289721	0.289915
12	0.297403	0.302431	0.296517	0.285317	0.298475	0.285863	0.288005	0.289216	0.292824	0.294348	0.290806	0.291468	0.289024	0.289103	0.290877
13	0.305556	0.303358	0.286243	0.298093	0.28375	0.288679	0.286415	0.294531	0.298913	0.29375	0.28961	0.287805	0.287179	0.291635	0.288519
14	0.3	0.3	0.303955	0.28817	0.298113	0.294771	0.295536	0.296739	0.288763	0.289643	0.28847	0.291774	0.293522	0.291171	0.290952
15	0.312698	0.304661	0.293155	0.299057	0.294314	0.290451	0.295342	0.289773	0.289286	0.288171	0.290559	0.295285	0.291026	0.290102	0.284804

Figure 3.5: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 15 using the highest-weight prediction mode.

information contained in each sequence were the joint probabilities between bases and the AMI profile. It was postulated that, if the AMI profile could contain a more *granular* description of the information content of the DNA sequence, it might be better at guessing which bases should be predicted.

The AMI profile, as defined in Equation (1.11), is an *average* measure which quantifies the mutual information between two base *positions* only—*without regard to the particular bases involved in this relationship*. Thus, the AMI profile doesn't contain any description of the relatedness of particular bases to other particular bases. Rather, it only contains a general description of how likely one base is to be

MAX	Gap Length (Bases)														
Group Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.441118	0.447605	0.416999	0.405473	0.372455	0.397436	0.390023	0.409598	0.40484	0.414286	0.409091	0.415584	0.405983	0.424307	0.431746
2	0.437126	0.432271	0.378109	0.411677	0.406993	0.431217	0.422194	0.424505	0.444444	0.441667	0.430933	0.4375	0.438576	0.447846	0.454237
3	0.434263	0.430348	0.389222	0.421329	0.426984	0.412202	0.414427	0.43956	0.436508	0.435065	0.435606	0.43408	0.442002	0.455206	0.435714
4	0.427861	0.437126	0.459207	0.448413	0.419643	0.435644	0.480377	0.440476	0.457431	0.465278	0.442334	0.449735	0.46545	0.442602	0.459119
5	0.419162	0.465035	0.433862	0.435268	0.447525	0.490842	0.452381	0.462662	0.459877	0.449254	0.443001	0.473164	0.449176	0.463612	0.449673
6	0.461538	0.440476	0.446429	0.45297	0.483516	0.452381	0.476809	0.465278	0.452736	0.444444	0.473035	0.459821	0.461538	0.44958	0.448611
7	0.484127	0.459821	0.462046	0.502747	0.454762	0.474026	0.472222	0.458955	0.451499	0.472881	0.461039	0.462264	0.45098	0.443452	0.475362
8	0.535714	0.465347	0.498168	0.452381	0.480519	0.467593	0.45629	0.454365	0.478343	0.453571	0.456261	0.447712	0.442308	0.473602	0.462121
9	0.49505	0.5	0.452381	0.477273	0.477778	0.465174	0.46712	0.485169	0.458333	0.45283	0.445633	0.449653	0.473244	0.469156	0.471429
10	0.483516	0.422619	0.493506	0.486111	0.471642	0.460317	0.472155	0.450893	0.454927	0.437255	0.445076	0.458333	0.47028	0.467687	0.469919
11	0.440476	0.512987	0.513889	0.477612	0.466667	0.460452	0.44898	0.45283	0.431373	0.445833	0.454545	0.462121	0.467033	0.465157	0.473504
12	0.467532	0.520833	0.472637	0.456349	0.481356	0.443452	0.447439	0.438725	0.435185	0.454348	0.460744	0.472222	0.469043	0.467033	0.449123
13	0.569444	0.5	0.465608	0.470339	0.439286	0.45283	0.431373	0.429688	0.468599	0.463636	0.463203	0.45935	0.473373	0.428571	0.442593
14	0.462687	0.507937	0.491525	0.4375	0.456604	0.444444	0.4375	0.464674	0.449495	0.459524	0.450111	0.474359	0.431174	0.428571	0.449524
15	0.492063	0.533898	0.452381	0.433962	0.45098	0.475694	0.456522	0.477273	0.468254	0.460976	0.473193	0.429825	0.42735	0.45102	0.452941

Figure 3.6: Heat map showing the maximum accuracy obtained predicting missing bases with varying group and gap lengths for any of the 20 regions on human chromosome 15 using the weighted-voting prediction mode.

related to any other base k base positions away. However, another type of "single-base" AMI profile can be defined which does take into account the mutual information shared between particular bases. This is accomplished by simply removing one of the averages from Equation (1.11). Thus, a single-base AMI profile is defined in Equation (3.1) for any particular base Z.

$$ {}_{Z}AMI_k = \sum_{Y \in \mathcal{B}} p_k(ZY) i_{Mk}(ZY) = \sum_{Y \in \mathcal{B}} p_k(ZY) \log_2 \frac{p_k(ZY)}{p(Z)p(Y)} \tag{3.1} $$

Thus, using single-base AMI profiles, there are now four different AMI profiles that could be used for any base-position separation k, one profile for each of A, C, G, and T. For the purposes of the prediction algorithm, the particular base of these single-base AMI profiles would represent the *known* base in the relationship that was being ascertained. It was hoped that single-base AMI profiles would provide a way to describe more specific elements of the information structure of the DNA sequences, leading to higher accuracy values. This was, however, not the case.

Identical trials to those performed previously were performed on human chromosomes 9 and 15 with the exception that single-base AMI profiles instead of

the standard AMI profile were used in making final base predictions. The resulting average accuracies for predictions using single-base AMI profiles can be seen in Figure 3.7 for prediction mode (1) and in Figure 3.8 for prediction mode (2). Similarly, the same results for chromosome 15 can be seen in Figure 3.9 and Figure 3.10. Once again, average accuracies within the range of 26-32% were common. The highest average accuracies for chromosome 9 were 31.8% and 30.5% for prediction modes (1) and (2), respectively. For chromosome 15, the highest average accuracies reported were 32.5% and 30.1% for prediction modes (1) and (2), respectively.

This showed, similar to the results using the full AMI profiles, that prediction mode (1) was slightly better than prediction mode (2). That observation is of little import, however, as neither prediction mode substantially distanced itself from the other in terms of accuracy. Furthermore, the truly surprising result of this permutation of trials was that using single-base AMI profiles, on average, regardless of the chromosome or prediction method employed, scored *worse* in terms of accuracy than using the more general, standard AMI profiles. This was contrary to the intuition which was the impetus for exploring single-base AMI profiles to begin with. Thus, the exploration of single-base AMI profiles proved to be futile, as no gains in accuracy of prediction were observed; rather, on average, it had the opposite effect.

3.2.4 Summary

The *ad hoc* prediction algorithm produced lackluster results for accuracy in predicting missing bases in DNA sequences, and it did not show any promise for being the basis of a compression technique for DNA sequences. A summary of the highest average accuracy results and the highest accuracy results that could be obtained from any particular trial can be found in Table 3.1. This clearly

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.264072	0.271856	0.269788	0.273818	0.268623	0.271154	0.267234	0.270982	0.270022	0.280055	0.269589	0.276082	0.269551	0.26839	0.272434
2	0.267814	0.276892	0.281426	0.283533	0.288671	0.283929	0.279464	0.284035	0.286508	0.280952	0.285714	0.278067	0.282032	0.282993	0.276497
3	0.261753	0.283582	0.28523	0.287937	0.280952	0.286533	0.28331	0.284959	0.286243	0.286623	0.284028	0.2824	0.284432	0.278511	0.283393
4	0.281095	0.292964	0.297552	0.290972	0.290893	0.292657	0.294741	0.295536	0.295166	0.288056	0.288399	0.295106	0.287679	0.287883	0.288239
5	0.278144	0.298951	0.291799	0.291853	0.291089	0.29707	0.29915	0.292614	0.294367	0.293582	0.294444	0.288771	0.287706	0.292385	0.282941
6	0.303147	0.295437	0.299107	0.297401	0.298571	0.295933	0.296939	0.299826	0.291211	0.296984	0.29114	0.288765	0.296154	0.294188	0.294931
7	0.280556	0.310938	0.305776	0.3	0.30369	0.297944	0.299901	0.297015	0.297354	0.297881	0.291883	0.29434	0.295324	0.293824	0.290362
8	0.307143	0.310891	0.302015	0.30372	0.300649	0.302546	0.292644	0.297619	0.302731	0.294286	0.297427	0.293056	0.297115	0.292391	0.293864
9	0.317822	0.298077	0.304762	0.30487	0.304722	0.295896	0.303515	0.301589	0.297321	0.303868	0.295187	0.303212	0.291639	0.298458	0.295317
10	0.294505	0.30625	0.302814	0.306597	0.297761	0.307937	0.297821	0.299777	0.301782	0.297843	0.303693	0.291486	0.301661	0.296259	0.293252
11	0.279762	0.304221	0.30625	0.296269	0.307778	0.3	0.302679	0.302123	0.300218	0.301875	0.291798	0.301989	0.290751	0.29547	0.289829
12	0.298052	0.307292	0.298259	0.306548	0.307288	0.30625	0.302291	0.300368	0.297917	0.295217	0.30093	0.291468	0.295403	0.292399	0.296404
13	0.3	0.297761	0.307672	0.305085	0.309286	0.300157	0.303501	0.301432	0.295048	0.300682	0.292316	0.293902	0.294083	0.299906	0.299537
14	0.291791	0.306349	0.305367	0.307812	0.301509	0.304248	0.302679	0.292799	0.303914	0.290952	0.295676	0.296154	0.297976	0.29881	0.292286
15	0.303968	0.298729	0.304464	0.296462	0.308431	0.304514	0.289907	0.303977	0.293254	0.299878	0.298485	0.302193	0.300107	0.292245	0.297059

Figure 3.7: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 9 using the weighted-voting prediction mode and single-base AMI profiles.

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.264072	0.271856	0.269788	0.273818	0.268683	0.271154	0.26712	0.270982	0.270077	0.280055	0.269589	0.276082	0.269551	0.26839	0.272434
2	0.281437	0.285159	0.28607	0.283234	0.282797	0.280423	0.278508	0.280755	0.285653	0.281429	0.284711	0.279803	0.282549	0.281746	0.27452
3	0.273108	0.285697	0.281038	0.287063	0.280556	0.277604	0.278147	0.284478	0.285582	0.284481	0.283649	0.280846	0.280037	0.280266	0.278988
4	0.305224	0.291018	0.290559	0.28621	0.284196	0.284901	0.286578	0.291295	0.284776	0.290625	0.283379	0.287368	0.285463	0.281824	0.28522
5	0.284431	0.288112	0.287698	0.284598	0.283564	0.287363	0.291071	0.283604	0.289043	0.284478	0.288095	0.286935	0.286813	0.283356	0.28085
6	0.294406	0.282738	0.288095	0.288738	0.281978	0.289683	0.287106	0.289062	0.278109	0.287143	0.285285	0.283631	0.28389	0.281863	0.29
7	0.267063	0.292411	0.294224	0.284615	0.286429	0.284091	0.29127	0.284049	0.285714	0.28839	0.286364	0.285849	0.28273	0.295685	0.280652
8	0.295982	0.294307	0.285165	0.278274	0.285455	0.28831	0.279211	0.286806	0.29275	0.286518	0.284563	0.284559	0.29351	0.279503	0.287652
9	0.300495	0.28489	0.275595	0.289773	0.289861	0.283085	0.282653	0.286758	0.286409	0.284528	0.283779	0.292014	0.276505	0.28474	0.278492
10	0.284066	0.277381	0.285714	0.292535	0.286567	0.289286	0.282809	0.291295	0.280608	0.286667	0.294886	0.277899	0.284703	0.279507	0.286829
11	0.264881	0.291883	0.288426	0.286194	0.289683	0.278955	0.28699	0.278538	0.28573	0.295625	0.27836	0.286174	0.27674	0.286934	0.281453
12	0.281169	0.296875	0.279851	0.293849	0.285424	0.285119	0.281941	0.282475	0.295486	0.280326	0.285124	0.278472	0.287617	0.28196	0.285439
13	0.296528	0.28694	0.293915	0.276271	0.278929	0.277201	0.280952	0.298568	0.278019	0.2875	0.278788	0.29126	0.281164	0.281485	0.286667
14	0.297761	0.280159	0.277966	0.27433	0.275283	0.283987	0.290923	0.276087	0.288258	0.277976	0.290909	0.284081	0.282287	0.287698	0.287143
15	0.269048	0.275847	0.279167	0.275943	0.279608	0.289931	0.273292	0.285653	0.278175	0.29	0.284848	0.285088	0.284829	0.285612	0.278235

Figure 3.8: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 9 using the highest-weight prediction mode and single-base AMI profiles.

demonstrates that the best accuracy values that could be expected on average were only around 30-34%, while the highest accuracy ever observed for any trial performed never exceeded 57%.

Upon closer examination of the prediction results, some sequences which were tested tended to display atypical results. For instance, in some sequences the appearance of one particular base was very high. This would cause the joint probabilities calculated for the sequence to skew heavily in favor of the base which appeared frequently. The result of this is that these high joint probabilities would influence the prediction algorithm such that every base in the sequence was

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.273653	0.277021	0.271182	0.270025	0.264371	0.272494	0.263209	0.266685	0.260176	0.266374	0.261634	0.262013	0.263408	0.266684	0.263228
2	0.296557	0.29253	0.283582	0.281287	0.27958	0.270899	0.278508	0.275248	0.280769	0.276845	0.275325	0.266377	0.271584	0.271712	0.270113
3	0.296813	0.290547	0.280539	0.283304	0.282302	0.281101	0.27942	0.277953	0.281217	0.279675	0.272285	0.274627	0.275885	0.277845	0.28244
4	0.288308	0.282036	0.287063	0.287103	0.280893	0.280693	0.286342	0.286979	0.278499	0.283333	0.275373	0.284524	0.283703	0.285204	0.271887
5	0.28982	0.292832	0.290079	0.282589	0.282475	0.288919	0.290221	0.283036	0.282485	0.279627	0.282395	0.284534	0.284272	0.276146	0.280523
6	0.302098	0.302579	0.287946	0.291708	0.290769	0.295536	0.289703	0.28724	0.285489	0.289206	0.286055	0.285268	0.279536	0.284034	0.282361
7	0.302381	0.29442	0.290594	0.296154	0.29	0.289286	0.293353	0.28694	0.286243	0.288729	0.28474	0.284827	0.284992	0.282143	0.285072
8	0.314732	0.303218	0.294139	0.2875	0.291558	0.292014	0.292537	0.288294	0.288795	0.289464	0.288937	0.285948	0.282772	0.289984	0.280909
9	0.292574	0.303022	0.289087	0.296591	0.29625	0.299378	0.285488	0.294174	0.291567	0.288774	0.288681	0.286632	0.291221	0.283685	0.278175
10	0.307143	0.297024	0.293939	0.293403	0.295224	0.286508	0.299879	0.289509	0.287736	0.288137	0.286174	0.29375	0.282168	0.280357	0.283984
11	0.302381	0.303896	0.291667	0.290858	0.284127	0.295056	0.286224	0.295755	0.288671	0.289063	0.29496	0.285227	0.280952	0.283537	0.287009
12	0.301299	0.294792	0.292289	0.287897	0.296441	0.290923	0.295148	0.290564	0.29213	0.294674	0.285434	0.284623	0.291182	0.283425	0.285526
13	0.297222	0.30597	0.292328	0.298941	0.291429	0.292453	0.292437	0.291406	0.298309	0.290795	0.287771	0.290854	0.286982	0.28656	0.28963
14	0.285821	0.302778	0.303107	0.287277	0.294151	0.301471	0.290179	0.294429	0.292551	0.285714	0.291685	0.287714	0.287753	0.285813	0.287714
15	0.324603	0.29661	0.291667	0.295991	0.30902	0.289583	0.29736	0.294602	0.286905	0.293537	0.290326	0.291557	0.286325	0.288571	0.285882

Figure 3.9: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 15 using the weighted-voting prediction mode and single-base AMI profiles.

Group Length (Bases) \ Gap Length (Bases)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.273553	0.277021	0.271315	0.270025	0.264371	0.272494	0.263379	0.266685	0.260176	0.266374	0.261634	0.262013	0.263248	0.266684	0.263175
2	0.290419	0.282669	0.276202	0.277171	0.274476	0.274537	0.275191	0.272463	0.272405	0.273155	0.266588	0.267824	0.269747	0.270011	0.264237
3	0.287849	0.280597	0.27984	0.278409	0.279286	0.283408	0.276096	0.276923	0.276984	0.274351	0.272664	0.271331	0.275336	0.273487	0.275893
4	0.279104	0.279491	0.284149	0.28006	0.277589	0.277228	0.278807	0.277381	0.27583	0.277778	0.275034	0.275198	0.274381	0.279401	0.269623
5	0.282635	0.287063	0.28082	0.28192	0.278614	0.282509	0.281378	0.274026	0.281327	0.276866	0.275685	0.273799	0.280495	0.26907	0.272288
6	0.291958	0.289087	0.285119	0.283787	0.276703	0.281746	0.2782	0.279688	0.27728	0.278016	0.273035	0.279911	0.272569	0.276961	0.277014
7	0.286111	0.28058	0.285644	0.286126	0.279762	0.280411	0.281548	0.28125	0.280159	0.279661	0.277029	0.275786	0.277451	0.275521	0.275072
8	0.2875	0.287624	0.289927	0.280506	0.281299	0.282755	0.282516	0.279861	0.281168	0.281161	0.278902	0.276797	0.279327	0.281289	0.276591
9	0.282673	0.299725	0.281548	0.286039	0.28125	0.285075	0.279025	0.283475	0.279464	0.278774	0.278966	0.280469	0.28495	0.274269	0.273492
10	0.298901	0.283036	0.285281	0.286285	0.282239	0.279233	0.283051	0.276674	0.281342	0.282549	0.281723	0.283062	0.275699	0.274575	0.280569
11	0.283333	0.286688	0.283796	0.287127	0.282698	0.286299	0.277296	0.282665	0.283551	0.278958	0.279348	0.27822	0.277289	0.281969	0.277009
12	0.283766	0.284375	0.285075	0.281548	0.288305	0.284375	0.283288	0.288113	0.276968	0.280435	0.279029	0.275595	0.281614	0.271154	0.277982
13	0.301389	0.294403	0.284392	0.281144	0.278214	0.28695	0.284734	0.278776	0.280435	0.279205	0.275974	0.278354	0.271598	0.278383	0.280278
14	0.277612	0.289286	0.285028	0.271429	0.284151	0.285621	0.277827	0.28125	0.277778	0.277738	0.280931	0.275641	0.281377	0.281052	0.278095
15	0.28254	0.289407	0.277381	0.275708	0.28902	0.27691	0.284627	0.280114	0.278042	0.281463	0.277156	0.280263	0.282265	0.279082	0.277549

Figure 3.10: Heat map showing the average accuracy of predicting missing bases with varying group and gap lengths for 20 regions on human chromosome 15 using the highest-weight prediction mode and single-base AMI profiles.

predicted to be this most frequently appearing base. Sometimes this effect could occur for a particular pair of bases as well, such as A and T, where the probability for A and T was so high that it skewed the joint probabilities in such a way that C or G were never predicted for an empty position.

Another element of the prediction algorithm that was thought to perhaps skew prediction results was the fact that the AMI profile values for short distances (e.g. lags $k = 1$ to $k = 5$) tend to be very large compared to other AMI values and dwarf the effect of the other mutual information relationships present for longer distances, as can be seen in Figure 3.11. Thus, other attempts were made to test the results of

	Prediction Mode 1		Prediction Mode 2	
	Max. Avg.	All Max.	Max. Avg.	All Max.
Chromosome 9				
Full AMI	33.08%	50.55%	32.55%	55.22%
Single AMIs	31.78%	48.47%	30.52%	48.61%
Chromosome 15				
Full AMI	32.58%	56.94%	33.63%	51.59%
Single AMIs	32.46%	57.14%	30.14%	56.94%

Table 3.1: Summary table of best results for each prediction mode attempted. The maximum value from among the averages over each position on the chromosome is presented as "Max Avg.," and the maximum result from any individual trial performed on an individual sequence is presented as "All Max."

prediction using AMI profiles that had the beginning lag values either eliminated entirely or set to an average value of the rest of the AMI profile. The results of these trials did not indicate any substantial improvement over results that had already been observed.

Yet another aspect of AMI profiles as related to their ability to predict DNA bases was explored. When the joint probabilities of the bases in DNA sequences were calculated, these tables were maintained asymmetrically. In other words, *order mattered*. Different joint probabilities were calculated for the case when, for instance, a G and then an A appeared or an A appeared and then a G, k base positions apart. Since the probability tables were maintained asymmetrically (if A came 3 bases after G, that was tabulated separately than if G came 3 bases after A), this asymmetry was utilized in the forwards and backwards predictions. However, another symmetric test was run where this was not the case, but it had no noticeably different results.

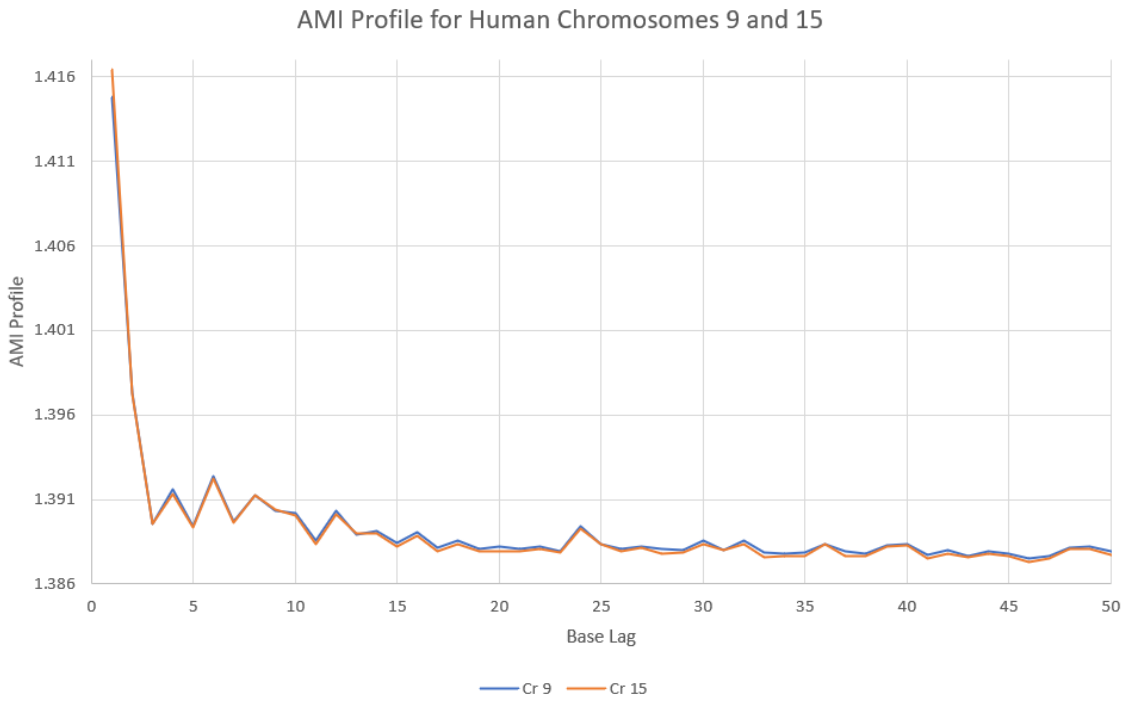

Figure 3.11: AMI profile for human chromosomes 9 and 15 showing the large AMI-profile values for low base lags (k values).

3.3 Best Predictor Analysis Method

Due to the unimpressive results obtained in the attempt to arbitrarily retain and discard bases in a DNA sequence for the purposes of prediction, a new method was sought that could perhaps help determine which bases are more likely or more suited to predict other bases more accurately. A method which could evaluate the prediction performance of a base would more closely approximate the method related to the original inspiration for this project—the inpainting methods described in [5]. In the algorithm that selects which pixels to retain and which to discard, the pixels retained are in strategic locations, such as near sharp features.

Thus, a second, less "*ad hoc*" method was devised for choosing a reduced compression set and seeing how missing bases could be predicted using this reduced set. This was done by first attempting to assign a score to all known bases on a

strand by how good of a "predictor" each base was. Then, only the bases that were scored as the best predictors were retained in the compression set, and the bases which were scored as the worst predictors were removed from the sequence. These were the bases to be predicted by the retained bases.

3.3.1 Process and Methods

To perform the best predictor analysis, again a completely known DNA sequence was considered. As before, the probability of each base's occurrence in the sequence was determined for each base according to Equation (1.7). Using this, the joint probabilities of the occurrence of each base with respect to every other base k positions away were determined according to Equation (1.9). Finally, using these probabilities, the AMI profile was calculated for this sequence according to Equation (1.11).

The prediction ability of each base was determined within a predefined "window" around its position. (For example, its ability to accurately predict bases up to 10 positions before and 10 positions after its position was considered.) Using the joint probabilities calculated for the entire sequence, the base in question would attempt to "predict" each base in its surrounding window. The prediction made by this base was simply the base with the maximum joint probability value for a base k positions away from the base in question. For example, if the base in question was A, and it was predicting a base three positions after its position, the prediction would be the base from the set {A, C, G, T} which had the highest joint probability with A for a distance of $k = 3$.

Once the base in question had "predicted" all bases around itself within the window, these predictions would be compared to the actual bases in those positions on the sequence to determine a score for how accurately the base in question could

predict all those around it using only the sequence's joint probabilities. Two methods were used for computing this "best predictor" score: (1) The "simple method" simply counted the number of correct bases which were predicted. For example, if the window was 10 bases before and after, and the base in question predicted 14 of those 20 bases correctly, then its score would be 14. (2) The "AMI method" computed the score in such a way that a correct prediction was weighted by the AMI-profile value for how many positions away from the base in question the correct prediction was. Thus, the final score was the sum of the associated AMI-profile values.

These "predictor scores" were then calculated for each base on the sequence. Once a predictor score for each base on the sequence was determined, only bases that had a high score were retained in the reduced compression set. This was also done in one of two ways: (1) To determine which bases to keep in the compression set, a static cutoff value was defined heuristically for the whole DNA sequence. Bases scoring above this cutoff threshold would be retained as good predictors and other bases below it would be discarded in order to be predicted by the good predictors. (2) The other way to do this was to use a moving average of predictor scores. Thus, instead of a static value for the whole sequence, each base was retained if it could beat the moving average of the scores of the bases around it, making the bases retained the best *local* predictors.

Once the bases with the lowest predictor scores were discarded, the bases with the higher scores would be used to predict the missing bases. However, in this scenario, due to the way bases were selected to be retained, these bases were scattered in positions throughout the DNA sequence. Thus, the prediction of bases would also need to be done using the same window length that was used to build the prediction scores. In a similar manner to how missing bases were predicted with

the *ad hoc* "group and gap" method, empty base positions were predicted by using all the retained bases within the window length surrounding the missing base. These bases all made a guess as to which base should be in the missing position, and their guesses were weighted by the AMI profile value for how far away they were from the empty base position in question. The base predicted with the best weighted score from all surrounding retained bases was chosen as the prediction for that empty base position.

3.3.2 Prediction Accuracy Results

Once again, the best predictor analysis was tested on DNA sequences obtained from human chromosomes 9 and 15. Best-predictor scores could be calculated for every base in a DNA sequence, and plots could be generated showing the relationship of predictor score to the base position on the chromosome. When examining the "lay of the land," so to speak, for best predictor scores over regions of 10,000 to 50,000 bases, it was noticed that these predictor scores did not maintain the same average over the whole region. Rather, they seemed to exhibit trends. In other words, the predictor-score values would rise or fall as one moved along the chromosome. Some regions of the chromosomes just seemed harder to predict in general, while bases in other regions had higher predictor scores. This could have been due to repeats or the frequent recurrence of a single base in a certain region. Nonetheless, scores for all bases in particular regions seemed to trend higher or lower together. Thus, the "moving average" method for determining which bases to retain and which bases to discard quickly proved itself to be superior. Trying to beat a moving average in order to be retained was better than a static threshold value, because in the case of a static value, it would end up that lots of bases would be retained in the compression set in one region and virtually none in another region.

As for the results of this method of prediction, the accuracy results when comparing predicted bases to known bases which had been discarded were virtually indistinguishable from the average accuracy results for the *ad hoc* "group and gap" method presented previously, and thus they are not reproduced here. It was still very difficult to break 50% accuracy with any regularity throughout the regions tested. However, it did raise the accuracy slightly over the *ad hoc* "groups and gaps" method. That method usually predicted with about 30-35% accuracy on average. This method usually predicted with about 35-40% accuracy on average, so that represented marginal but not significant improvement.

In order to better ascertain the effectiveness of this best-predictor score method, an approach was devised to rank the "confidence" that could be expected for a certain base prediction. In other words, can this prediction algorithm know the quality of its predictions by quantifying how good its predictions are likely to be? As described previously, during the process of determining which bases to retain and which to discard, a predictor score was used to score the bases on how many surrounding bases they could correctly predict. During the base prediction phase, the confidence score was then calculated based on how "sure" each base was in predicting the bases that it predicted. When a base in the retained set predicted another base, the value of its weight in the voting was added to the confidence score of the base predicted *only if* its guess was chosen as the final base prediction. For example, in predicting a missing base with AC_GT, the confidence score for prediction of a base in the empty position would be determined from the surrounding bases. Thus, if it is assumed that A, C, and G all guessed the missing base to be C, but T guessed it to be A, then C would be chosen as the prediction for that empty position (assuming the weight of T's vote didn't substantially outweigh

the other three bases), and the value of the weighted votes of A, C, and G (but not T) would be added together to obtain the confidence score of the prediction.

The hypothesis was that base predictions which had a higher confidence score would hopefully be more accurate predictions. If this were to be the case, it was surmised that an iterative method could be developed wherein a DNA sequence with scattered missing bases would undergo a first phrase of prediction, but only base predictions with a high enough confidence score would be retained. Then, a second phase of prediction would occur, now using the "highly confident" predictions as predictors for the base positions that remained empty. Doing this iteratively until all positions were filled could increase accuracy if the confidence in base predictions could be shown to be correlated with correct predictions.

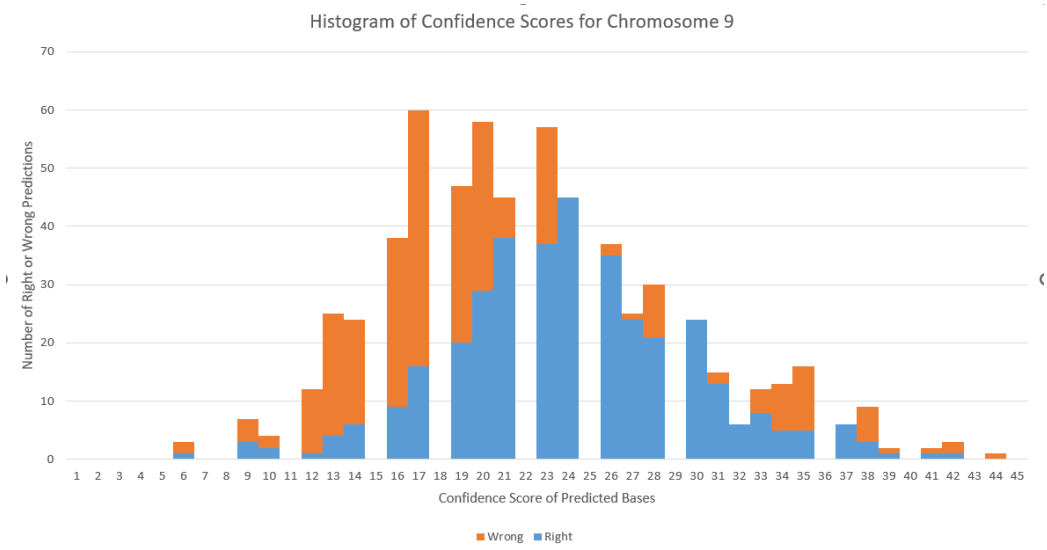

Figure 3.12: Histogram showing the numbers of right base predictions and wrong base predictions for values of the confidence score for a DNA sequence from human chromosome 9.

However, this was found not to be the case. Rather, predicted bases that had higher confidence scores were only found to be slightly more likely to be a correct prediction. A histogram containing the confidence scores for bases predicted

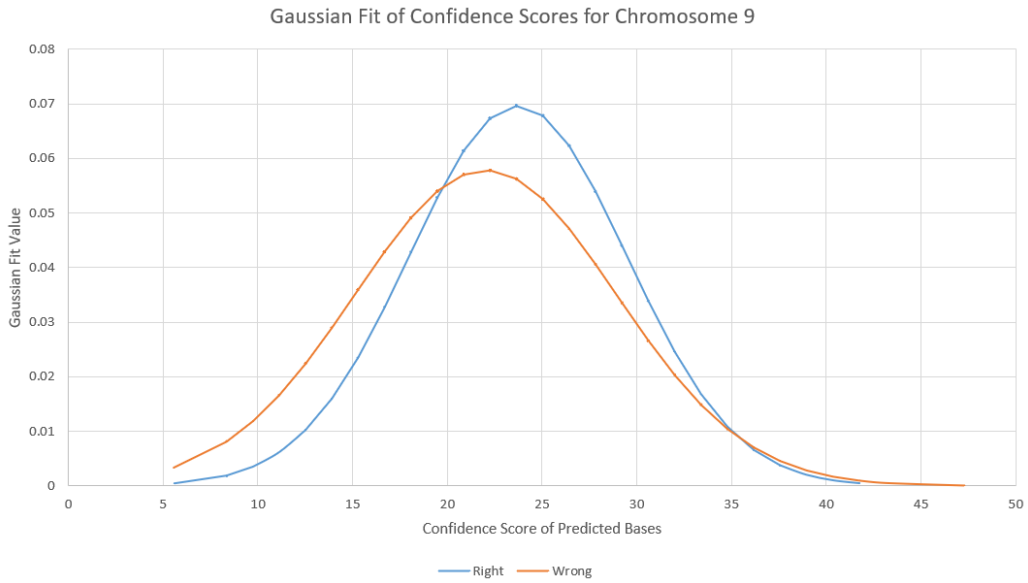

Figure 3.13: Gaussian fit showing the relative distributions of right and wrong base predictions over the resulting confidence score for a DNA sequence from human chromosome 9.

correctly and incorrectly on chromosome 9 is shown in Figure 3.12. A Gaussian fit to the normal distributions present in this histogram is shown in Figure 3.13. As it can be seen from these figures, the distribution of correct predictions has a slightly higher confidence-score mean than the distribution for incorrect predictions; however, the overlap of distributions for right and wrong predictions is clearly too significant, indicating that right and wrong predictions cannot be clearly differentiated from each other based on confidence score alone.

These same results are confirmed again with chromosome 15. Both the histogram in Figure 3.14 and its Gaussian fit in Figure 3.15 reveal that the separation that would be needed between the right and wrong prediction distributions is not present in the result. This indicates that, not only can the best predictor algorithm not predict DNA bases with significant accuracy, it also cannot differentiate between the predictions that it has made, as to whether they are more likely to be correct or incorrect predictions.

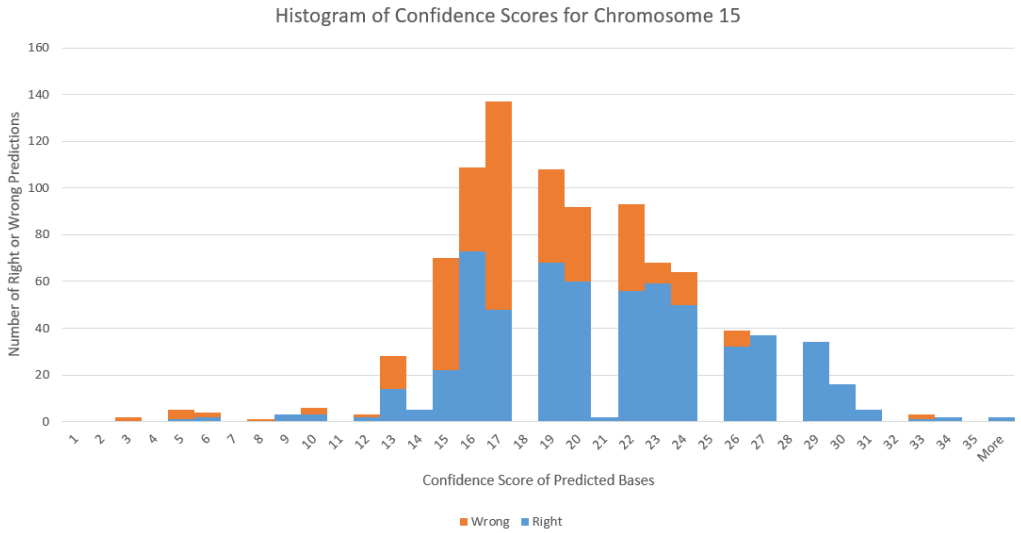

Figure 3.14: Histogram showing the numbers of right base predictions and wrong base predictions for values of the confidence score for a DNA sequence from human chromosome 15.

3.4 Coding Region Prediction

As a final way to truly ascertain whether using the AMI profile to predict DNA bases will be of any substantial benefit to developing a compression technique, a few more focused experiments were conducted. Since the prior analysis focused on arbitrarily selected regions of human chromosomes, it was thought that an analysis should focus on regions which, through prior biological knowledge, are known to contain highly structured and functional information. It was hypothesized that regions of this type would have the highest chance of correct predictions being possible using methods heretofore explored. The obvious candidates for this criterion are coding regions. Two well-attested genes were selected: the human beta globin gene (BGLT3 beta globin locus transcript 3: NC_000011.10:c5245546-5244554 *Homo sapiens* chromosome 11, GRCh38.p13) and the polymerase III beta gene from *Staphylococcus aureus* (SAOUHSC_00002 DNA polymerase III subunit beta).

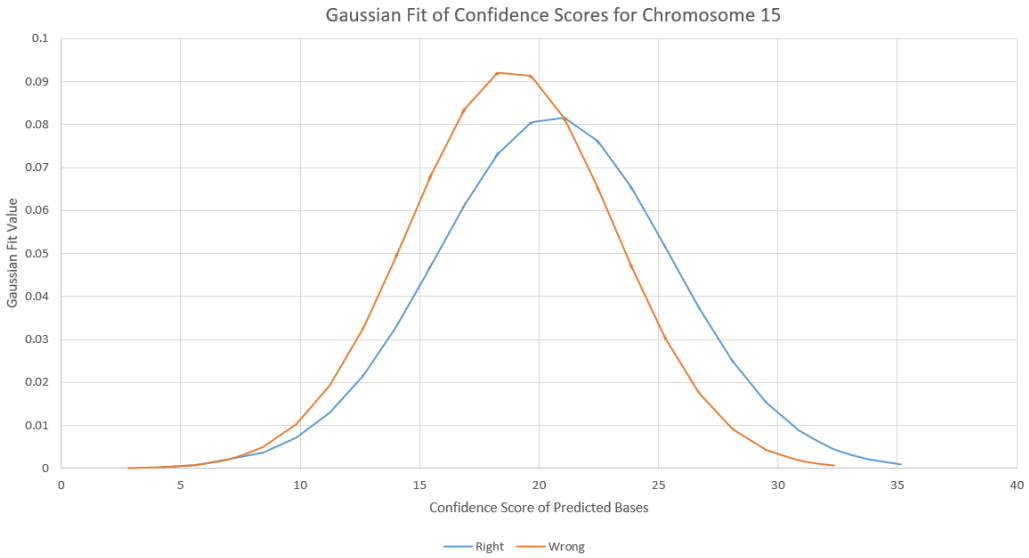

Figure 3.15: Gaussian fit showing the relative distributions of right and wrong base predictions over the resulting confidence score for a DNA sequence from human chromosome 15.

The best predictor analysis method was used to both decide which bases to retain in the compression set for each of these sequences and then to predict the missing bases using a window length of 25 bases surrounding each missing base. Instead of simply using the AMI profile, three different types of weighting methods were employed to determine how each surrounding base's guess was to be weighted in making the prediction. First, simply the raw joint probability was used for each base to make its guess for the overall prediction. The guess of the surrounding base for the empty position k spaces away was simply the base that had the highest joint probability k positions away with the surrounding base. Second, as defined before in Equation (3.1), the single-base AMI profile value was used. Third, a new metric was explored to quantify the strength of the association between two base positions. This was the log odds ratio, and it was calculated as shown in Equation (3.2) for bases X and Y that are k positions apart. (This is equivalent to the simple mutual information defined in Equation (1.10).) Once each surrounding base had made its

guess for the empty position in question using one of the three prior methods, these results were combined using either unweighted voting (i.e., each base gets a single vote) or weighted voting (i.e., each base's vote is also weighted by its resulting guess metric, whether joint probability, single-base AMI, or log odds ratio).

$$LOR_k(XY) = \log_2 \left(\frac{p_k(XY)}{p(X)p(Y)} \right) \tag{3.2}$$

The results for these trials can be found in Table 3.2. It can be seen that all of the accuracy values observed are mostly within the same range (30% - 50%) as those previously observed for prior experiments. Thus, none of these adjustments nor the fact that the DNA sequences were specifically selected from coding regions significantly increased the accuracy of the predictions made. It can be seen that the log odds ratio was not helpful in correctly quantifying the DNA sequence information structure and was, in fact, detrimental, producing some results which were worse than random guessing (25%). Applying the log odds ratio in this context was not useful. Furthermore, the single-base AMI profile values still performed the best when compared with simply the raw joint probabilities, indicating that the AMI profile does add at least a slight measure of usefulness in quantifying the information structure of DNA sequences. In the end, nothing was found with these additional trials to warrant any further investigation of these prediction techniques.

3.5 Conclusion

As the explorations in this chapter have repeatedly shown, models of the information structure contained in DNA sequences extracted from human chromosomes which rely on pure probability-based calculations are not sufficient to quantify the information structure to a degree that would make compression

Prediction Function	Human	*S. aureus*
Weighted Voting		
Raw Joint Probabilities	35.35%	36.16%
Single-base AMI Values	38.07%	38.98%
Log Odds Ratio	29.31%	21.25%
Unweighted Voting		
Raw Joint Probabilities	35.95%	36.16%
Single-base AMI Values	37.16%	42.33%
Log Odds Ratio	32.62%	24.78%

Table 3.2: Summary table of average accuracy results for using the best predictor analysis method to predict bases on the human beta globin gene and the *S. aureus* polymerase III subunit beta gene using three different types of prediction methods: raw joint probabilities, single-base-AMI-profile values, and the log odds ratio.

possible. Solely relying on the joint probabilities between bases as well as various information metrics related to those probabilities (chiefly the AMI profile and its derivative single-base AMI profiles) cannot sustain the level of accuracy needed to indicate a deep understanding of the sequence's information structure. From this point forward in this investigation of DNA's information structure, other facets of DNA sequences, such as triplet codes and the difference between coding and noncoding regions, which are known from biological techniques, will be employed in order to determine whether the information structure of DNA sequences can be better described using a combination of probability-based models and these known entities.

Chapter 4

Analysis of Triplet Code Features

4.1 Introduction

DNA is structured as a sequence of bases that constitutes a code which bears information relevant to the biological processes of an organism. It is well known that one of the main ways in which this information is exposed and made usable is through the processes of transcription and translation. These processes are how the cell uses DNA to create proteins, and they are based upon interpreting DNA as a triplet code. For coding regions of the DNA sequence, each group of three bases (a triplet) corresponds to a single amino acid. These amino acids are then linked end-to-end in a polypeptide chain which becomes the "building block" of a protein.

While it is known that coding regions of DNA operate in this manner, noncoding regions *may* or may not. Since noncoding regions are generally expressed with less frequency than are coding regions, their interpretation and structure are harder to assess. However, if the assumption is made that noncoding regions operate in like manner to coding regions, with a triplet code, those regions can be analyzed together with coding regions to explore how their information content is related.

4.2 AMI Profiles Based on Triplet Codes

As described in Chapter 1, AMI profiles as described in [1] are calculated with regard to single bases. In other words, the AMI profile uses the joint probabilities of each base encountering another base k positions apart. The AMI profile then averages over these joint probabilities for the four letters in the DNA alphabet to discern a relationship based *solely* on the distance between bases (k).

A similar metric can be formulated that calculates the relationship not between bases separated by k base positions but rather between triplets of bases which are separated by k base positions. Thus, each triplet in the DNA sequence is considered a unique "letter" in this alphabet, and thus these triplet AMI profiles are calculated by averaging over 64 possible triplets to obtain the relationship of triplets with respect to the distances between triplets only. The triplet AMI profile can be calculated in two ways: (1) An "overlapping" triplet AMI profile can be calculated that moves along the DNA sequences base-by-base and thus considers each triplet as existing one base apart. This means that each triplet overlaps with two others, the ones before and after it. (2) A "non-overlapping" triplet AMI profile can be calculated that moves along the DNA sequence triplet-by-triplet and thus considers each triplet as existing three bases apart. This means that triplets do not overlap with other triplets. It also means that this formulation is subject to a reading frame. Since DNA is read by a triplet code, the position at which the reading begins is very important. A reading that starts one base later than another will produce very different triplet results.

For example, in the DNA sequence GAGACATTACGTACC, the overlapping triplet AMI would analyze the triplets in the sequence as follows: GAG, AGA, GAC, ACA, CAT, ATT, TTA, and so on. This AMI profile is indexed by each *base*

position (the first base in the triplet) and thus has the same indexing as the standard AMI profiles discussed in Chapter 1. Using the same example sequence, a non-overlapping triplet AMI would analyze the triplets in the sequence as follows: GAG, ACA, TTA, CGT, ACC. However, this is only one of the possible reading frames for this sequence. If a reading frame starting at the second base in this sequence is used, then the result is as follows: AGA, CAT, TAC, GTA, CC _ (whatever the next base in this sequence is). Obviously, then, the last possible reading frame would begin at the third base position in the sequence: GAC, ATT, ACG, TAC, C _ _ (whatever the next two bases in this sequence are).

When studying the effects of the reading frame on the resulting triplet AMI profile, it was discovered that reading frame had no effect on the resulting triplet AMI profile. Though the reading frame is *critical* for protein translation (and thus a frameshift can be catastrophic for decoding the correct amino acids in the polypeptide chain), the general information structure of DNA, even when studied with respect to triplets, is the same. This can be seen demonstrated on human chromosome 9 in Figure 4.1 and Figure 4.2 (which is just a close up of the first 40 AMI-profile lags of Figure 4.1). Despite extremely small differences, the resulting triplet AMI profiles are virtually indistinguishable regardless of the reading frame employed. Thus, for non-overlapping AMI profiles, different reading frames were not tested and were not taken into account.

4.2.1 Triplet AMI Profiles in the Chromosomes of Humans and Other Similar Species

As demonstrated thoroughly in [1], the AMI profile calculated from any piece of genetic material (such as a full chromosome or even a fragment of DNA) resembles a "common shape" for all the genetic material from a single species; however, that

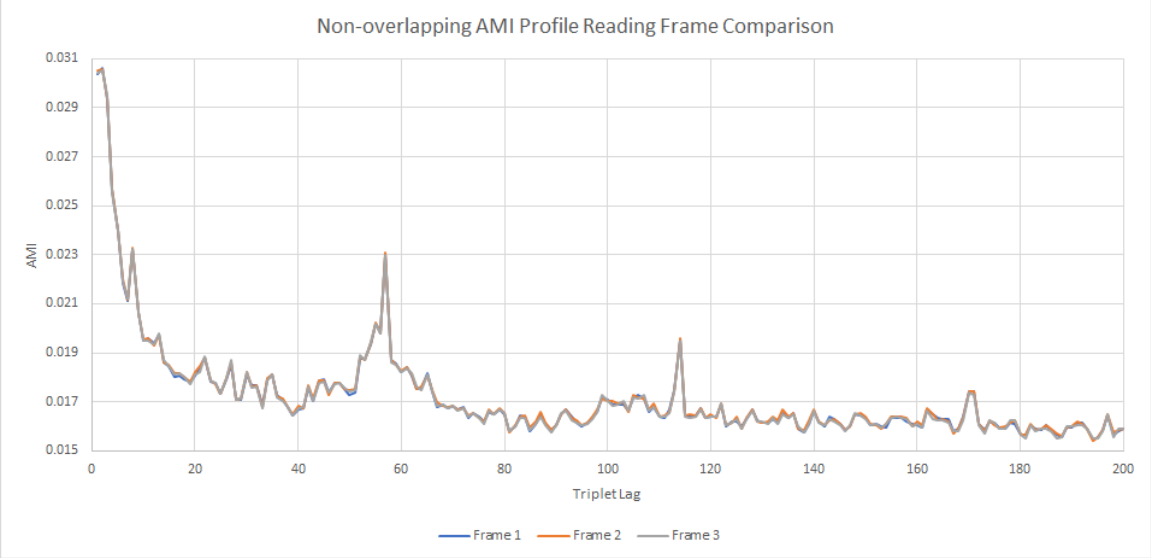

Figure 4.1: Non-overlapping triplet AMI profile for human chromosome 9 with 200 triplet positions (600 base positions) of lag for all three reading frames.

shape is different from species to species, leading to the conclusion that the AMI profile can be used as a reliable "species signature" to differentiate between different species. Thus, when the standard AMI profile (as described in Chapter 1) is calculated for all the chromosomes in the human genome, they all exhibit similar shapes, as can be seen in Figure 4.3.

Calculating the non-overlapping triplet AMI profiles for all the chromosomes in the human genome produced similar results to the standard AMI profiles in that the non-overlapping triplet AMI profiles calculated for each chromosome were all similar in shape. These results are shown in Figure 4.4. However, it can be seen that, when compared to the standard AMI profiles, fewer features are observed. The only major features that appear are at triplet position 8 and triplet position 57. The very strong feature present at triplet position 57 is somewhat striking in how it uniformly stands out on all human chromosomes. Overlapping triplet AMI profiles were also calculated for all human chromosomes, and the results were largely the same for non-overlapping triplet AMI profiles. These results can be seen in Figure

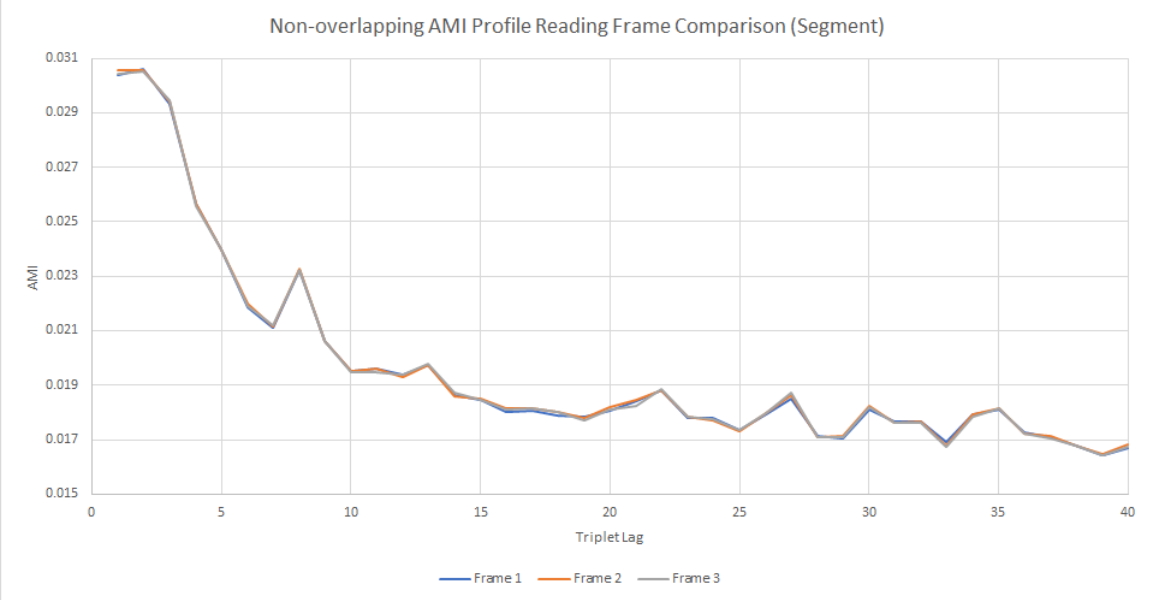

Figure 4.2: Non-overlapping triplet AMI profile for human chromosome 9 with 40 triplet positions (120 base positions) of lag for all three reading frames.

4.5. Note that overlapping triplet AMI profiles are indexed according to *base* position rather than *triplet* position, and thus the feature seen at base position 24 corresponds to the one seen at triplet position 8, and the feature seen at base positions 169-171 corresponds to triplet position 57.

The appearance of this type of feature at around base position 171 is not something that would naturally be expected. It means that, on averages, bases that are 171 positions away from each other, and similarly triplets that are 57 triplets away from each other, share mutual information, on average, more than most other bases, even those only 40 base positions away. To be sure, this feature can be seen within the standard AMI profiles, but it is only the triplet profiles that really accentuate its presence, while most other features seen in the standard AMI profile are muted when triplet relationships are considered. (Compare base positions where features appear at places like 56 and 135 in the standard AMI profile shown in

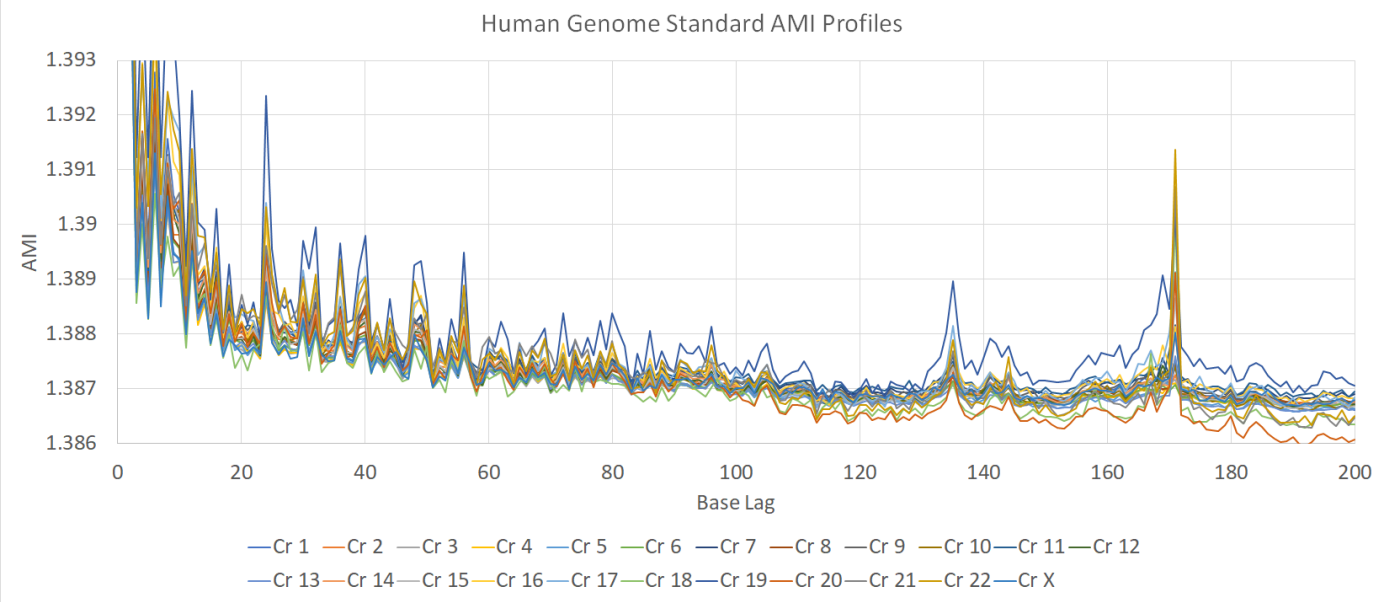

Figure 4.3: Standard AMI profiles for all 23 chromosomes of the human genome.

Figure 4.3 to the corresponding positions in the triplet AMI profiles shown in Figures 4.4 and 4.5.)

To ascertain whether considering the AMI profile using triplet pairs had any effect in identifying the feature, another type of AMI profile was created that would test for "duplets" (pairs of two bases). These duplet AMI profiles were constructed as a control, since bases are not known, biologically, to function in pairs. Rather, the goal was to see whether calculating the AMI in terms of triplets was contributing to the presence of the sharp feature at base position 171 or whether this was feature of the information structure was unrelated to DNA's triplet structure. Duplet AMI profiles, with a 16-letter alphabet, were calculated for all chromosomes in the human genome. Like triplet AMI profiles, these have both non-overlapping and overlapping versions. The results are shown in Figures 4.6 and 4.7.

The results obtained from the duplet AMI profile analysis were both expected and unexpected. When considering the bases as pairs, all distinct features in the

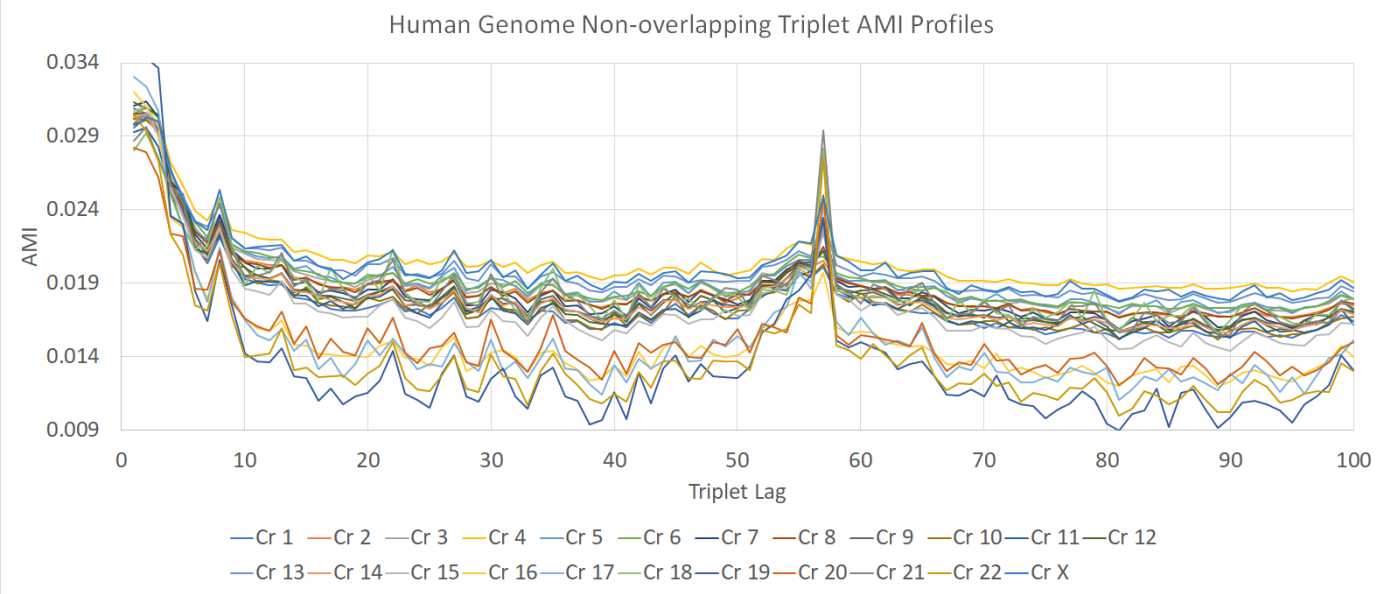

Figure 4.4: Non-overlapping triplet AMI profiles for all 23 chromosomes of the human genome.

AMI profiles vanished except for the one noted at base position 171. Even though this feature can be clearly seen in both the overlapping and non-overlapping duplet AMI profiles, its presence is muted and not as pronounced as it appears in the triplet AMI profiles. The other surprising result was that the feature did *not* appear in the expected location for the non-overlapping duplet AMI profiles. It was assumed that the feature would be present around duplet position 85, which would correspond to base position 171 ($85 * 2 = 170$). However, the feature still manages to appear around base position 171. It can be inferred, at least, that, though the feature appears amongst duplet AMI profiles, it is more accentuated in triplet AMI profiles, and thus considering DNA in triplets is relevant to the appearance of this feature.

Lastly, the relationship of this feature of the human genome was explored as it relates to the genomes of other similar species. The question of whether this feature was unique to the human genome seemed pertinent. Since the non-overlapping and overlapping methods of calculating the triplet AMI profiles for human chromosomes

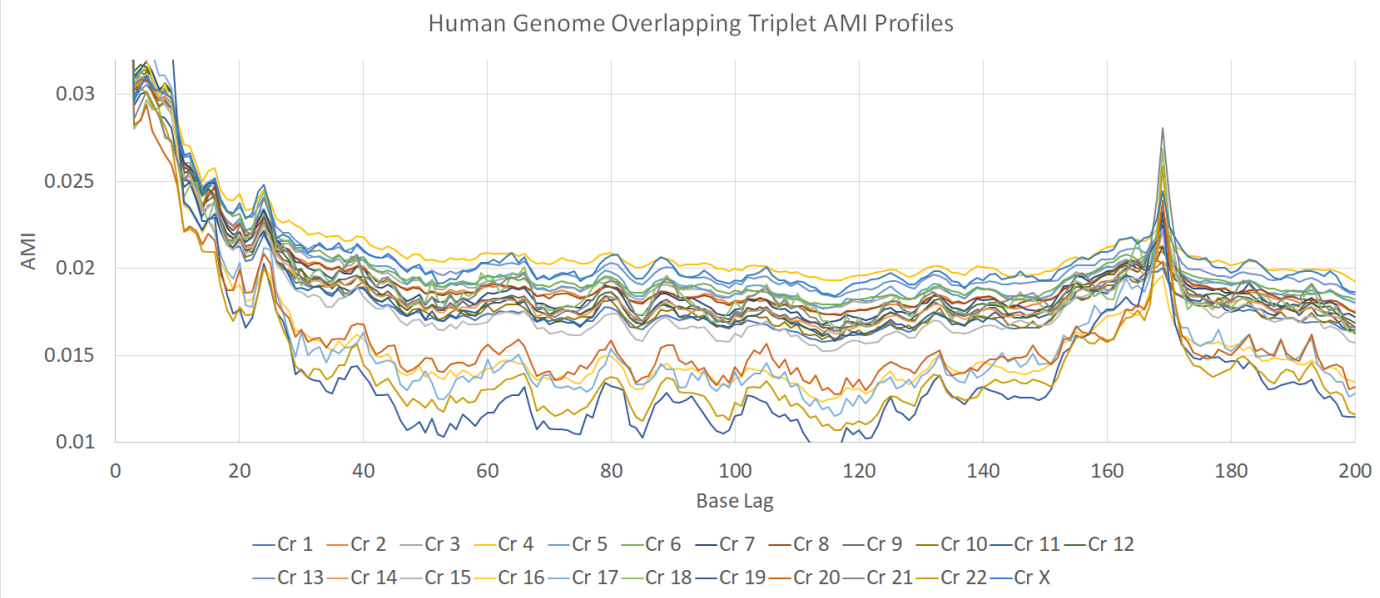

Figure 4.5: Overlapping triplet AMI profiles for all 23 chromosomes of the human genome.

seemed similar, only non-overlapping triplet AMI profiles were used to explore the genomes of other species. As a control, the triplet AMI profiles were calculated on the mouse (*Mus musculus*) genome, a mammal whose genome is sufficiently different from that of human beings. Those results can be seen in Figure 4.8. As expected if this feature is indeed unique to the human genome, no discernible features are observed in the triplet AMI profile calculated for the mouse DNA.

Next, triplet AMI profiles were calculated for all the chromosomes of the genomes of four species similar to human beings: the chimpanzee (*Pan troglodytes*), the gorilla (*Gorilla gorilla*), the orangutan (*Pongo abelii*), and the macaque (*Macaca fascicularis*). These results can be seen in Figures 4.9 to 4.12. All of these results show that features corresponding to those of the human genome at triplet positions 8 and 57 can be seen. However, the expression of these features in these genomes was weaker in each case than the strong feature present in the triplet AMI profiles of the human genome. Although an attempt was made to find any type of

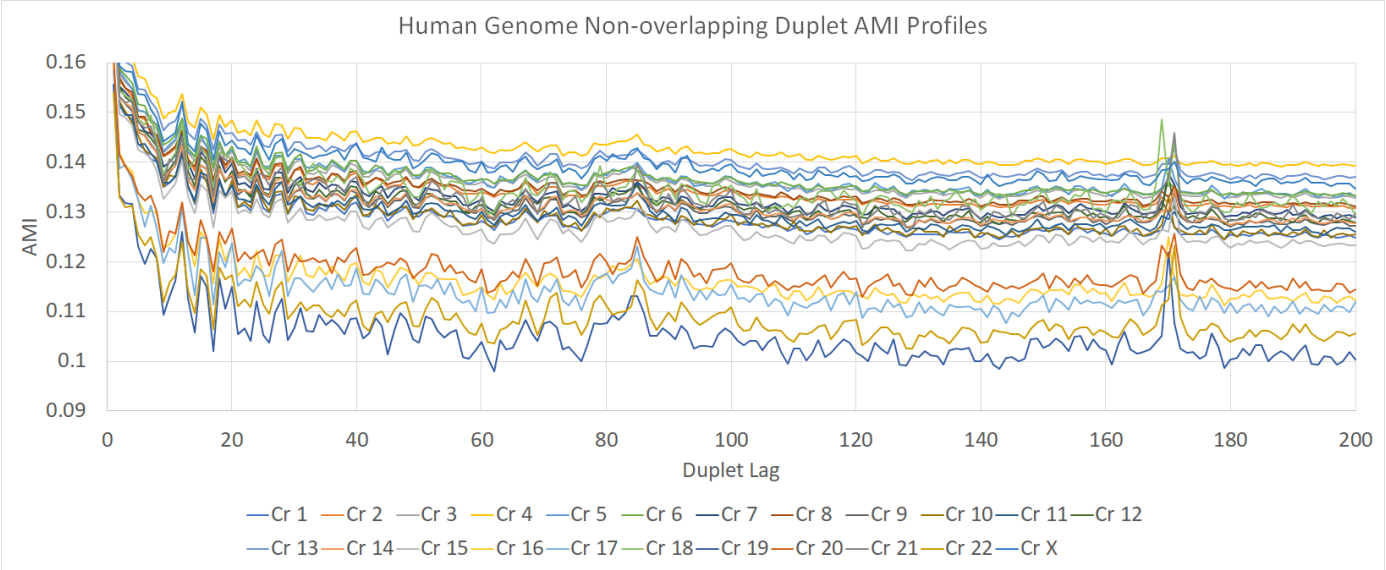

Figure 4.6: Non-overlapping duplet AMI profiles for all 23 chromosomes of the human genome.

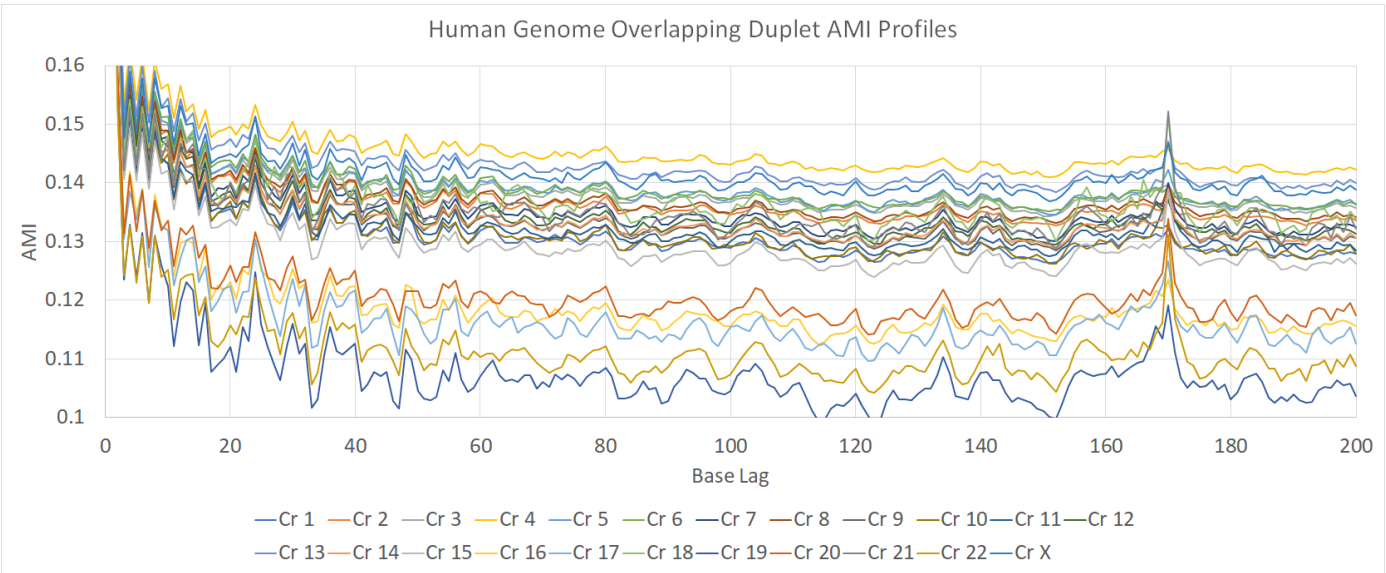

Figure 4.7: Overlapping duplet AMI profiles for all 23 chromosomes of the human genome.

reconstructed genome of extinct hominid species, no complete assemblies could be found which would allow sufficient triplet AMI profiles to be calculated to serve as a basis for comparison.

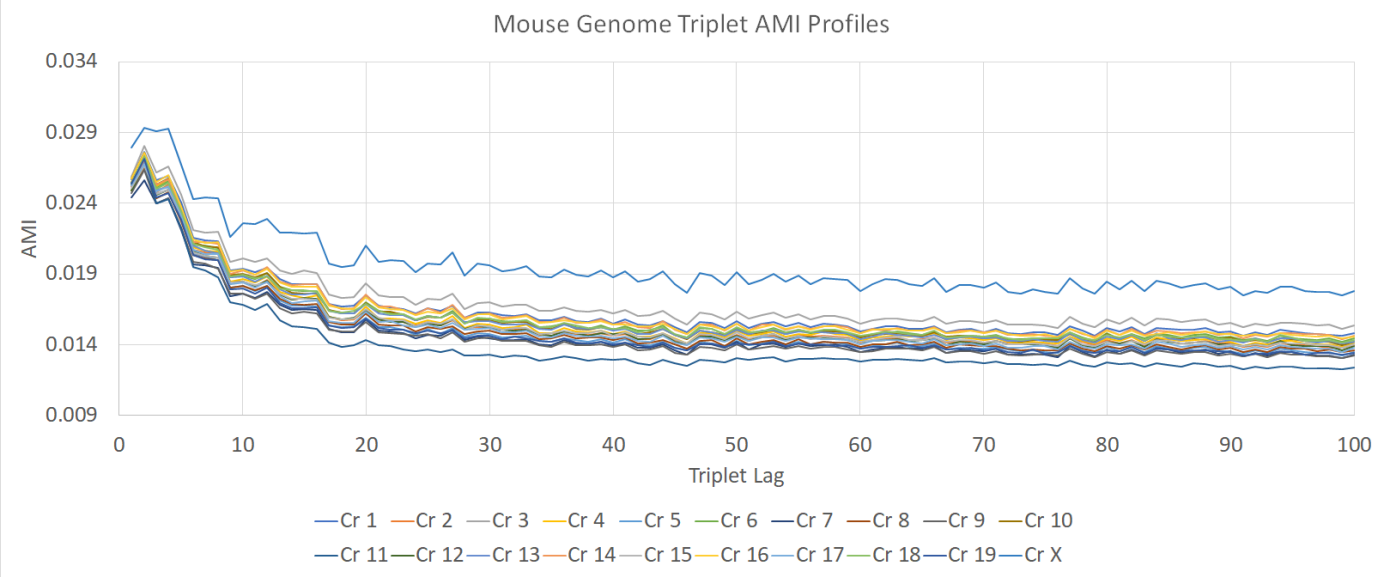

Figure 4.8: Non-overlapping triplet AMI profiles for all 20 chromosomes of the mouse genome.

4.2.2 An Analysis of the Base Repetition within Human Chromosomes

The answer to the question of what explains the appearance of this strong feature at base position 171, its sole relationship to triplet analysis, and its uniqueness to the human genome was elusive. An attempt was made to study the nature of repetition of base sequences in human DNA with a special focus on bases that are 171 positions apart. Knowing what actual repeats of DNA sequences can be found and how prevalent they were could help answer this question.

To accomplish this, a rather brute-force method was employed. Selected human chromosomes were searched for sequences which occurred at fixed distances apart that either matched or were close to matching. As is well-attested for algorithms such as BLAST which search for DNA sequences that match other DNA sequences, sometimes matches are not always exact. Sometimes a handful of bases are either missing, added, or substituted for other bases. This is generally due to the various types of mutation. Thus, a flexible definition of what constitutes a matching

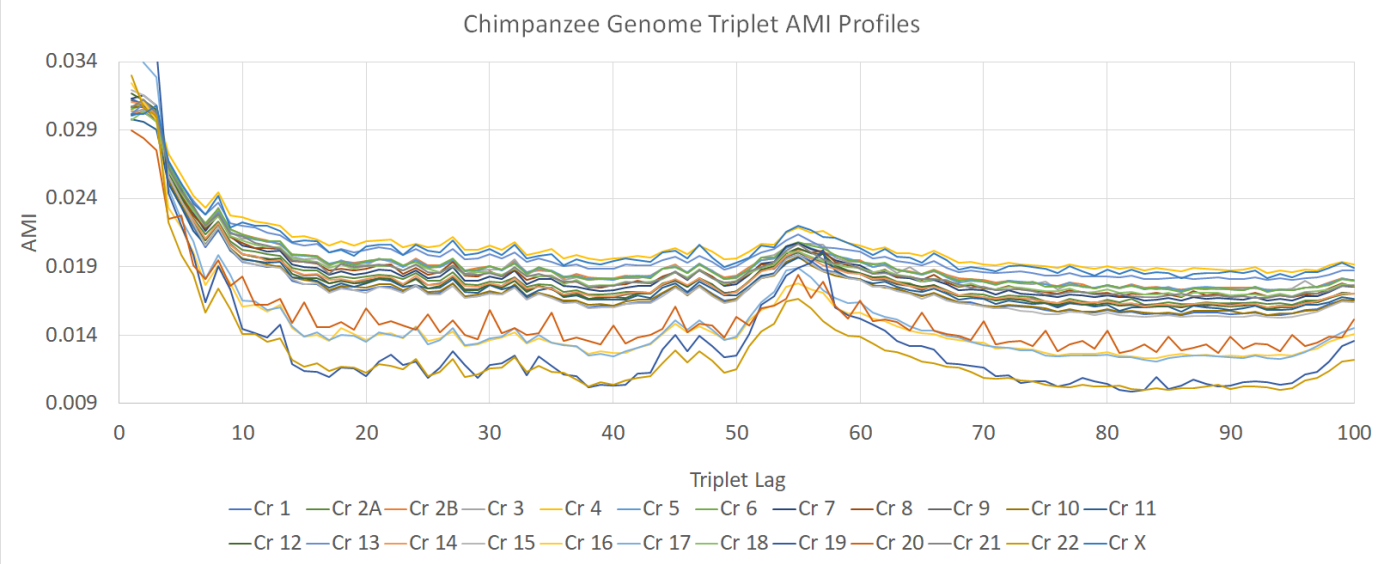

Figure 4.9: Non-overlapping triplet AMI profiles for all 23 chromosomes of the chimpanzee genome.

sequence for purposes of this exploration was developed. Since the *length* of matching DNA sequences was pertinent to this analysis, this was critical in determining when a discovered matching sequence was ended.

Different error tolerances were built into searching for repeats, allowing the algorithm to find repeats that were very close but not exact. Two types of error tolerance were tested. The first type of error tolerance defined a threshold value X, and when a sequence was being examined for a match, if X number of deviations were encountered, then the sequence was considered no longer a match. The second type of error tolerance defined a threshold value Y, and when a sequence was being examined for a match, if Y number of deviation were encountered *in a row*, then the sequence was considered no longer a match. The first method only allowed a fixed number of errors in a sequence regardless of size, whereas the second method allowed any number of errors as long as those errors were not sufficiently close. The drawback of the first approach was that a sequence that matched in large sections

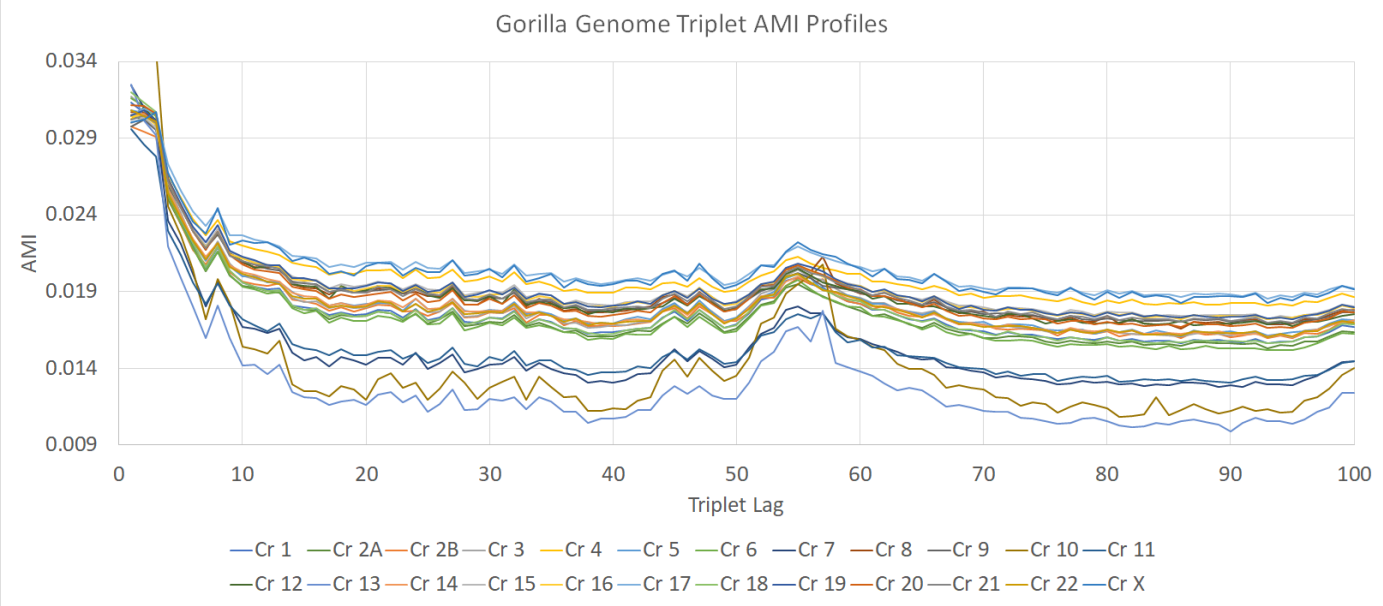

Figure 4.10: Non-overlapping triplet AMI profiles for all 23 chromosomes of the gorilla genome.

except for X+1 minor errors which were sufficiently far away from each other would not count as a match. The drawback of the second approach was that sequences that had many errors, as long as they were not adjoining, would count as a match. (For example, a sequence of AGAGAGAGAGAG and ACACACACACAC with even a Y tolerance of 2 would count as a *match* under the definition of the second approach.)

To avoid these difficulties, a mixed approach was devised which combined both error tolerance methods. The total error tolerance was eventually set to 10, and the tolerance for errors in a row was varied between 0 and 5. Thus, a repeated sequence would terminate if it totaled more than 5 errors in a row or more than 10 errors total. Arbitrarily chosen base separation distances of 135 and 225 were chosen as control groups by which to measure how many repeated sequences at 171 bases apart could be considered "more" than what is to be expected, when compared with 135 and 225. When testing for how many sequences separated by 171 bases could be identified that exhibited base repetition, only sequences that totaled more than 100

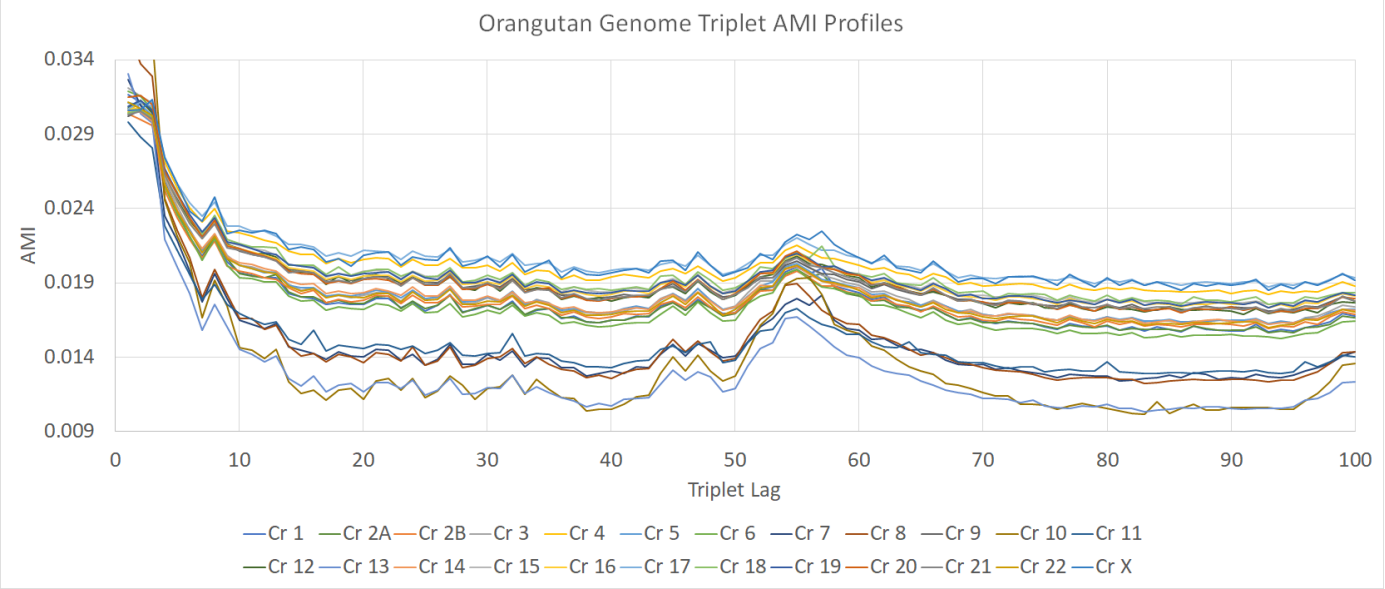

Figure 4.11: Non-overlapping triplet AMI profiles for all 23 chromosomes of the orangutan genome.

Chrom.	0 Err.	1 Err.	2 Err.	3 Err.	4 Err.	5 Err.	*135-Base*	*225-Base*
Chr. 1	4	32	41	43	45	47	29	52
Chr. 19	2	47	81	76	72	69	30	18
Chr. 22	8	142	172	183	186	188	45	14

Table 4.1: Number of sequences on each chromosome that were shown to correspond to a repeated sequence on the same chromosome 171 bases away for "in a row" error tolerances between 0 and 5. The average results for sequence repeats given base position separations of 135 and 225 are shown for comparison.

bases in length were considered so as to exclude trivial sequences from the results. These results are shown in Table 4.1. Human chromosome 1 is shown to indicate that some chromosomes did not have as much of a strong presence of repeated sequences separated by 171 bases, while chromosomes 19 and 22 are shown because these are the ones that exhibited the highest spikes in their triplet AMI profiles at the 57-triplet, 171-base position and, consequently, have the highest amount of repeated sequences separated by 171 bases.

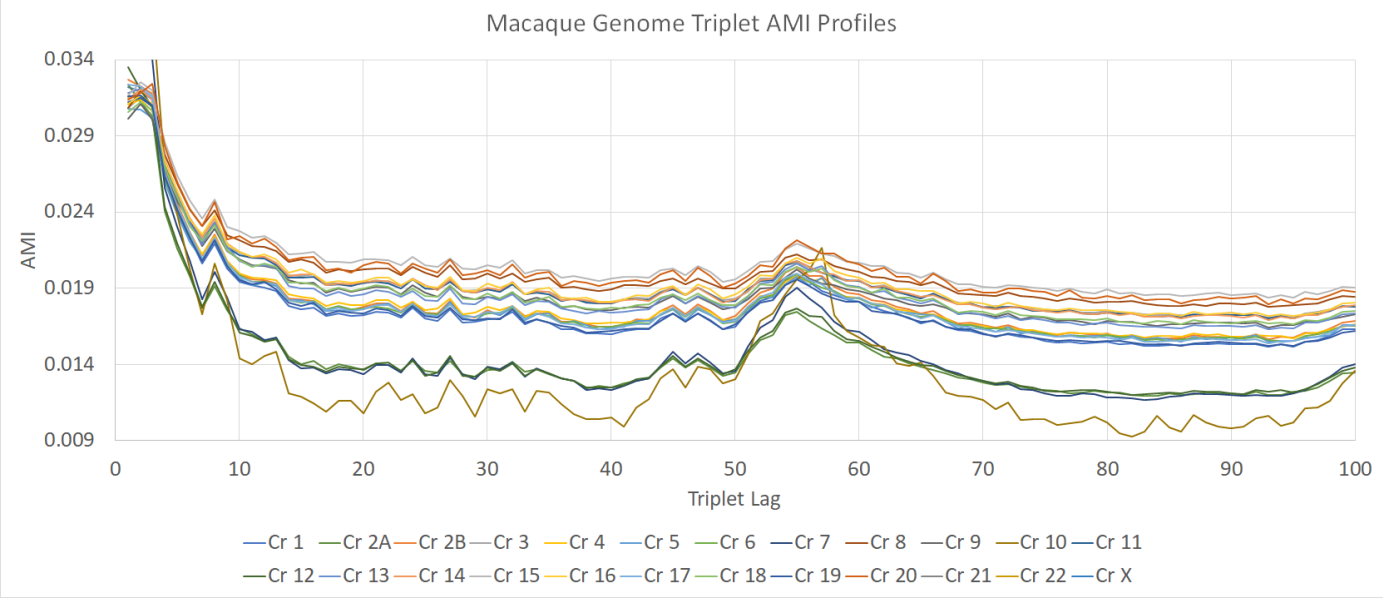

Figure 4.12: Non-overlapping triplet AMI profiles for all 23 chromosomes of the macaque genome.

These results indicated that a higher than incidental number of repeated sequences did occur for bases that were 171 positions apart. ("Higher than incidental" means that, when choosing some arbitrary length of bases, the number of repeats found were greater with a 171-base separation than with other arbitrary separation distances.) In fact, many repeated sequences *longer* than 171 bases were found, indicating that these repeats had to be "cyclical" or evidence of frequent tandem repeats. In other words, if a repeated sequence ended up being longer than 171 bases in length, that means that the sequence extended for 171 bases and then reached its repeated sequence *while still counting the repeated base distance.* For example, chromosome 22 had the longest length of repeated sequence found, which was 483 bases long no matter the error tolerance used.

4.3 Conclusion

While the observations that emerged from the study of triplet AMI profiles proved intriguing, no firm conclusions were reached as to the *cause* of these effects. Triplet AMI profiles were similar to standard AMI profiles, but they showed strongly that triplets which exist 57 triplet positions (171 bases) apart share more information than do other triplets much closer. Triplet AMI profiles were shown to be invariant under the reading frame selected, and the effects seen in the triplet AMI profiles were not observed to the same acute degree as in "duplet" AMI profiles calculated on base pairs, indicating the triplet nature of the code was contributing to the detection of these anomalous features. Additionally, these strong features were only present in studies of the human genome, and similar results were unobserved in any other genomes to the same degree, although weaker indications of the same feature did appear in similar primate genomes.

Of course, the source of these unique features is the presence of repeated sequences that occur 171 bases apart, some of these consisting of tandem repeats. The *reason* that such repeats are found was not evident. One possibility is that these frequent repeated sections could be due to the insertion of viral DNA into the human genome at regular intervals, similar in concept to what was identified by Mojica et. al. in prokaryotic organisms.[6] Future work to explore this question could consist of using more sophisticated sequence alignment techniques that could find the specific instances of repeated sequences causing these features. This work would then involve searching through annotations on the human chromosome to attempt to discover a connection between the exact location of repeats the presence of specific biological entities at those locations.

Additionally, if the triplet AMI profiles proved to be more indicative of the information structure of DNA than standard AMI profiles, they could be employed as an extension of the work of Chapter 3 to determine whether triplet AMI profiles could form the basis of a reliable DNA compression and recovery technique. Though some preliminary tests were attempted to that end, no significant deviation from the results reported in Chapter 3 were observed, and the results of that chapter along with this one don't indicate a strong possibility that a significant improvement in compression would be obtained.

Chapter 5

Prediction of Coding and Noncoding Regions via Machine Learning

5.1 Introduction

The DNA code contains the blueprint to construct every protein that is used by the cells of an organism. For this purpose, the sequences of DNA bases contain highly organized information, but not all sequences found on chromosomes are used to code for specific proteins. Some regions of DNA, referred to as noncoding regions, have other functions or no apparent function at all. It is postulated that these two types of DNA regions, coding and noncoding regions, may have different underlying structures present with regard to the information they contain. The aim of this chapter is to examine whether the AMI profile has a substantial relationship to one of the most basic facts about a DNA sequence: whether it is a coding or noncoding sequence. Through biological "wet-lab" experimentation, the locations and length of coding and noncoding segments are well known and catalogued for many organisms; however, for these purposes of investigation, it needs to be determined if this classification can be ascertained only with reference to the information structure of the DNA strand itself, modeled by the AMI profile alone, without reference to a direct observation of its related biological processes.

This chapter seeks to answer four main questions about the ability of the AMI profile to differentiate between coding and noncoding regions. First, can the AMI profile be used to make predictions about the coding structure of a DNA sequence? In other words, is the amount of structure-related information that the AMI profile is able to quantify about DNA sequences sufficient for making this determination? Second, if the AMI profile does contain enough information to make moderately accurate predictions about DNA sequences, how much information must the AMI profile contain to make these predictions with reasonable accuracy? AMI profiles can be calculated to contain mutual information about bases that are separated by a large distance, but how much of this mutual information is actually needed to make reasonably accurate predictions? For example, is mutual information about bases separated by 100 or more base positions relevant enough to the structure of a coding region that its inclusion would improve the accuracy of the prediction methods? Third, how does the prediction of coding regions compare to noncoding regions? Specifically, it needs to be known if noncoding regions, which presumably have less overall structure than highly structured coding regions, have enough discernible structure to allow for accurate classification when compared to coding regions. Finally, if the AMI profile is sufficient for DNA-sequence classification, which binary classification implementation is the best suited for this purpose? This study will determine whether a multilayer perceptron neural network (MLPNN) or a support vector machine (SVM) is a better classifier for predicting the coding status of a DNA sequence based solely on its AMI profile.

5.2 Classifiers and Data Sets Used

This chapter examines whether two commonly used binary classifiers, a MLPNN and a SVM, can predict whether a DNA sequence is from a coding or noncoding region based solely on its AMI profile. The results of both of these methods are compared, and it is determined which method is best suited to this situation. The success of both of these methods is evaluated to determine whether the AMI profile of a DNA sequence contains sufficient information about the structure of its information to make this determination.

To train both of these classifiers, a training data set and a testing data set were developed. For this evaluation, the genome of the bacterium *Escherichia coli* (*E. coli*) was used to develop these data sets on which to evaluate the classifier implementations. The *E. coli* genome (NC_000913.3) was selected because of its simplicity; since *E. coli* is prokaryotic, it only contains coding and noncoding regions, as opposed to most eukaryotic organisms that have more complex segmentation in their genomes (such as the presence of introns, which are sub-units of coding regions that are removed before translation and are not used directly for protein coding). The *E. coli* genome was also selected because it is well-known and commonly used to evaluate numerous bioinformatics methods, such as in [1].

A set of training data and a set of testing data were developed as follows: Coding and noncoding regions were selected in order from the beginning of the *E. coli* genome with the stipulation that each region had to include at least 100 bases. This restriction was implemented to ensure that the AMI profiles calculated from these regions were truly representative of their information structure and not affected by anomalies. It was determined that each AMI profile would be calculated with a maximum lag of 50 bases, based on a visual inspection of typical AMI profiles

of the *E. coli* genome, which showed that the AMI values beyond about 50 bases of lag were generally redundant of what was already present in the first 50 bases of lag. By selecting only regions that contained 100 or more bases, it was ensured that the AMI profile would at least be averaging over one whole "window-shift" in each coding and noncoding region, since the AMI of bases up to 50 positions apart was included in the AMI profile calculations. The AMI profile for each of the first 400 suitable regions in the *E. coli* genome was included in the training set and labeled either "coding" or "noncoding," and the next 400 suitable regions in the *E. coli* genome were included in the testing set and also appropriately labeled. Due to the abundance of coding regions in the *E. coli* genome compared to noncoding regions, each data set contained approximately 70% coding and 30% noncoding regions.

The MLPNN implementation used was the default MATLAB implementation for a shallow, feed-forward neural network, using the function *feedforwardnet()*. The MLPNN was given the AMI-profile vector as its input, produced a single output, and contained one layer of hidden neurons with sigmoidal activation functions. The Levenberg-Marquardt optimization algorithm was used to train the weights and bias terms of the MLPNN because it was recommended by the MATLAB package documentation and because it performed more efficiently than the traditional gradient-descent method in preliminary tests. The MLPNN training was conducted by introducing the training data set to the MLPNN repeatedly in epochs of randomized pattern order until one of the following three conditions was met: the number of epochs reached a maximum of 4,000, the Levenberg-Marquardt gradient value reached a minimum of 1e-10, or the number of successive iterations without a validation failure reached 200. Since the output of a general MLPNN is a real number, and the classification required is binary, coding regions were mapped to a value of 1 and noncoding regions to a value of 0. To determine the prediction of the

MLPNN, its output was passed through a unit step function centered at 0.5. Thus, all outputs greater than 0.5 were considered predictions of coding regions, and those less than 0.5 were considered predictions of noncoding regions. The range of values $[0.4, 0.6]$ was considered a "window of uncertainty" where predictions may not be as accurate, and the effects of this are discussed below.

The SVM implementation used was also the default MATLAB implementation for an SVM using the *fitcsvm()* function. The SVM was given the AMI-profile vector as its input and produced a single, binary output. The default MATLAB standardization algorithm for SVM inputs was used to standardize the input predictors by their mean and standard deviation as was recommended by the MATLAB reference documentation to make the SVM less sensitive to the scale on which the input vectors were measured. Multiple kernel functions for the SVM were tested, including a linear, a quadratic, and a Gaussian kernel function. The results of the linear and quadratic kernels are presented below, but the Gaussian results were abnormal, and the Gaussian kernel did not seem to be well-suited to this classification problem, so the Gaussian kernel results are omitted.

5.3 Classifier Results

Both the MLPNN and SVM classifiers were evaluated with both the training data set and the testing data set. Various parameters of each classifier as well as the length of the AMI-profile input vectors (i.e. the maximum AMI-profile lag) were varied to observe their effect on the classification results. The results of each of these adjustments in terms of the errors produced in the classification are presented below.

5.3.1 AMI Profile Size

The size of the AMI-profile input vector provided to the classifier intuitively bears some relationship to the expected accuracy of that classifier, as having only a one- or two-length AMI vector does not provide enough overlap to predict the structure of an entire 100+ base region of a DNA sequence. Conversely, AMI-profile values tend to decrease as the lag increases, meaning that as bases become more separated on the sequence, they contain, in general, less mutual information about each other, such that bases sufficiently far away will not affect the determination of information structure. The amount of AMI lag needed in order to make predictions with reasonable accuracy was studied for both classifiers. The size of the AMI-profile input vector was varied between 3 (the size of one triplet in a DNA code) and 50 (the maximum AMI-profile vector length contained in the training data set) to train both the MLPNN and SVM. The accuracy of the MLPNN and the SVM trained with only the corresponding amount of lag in terms of absolute errors was examined.

For the MLPNN, a nominal value of 100 neurons was selected for evaluation due to the maximum size of 50 for the AMI-profile input vectors. Initially for the MLPNN, AMI-profile lag values were tested in multiples of 5, as can be seen in Figure 5.1. However, the results of these tests indicated little variation for both the training and testing data sets on the MLPNN prediction accuracy. A second test was conducted which tested AMI-profile lag values in multiples of 3, since it is known that DNA is structured as a triplet code. These results, seen in Figure 5.2, produced a slight indication that at least an AMI-profile lag of 12 was needed to be able to predict the training data set well. In other words, AMI-profile lags greater than 12 did not have any noticeable effect on the accuracy of the MLPNN results. It can be seen from the results in Figures 5.1 and 5.2 that the accuracy of the

MLPNN for predicting the coding or noncoding nature of DNA segments based on the AMI profile is reasonably accurate regardless of the AMI-profile lag value used, as the MLPNN results for the training data set never exceeded 15% error for 400 predictions (excluding the single high-error result for an AMI lag of only 3, which is intuitively nonideal). Likewise, the MLPNN results for the testing data set never exceeded 27% error for 400 predictions. Overall, the MLPNN seemed to be fairly invariant for changes in the size of the input AMI-profile vector.

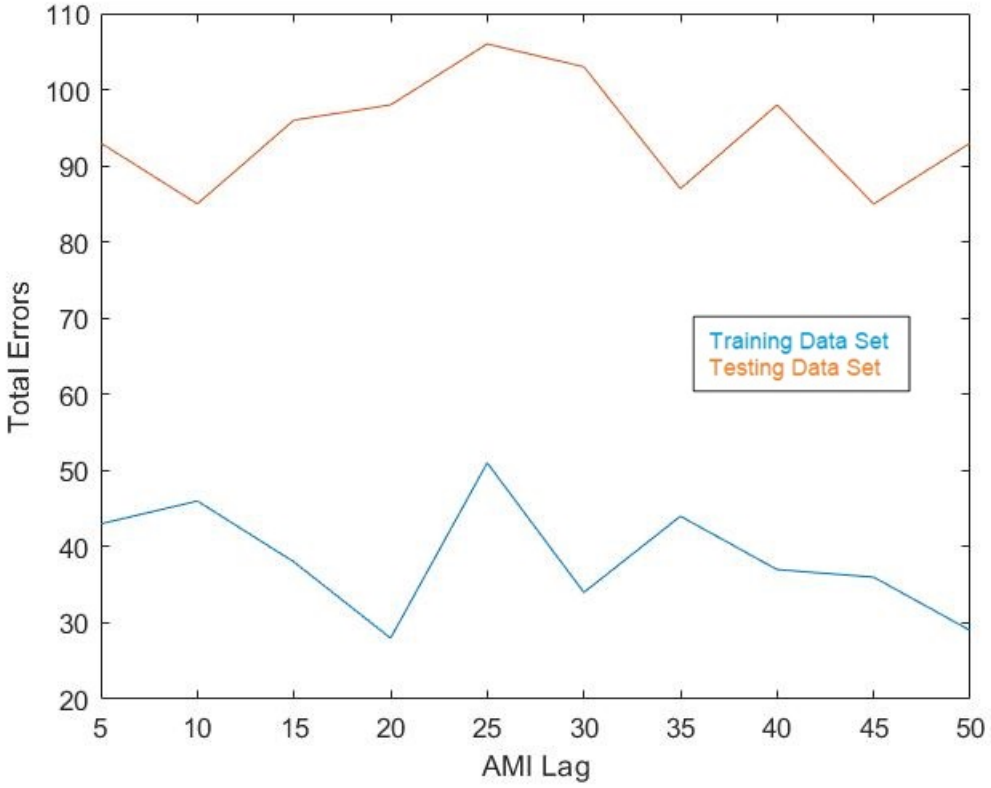

Figure 5.1: Total errors out of 400 input patterns while varying the input AMI-profile lag by multiples of 5 for a 100-neuron MLPNN shown for both training and testing data sets.

The SVM implementation was more sensitive in terms of prediction accuracy to the length of the input AMI-profile vector. It can be seen from Figure 5.3 that the SVM with a linear kernel produced its best results with a minimum AMI lag

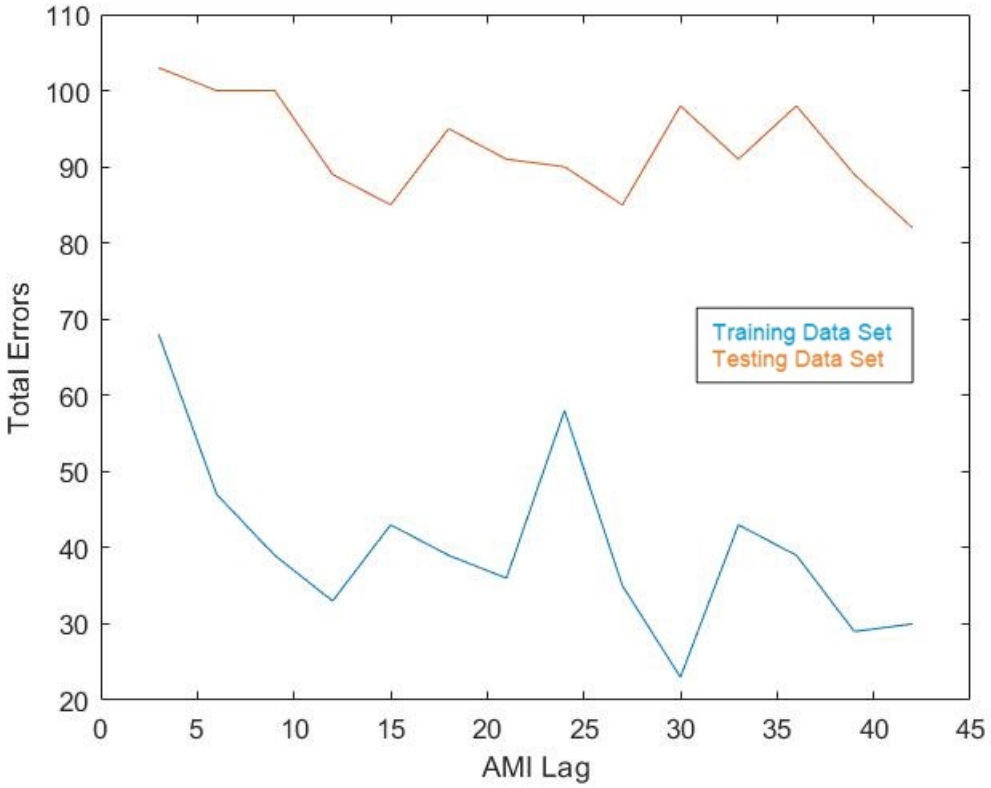

Figure 5.2: Total errors out of 400 input patterns while varying the input AMI-profile lag by multiples of 3 for a 100-neuron MLPNN shown for both training and testing data sets.

between about 15-21, especially for the training data set but also roughly for the testing data set. In fact, as more AMI lag is included, the prediction accuracy for the testing data set increases slightly, showing that having "too much" AMI-profile lag possibly overfits the training-set data and makes the prediction of the testing set slightly less accurate. When evaluating the SVM with the quadratic kernel, it was found that the quadratic-kernel SVM greatly outperforms the linear-kernel SVM for the training data set but underperforms the linear-kernel SVM for the testing data set, as the results in Figure 5.4 show. This is a clear indicator that the SVM with a quadratic kernel for sufficiently large values of AMI lag (around 33 and above) overfits the training data set, thus making it less accurate to predict sequences not

found in the training data set. On average, the linear-kernel SVM with AMI lag greater than 18 had an error rate of 9.5% for 400 predictions of the training data set and an error rate of about 21.5% for 400 predictions of the testing data set. By comparison, the quadratic-kernel SVM with AMI-lag greater than 18 had an average error rate between 1% and 2.5% for the training data set depending on how many AMI-profile lags were used. However, it had an average error rate of about 24.8% for the testing data set. Since the linear-kernel SVM performs best on the testing data set and avoids overfitting for AMI-profile lags between 15-21, it was determined that the SVM with a linear kernel produces the best results compared to both the quadratic-kernel SVM and the MLPNN classifier with a reasonable AMI-profile lag of about 18 base positions necessary for optimal accuracy.

5.3.2 Prediction of Noncoding Errors

It is known that DNA has a highly organized triplet structure that is used by the cell's translation mechanism to convert the coding regions of DNA into proteins. Because of this, it is postulated that DNA has a more highly ordered structure in coding regions than noncoding regions, and it is hypothesized, then, that more errors should occur when predicting noncoding regions than coding regions. Thus, the number of noncoding-region errors (NCREs) resulting from a noncoding region being incorrectly predicted as a coding region were examined.

As can be seen from the results for the MLPNN in Figure 5.5 and the SVM in Figure 5.6, NCREs certainly account for more of the total errors than do errors in coding regions. It can be seen that, on any particular test, the number of NCREs is always between 55% and 95% of the total number of errors; it always exceeds 50% of the total errors, even if only slightly. It can also be seen from these results that NCREs are overrepresented regardless of the AMI-profile lag at the input, as the

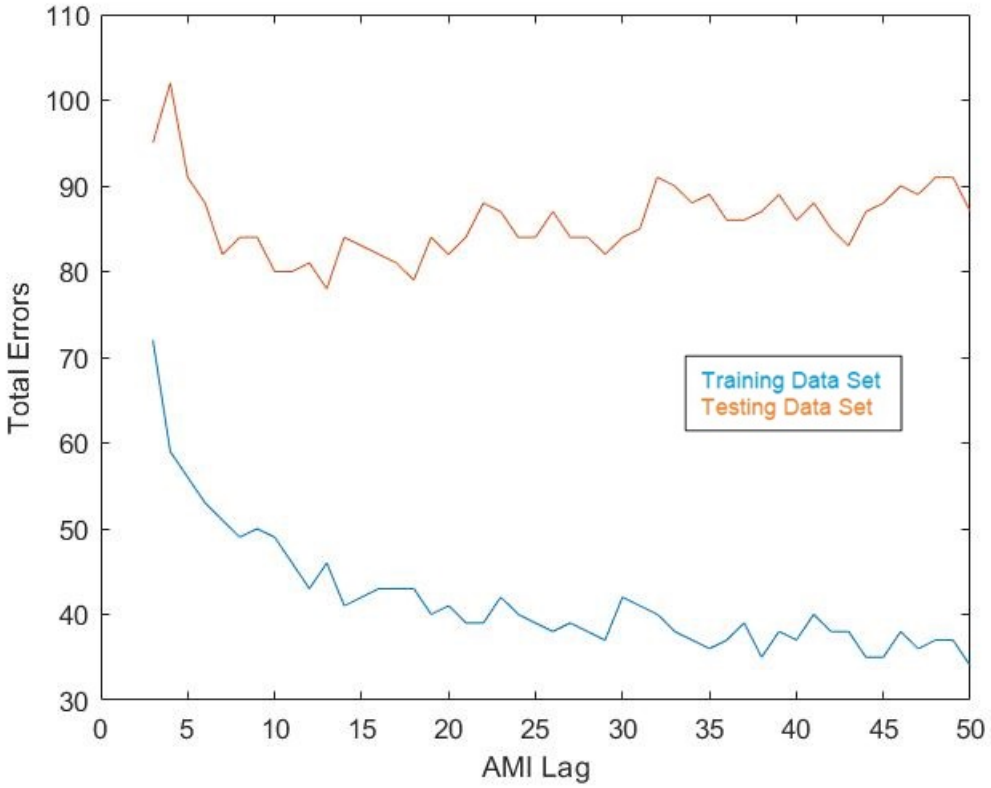

Figure 5.3: Total errors out of 400 input patterns while varying the input AMI-profile lag for the SVM with a linear kernel shown for both training and testing data sets.

results are mostly invariant over many AMI-profile vector lengths. However, the results of Figure 5.6 do provide another piece of evidence to confirm that an AMI-profile lag of about 18 is ideal for the SVM classifier, because having a sufficiently large AMI-profile lag reduced the number of NCREs relative to the total number of errors. When a sufficiently high AMI-profile lag of 18 was used with the SVM, the number of NCREs versus coding-region errors was closer to even. When comparing the MLPNN and the SVM, it can be seen that, for the testing data set, the MLPNN averages about 67% NCREs, while the SVM with a sufficient AMI-profile lag of greater than 18 averages about 61% NCREs.

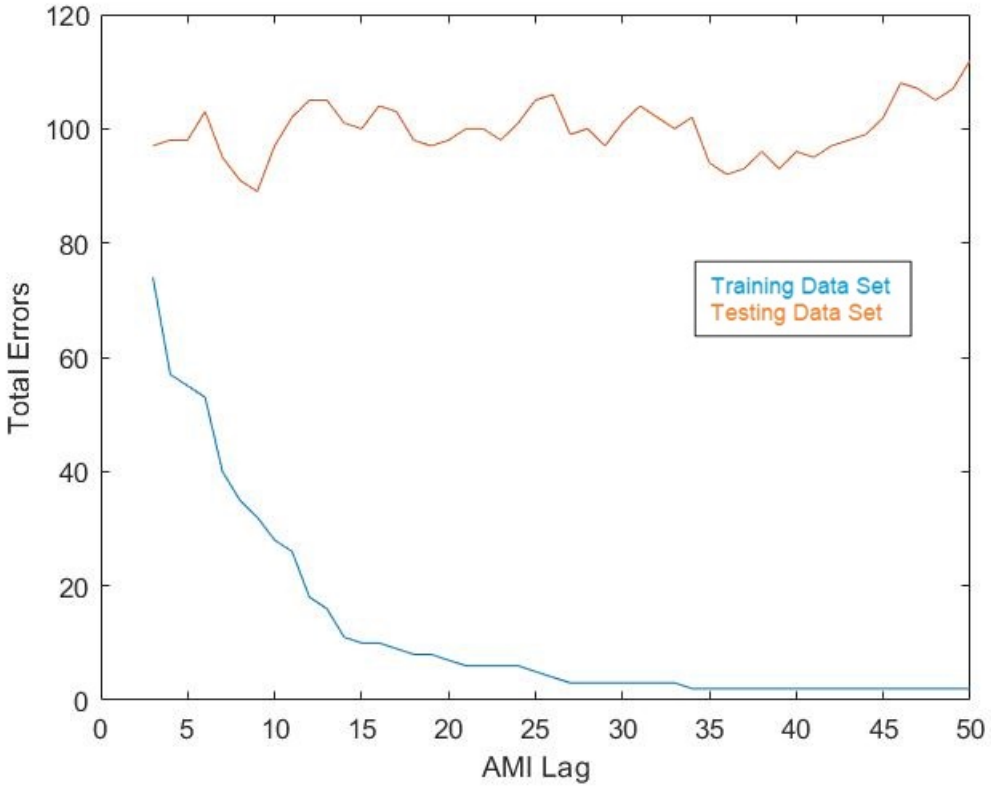

Figure 5.4: Total errors out of 400 input patterns while varying the input AMI-profile lag for the SVM with a quadratic kernel shown for both training and testing data sets.

In general, for both the MLPNN and SVM, the testing data results represent a lower percentage of NCREs than the training data set does, regardless of AMI lag. This is due to the fact that, when testing a classifier on a testing data set that it has not seen before, more errors will result, and thus when the MLPNN and SVM were applied to the testing data set, more coding errors were encountered than with the training data, and thus the number of NCREs represented a lower percentage of the total number of errors due to the increased number of coding-region errors.

There are three significant reasons that can explain the higher percentage of NCREs compared to coding-region errors. The first is that noncoding regions are less represented on the *E. coli* genome, and this was reflected in the construction of

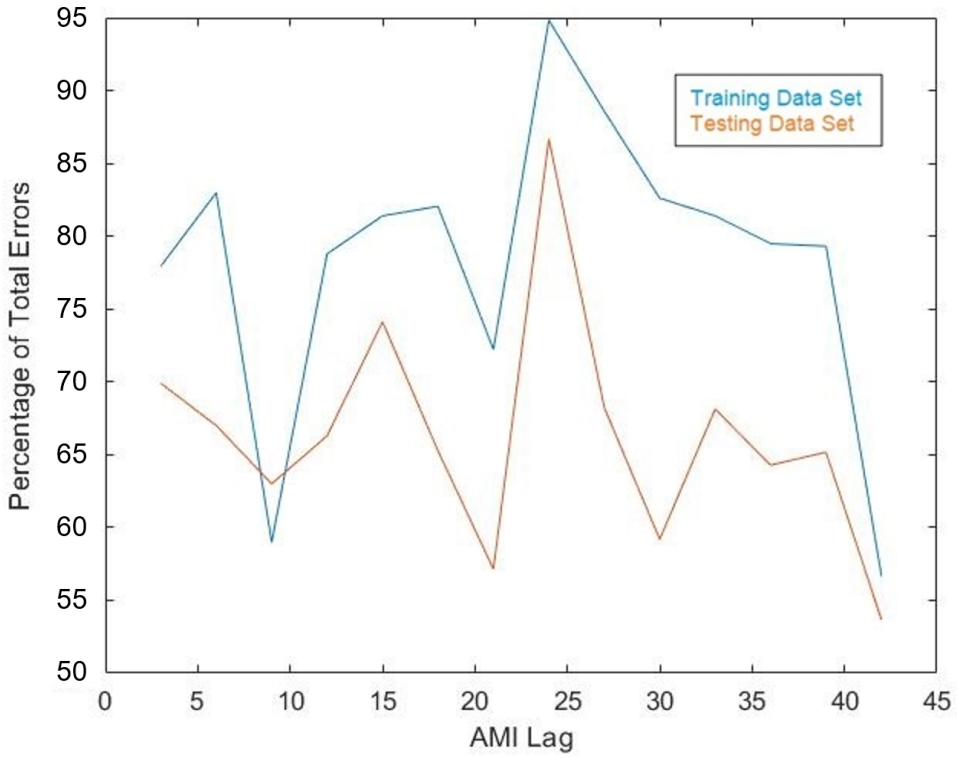

Figure 5.5: Percentage of total errors with the 100-neuron MLPNN that accounted for results where a noncoding region was incorrectly predicted as a coding region, shown for both training and testing data sets.

the training and testing data sets as discussed previously. Because of the underrepresentation of noncoding regions in the training data, the classifiers may not have predicted noncoding regions as well as coding regions since they had more coding regions on which to train than noncoding regions. Secondly, DNA is structured in a double-helix structure, meaning that for every DNA sequence there are two strands, one of which contains a sequence of bases and the other contains its opposite. However, only one side of a strand is used for protein coding, so while one side of the DNA strand may be noncoding, it could be the case that this noncoding region is actually the reflection of a coding region, meaning that its structure would be highly similar to that of the coding region it reflects. However, it is assumed that

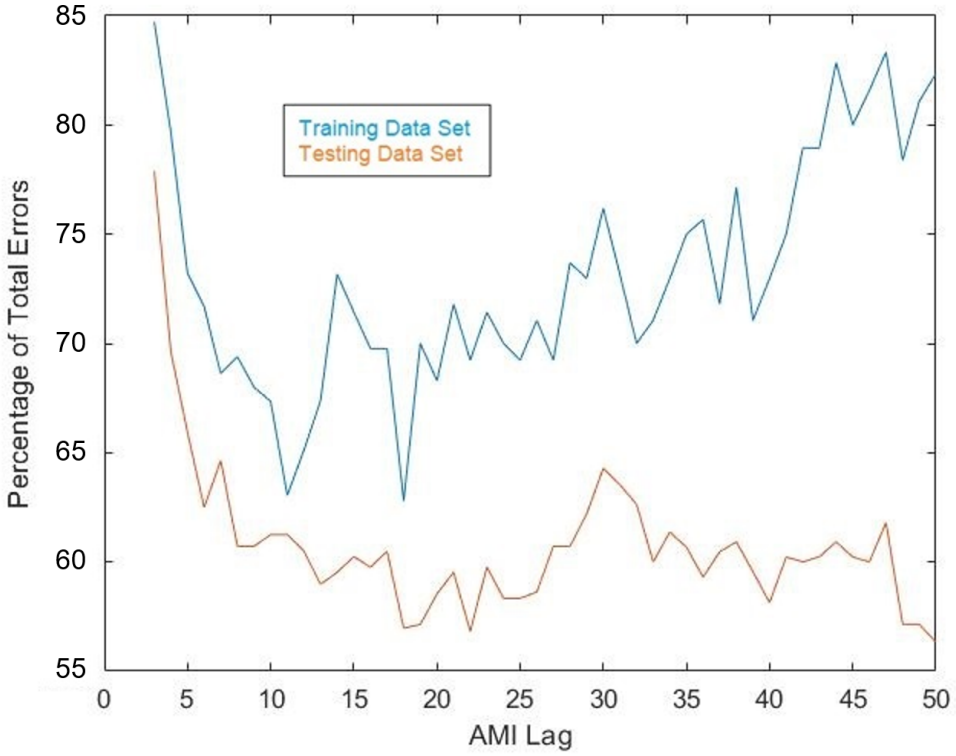

Figure 5.6: Percentage of total errors with the linear-kernel SVM that accounted for results where a noncoding region was incorrectly predicted as a coding region, shown for both training and testing data sets.

this specific factor plays little role in the results of the classifiers since the annotated *E. coli* data set used to construct the training and testing data sets was organized in such a way as to minimize this possibility. Finally, the hypothesis mentioned previously, that noncoding regions are overall less structured and exhibit more "random" characteristics as compared to coding regions, is consistent with these results and could explain why coding regions are easier for both the MLPNN and SVM classifiers to predict.

5.3.3 Tuning the MLPNN

Lastly, a brief examination was made of MLPNN parameters to observe whether or not the number of neurons affected the error rate and how much of the error rate for the MLPNN was caused by predictions falling into the "uncertainty window" of [0.4, 0.6]. It can be seen from Figure 5.7 that only about 25% of total errors resulted from the uncertainty window for the training data set and only about 19% of total errors resulted from the uncertainty window for the testing data set. This result shows that predictions of the MLPNN that fell within the uncertainty window did not have an out-sized detrimental effect on the resulting accuracy of the MLPNN, regardless of the size of the AMI lag. From Figure 5.8, it can be seen that the number of neurons of the MLPNN does not largely affect its accuracy. In fact, for the testing data set, more neurons appear to slightly increase the total number of errors. Therefore, the nominal value of 100 neurons used for the evaluations above was sufficient for judging the performance of the MLPNN and appears to be the best balance between accuracy and unnecessary computational complexity. Thus, these two minor concerns for the MLPNN were shown to not have affected the results in a major way that would drastically alter the conclusions should the MLPNN be instantiated slightly differently. Overall, the results above are indicative, in general, of the performance of an MLPNN in this context.

5.3.4 Which Classifier is Better?

When all results are taken into consideration regarding the percentage of total errors produced on the unseen data in the testing data set, the size of the AMI-profile lag required, and the mitigation of high relative numbers of noncoding-region errors, the linear-kernel SVM implementation appears to be the

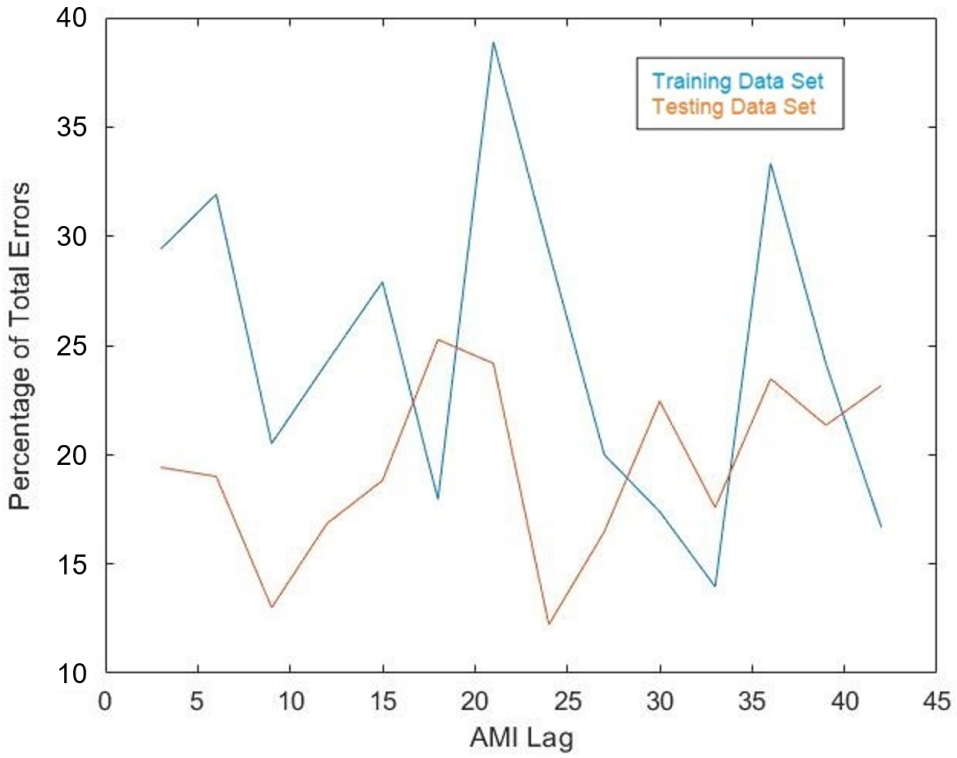

Figure 5.7: Percentage of total errors that accounted for prediction results that fell within the "window of uncertainty" for the 100-neuron MLPNN shown for both training and testing data sets.

best suited for classifying DNA sequences by their AMI profiles. The linear-kernel SVM outperforms the quadratic-kernel SVM and all instances of the MLPNN. For the MLPNN, 100 neurons was a reasonable balance between accuracy and computational complexity that represented the general performance of an MLPNN for classifying AMI-profile vectors. On average, regardless of the maximum AMI-profile lag, the MLPNN exhibited around a 24% error rate where NCREs were 67% of the total errors on average. The SVM, though, exhibited a lower rate of errors, about 21.5% on average over the value of AMI-profile lag used, where NCREs were 61% of the total errors on average. Thus, the linear-kernel SVM is more accurate, on average, than the MLPNN regardless of the length of AMI-profile

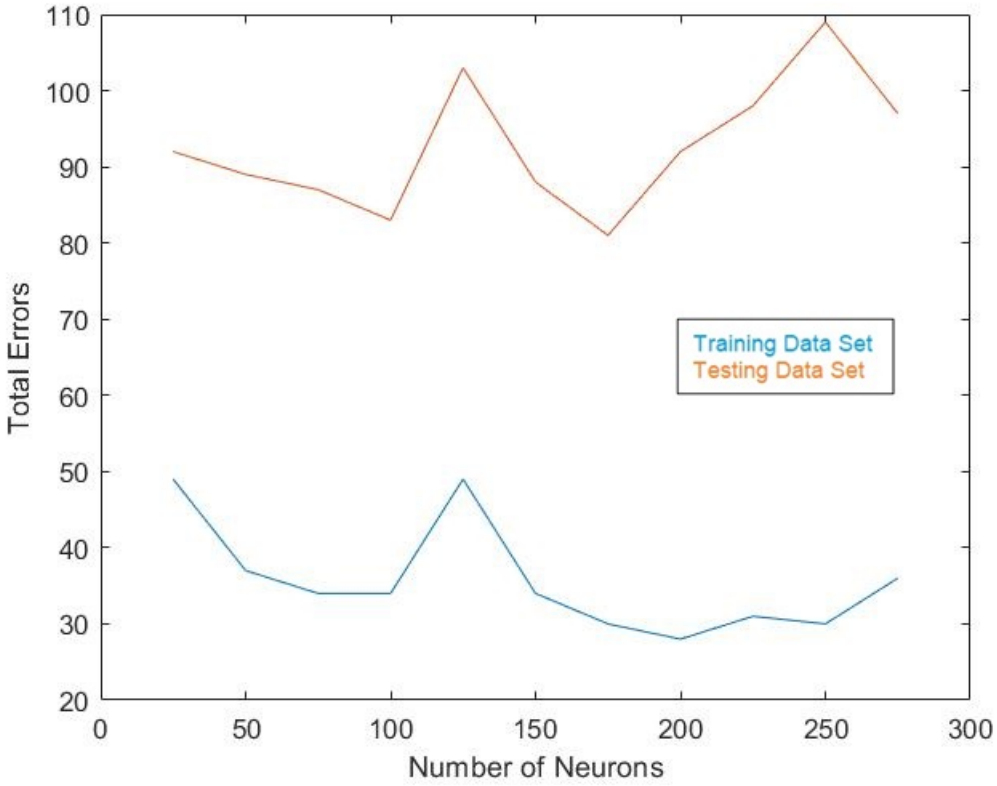

Figure 5.8: Total errors out of 400 input patterns when varying the number of neurons in the MLPNN using the ideal input size of 18 AMI lags.

vector used as input, and the linear-kernel SVM is slightly closer to an even balance between coding and noncoding errors than the MLPNN as regards the total percentage of NCREs to total errors when an AMI-profile lag of about 18 is used. For the problem of predicting coding and noncoding regions of the DNA sequence, the SVM appears slightly better suited based on the results from the genome of the bacterium *E. coli*. Based on these results, it seems that the AMI profile does contain sufficient information to make a reasonably accurate prediction about the structure of the information present in a DNA sequence, as the reasonable success of both the MLPNN and SVM implementations demonstrate.

5.4 Conclusion

In this chapter, two types of common binary classifiers, an MLPNN and an SVM, were implemented and used to test the hypothesis that the AMI profile for a DNA sequence can be used to predict whether the sequence is coding or a noncoding. In broader terms, this was used to verify whether the AMI profile is sufficient to quantify the underlying information structure of DNA sequences and whether this can be related to biologically relevant features of such sequences. As a result, the AMI profile seems to be a sufficient metric to quantify the information structure of DNA sequences in that it can predict at least a basic fact about its biologically relevant structure. It was also discovered that, for the two classifiers implemented, the minimum length of the AMI-profile vector needed to produce the most accurate results was a lag of about 18 base positions. Thus, the AMI profiles that performed the best consisted of mutual information of bases up to 18 bases away; more mutual information content than that did not seem to improve accuracy in any substantial sense. It was observed that errors predicting sequences from noncoding regions, in general, accounted for a higher percentage of total errors than errors for sequences from coding regions. Finally, since it was confirmed that the AMI profile can be used to predict whether a DNA sequence is from a coding or noncoding region with reasonable accuracy, it was determined based on the accuracy of results that a SVM with a linear kernel was the best-suited classifier for this context.

Chapter 6

Classification of DNA Sequence Regions via Principal Component Analysis

6.1 Introduction

The theory that has driven this investigation from the start has relied on the assumption that DNA sequences can be divided or segmented into informationally significant regions. Since DNA can be compared analogously to a language, the bases can be likened to letters in an alphabet, triplet codes to words, and larger segments to paragraphs. It was the search for these "paragraphs" which engendered this exploration.

As has been previously stated, chromosomes contain regions which are known as "coding regions," those which are responsible for protein coding within the cell, and "noncoding regions," those which fall outside of coding regions which have either another function, an unknown function, or no function at all. The attempt to predict missing bases was an attempt to discover which bases may be more informationally significant than others in the hope that this would reveal which bases could serve as "hinge points" for dividing the DNA sequence into natural regions. Instead of searching for informationally significant regions from a microscopic perspective, by considering the qualities of individual bases and using

those to try to determine informationally significant regions, an alternative approach was attempted. In this approach, regions of the DNA sequence are considered *qua* regions and are analyzed on the basis of their qualities determined for the entire region. The AMI profile was again used as a tool to quantify the properties of various regions of a DNA sequence.

6.2 Principal Component Analysis

In order to ascertain whether the AMI profile contains information specific enough about a region in order to allow it to be quantified, and furthermore whether those regions were differentiable from each other, Principal Component Analysis (PCA) was used to see if certain regions would cluster together based on their AMI profiles.

PCA is a method of dimensional reduction, meaning that it is used to take highly dimensional vector data and project it into a lower dimensional space. Doing this can yield a projection of the original data that emphasizes its most important components, i.e., the components of the data exhibiting the highest variance. These components are identified first by finding the covariance matrix between all vector components and then by computing the eigenvectors of the covariance matrix. Projecting data into a lower-dimensional space will inevitably result in the loss of some information contained in the data, but the method of PCA which chooses to retain the components exhibiting the highest variance seeks to preserve as much critical information as possible about the data set. In other words, PCA preserves the strongest signals present in the data set.

When using PCA to compare the AMI profiles of various DNA regions, preserving their most significant information and hopefully reducing noise, the hypothesis was that informationally similar regions of the DNA sequence might

appear closer together in a lower dimensional space, indicating similarities between the information content of the bases contained in those DNA segments.

6.3 Arbitrarily Selected Human Chromosome Regions

6.3.1 Process

Telomeres are the regions on the end of chromosomes, and they can be known for containing DNA that is variant from the DNA contained in other, more central regions of chromosomes. Additionally, such as with human chromosome 9, telomeres can be hard to sequence, and thus several chromosomes in various assemblies of the human genome do not record bases for lengthy segments within the telomeres. (In human chromosome 9, for instance, the first 10,000 bases in the chromosome are undetermined.) Thus, when selecting regions in which to extract segments for PCA, telomeres were avoided. To accomplish this, a starting position of 17,360,000 bases within each chromosome was chosen at which to start collecting regions for analysis. (This number was ascertained by trial and error after examining various chromosomes.)

The AMI profile of successive regions on a chromosome of an arbitrary number of bases were used as vectors on which PCA was performed. Initially, PCA was performed with 250 AMI-profile vectors that came from 2,000-base-long segments, with 30 elements of lag in the AMI-profile vector. Stating with the previously determined starting position that was well within the chromosome being considered, a group of 2,000 sequential bases was selected as a region, and its AMI profile was calculated. This was continued moving sequentially down the chromosome until 250 regions has been selected and their AMI profiles calculated. The AMI profiles of each region were supplied to the PCA algorithm, which projected them down to a

two-dimensional space, preserving the two most significant components of the AMI-profile data. Longer regions consisting of 10,000 bases each, for which a single AMI profile was calculated, and 50,000 bases each, for which also a single AMI profile was calculated, were also tested. The longest 50,000-base regions were also tested with AMI profiles that extended to a lag of 200. A total of 500 regions, and thus 500 AMI-profile vectors, were generated for these experiments.

6.3.2 Results

The PCA that was performed with AMI-profile vectors obtained from DNA regions consisting of 2,000 bases showed almost no clustering or other identifiable data features for any human chromosomes. The behavior of the PCA even varied for each chromosome, yielding no identifiable consistency between results either. Figures 6.1 to 6.5 show these PCA results on a representative sample of five chromosomes from the human genome. Only one general "cluster" is seen in each of these results; some chromosomes show all of their representative regional vectors as being tightly clustered (such as chromosome 15 in Figure 6.3) while others show relatively loose clusters (chromosome 22 in Figure 6.4) and yet others exhibit a cluster with several outlier vectors. The values on the axes of the plots for all PCA results represent projected values of AMI profiles into two dimensions and thus do not carry substantial meaning as specific quantities. Rather, the relevant behavior of each PCA result is qualitative.

Since no discernible clustering or other unique behavior was observed for regions that only consisted of 2,000 bases, longer regions consisting of 10,000 bases per region were examined. Most chromosomes still did not exhibit any clustering behavior. However, human chromosomes 13, 14, and 15 produced PCA results that

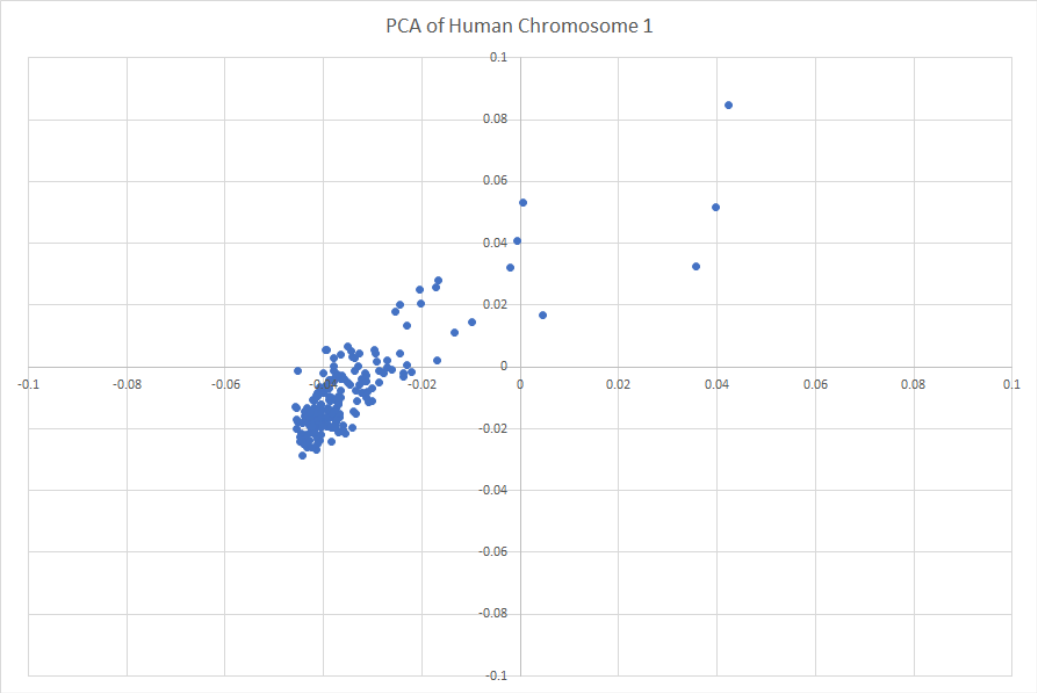

Figure 6.1: Two-dimensional PCA results for human chromosome 1 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 2,000-base length.

seemed to show some slight resemblance to clustering behavior. Figures 6.6 to 6.8 demonstrate this development.

Focusing on these chromosomes (13, 14, and 15), the parameters of the PCA were then varied to determine if the slight clustering effect seen in the results could be accentuated. This included increasing the base length of the regions, increasing the among of lag included in each AMI profile (and thus the dimensionality of the PCA input vectors), and the number of data points considered. The combination of parameters that seemed to exhibit the most noticeable clustering effect were found to be regions of 50,000 bases in length, AMI-profile vectors with a lag of 200, and 500 total vectors in the input dataset for the PCA. Other chromosomes were then also tested with these parameters, and chromosomes 13, 15, and 19 showed the most promising results. These can be seen in Figures 6.9 to 6.11. However, other

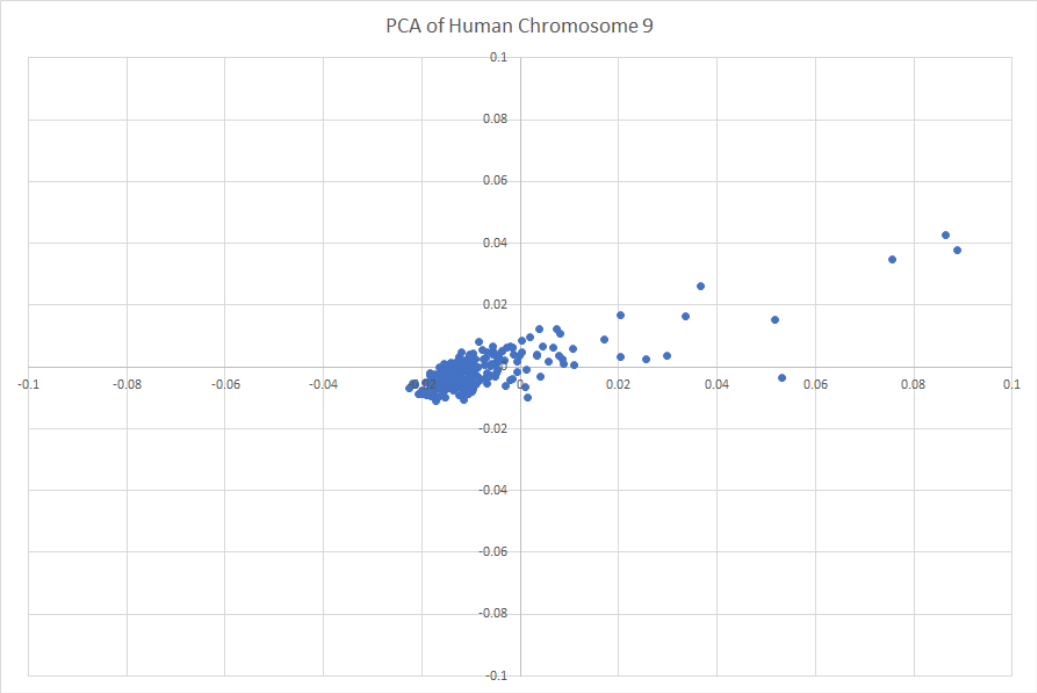

Figure 6.2: Two-dimensional PCA results for human chromosome 9 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 2,000-base length.

chromosomes did not necessarily exhibit any correlative or unique behavior with their PCA results, as seen in Figure 6.12. PCAs were also examined with regard to the chromosomes of species who have similar genomes to that of human beings: gorillas and chimpanzees. However, no discernible clustering or other unique behavior was observed, as can be seen in Figure 6.13 for gorillas and Figure 6.14 for chimpanzees.

6.3.3 Analysis

The explanation for the increase in clustering on certain human chromosomes under certain parameters of PCA experiments was elusive. Various attributes of the arbitrarily chosen regions were examined to determine if these were correlated with the clustering results in any way. After examining the area of the chromosome from

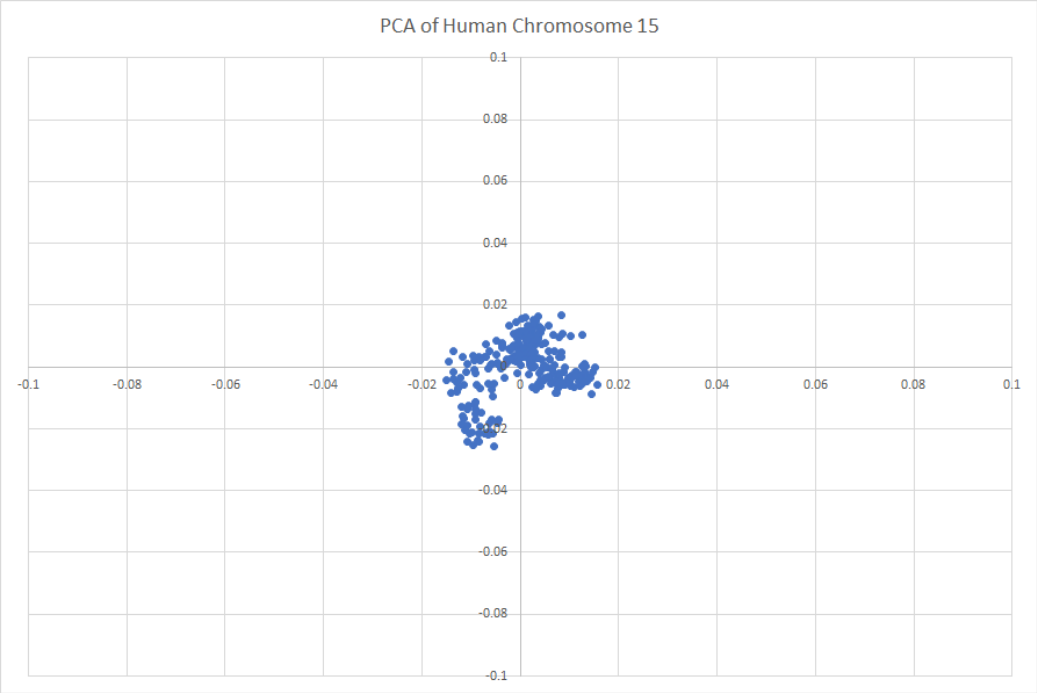

Figure 6.3: Two-dimensional PCA results for human chromosome 15 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 2,000-base length.

which the regions were selected, it was discovered that chromosomes 13, 14, and 15 had one thing in common with regard to the regions that were selected for the PCA: these regions were taken from the area of the chromosome existing around its centromere.

All human chromosomes come in pairs (with the exception of the X and Y chromosomes in a male, which are their own pair despite containing different genetic information). They exist as strands which are connected at a central point along each copy of the chromosome, referred to as the centromere. This is why visual representations of whole chromosomes often show an "X" shape, the center of the X being the centromere where both copies of the chromosome connect.

Recall that the clustering was observed for human chromosomes 13, 14, 15, and 19 but not for chromosomes like 1, 2, 5, etc. By convention, chromosomes (with the

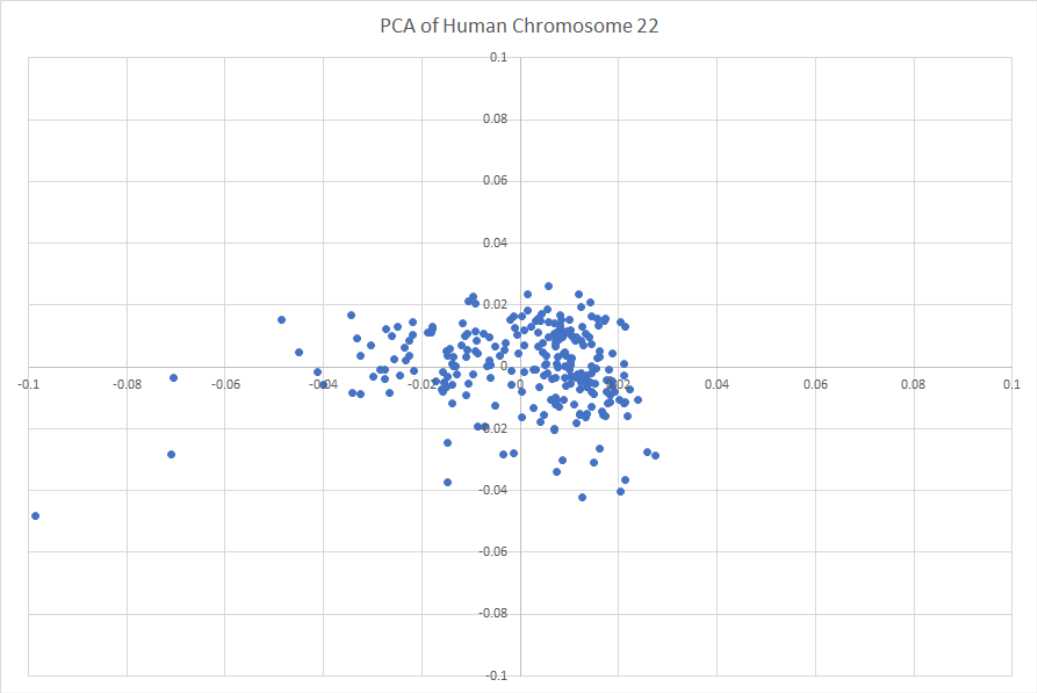

Figure 6.4: Two-dimensional PCA results for human chromosome 22 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 2,000-base length.

exception of X and Y sex chromosomes) are numbered according to length, with 1 being the longest chromosome and 22 being the shortest. Thus, chromosomes of a certain length exhibited this affect while longer ones did not. This was because the region selected for analysis (which was determined by the previously chosen starting point to avoid the telomeres) was sufficiently close to the centromere for shorter chromosomes such as 13 but was not close to the centromere for longer chromosomes such as 1, meaning that none of the material in the regions analyzed by the PCA for these longer chromosomes came from the centromere area.

Using the *UCSC Genome Browser* (UGB), these results can be examined on a more granular level. The results of the PCA for several chromosomes were regenerated with regions selected specifically to surround the centromere of the chromosome in question. These PCA results can be seen in Figure 6.15 for

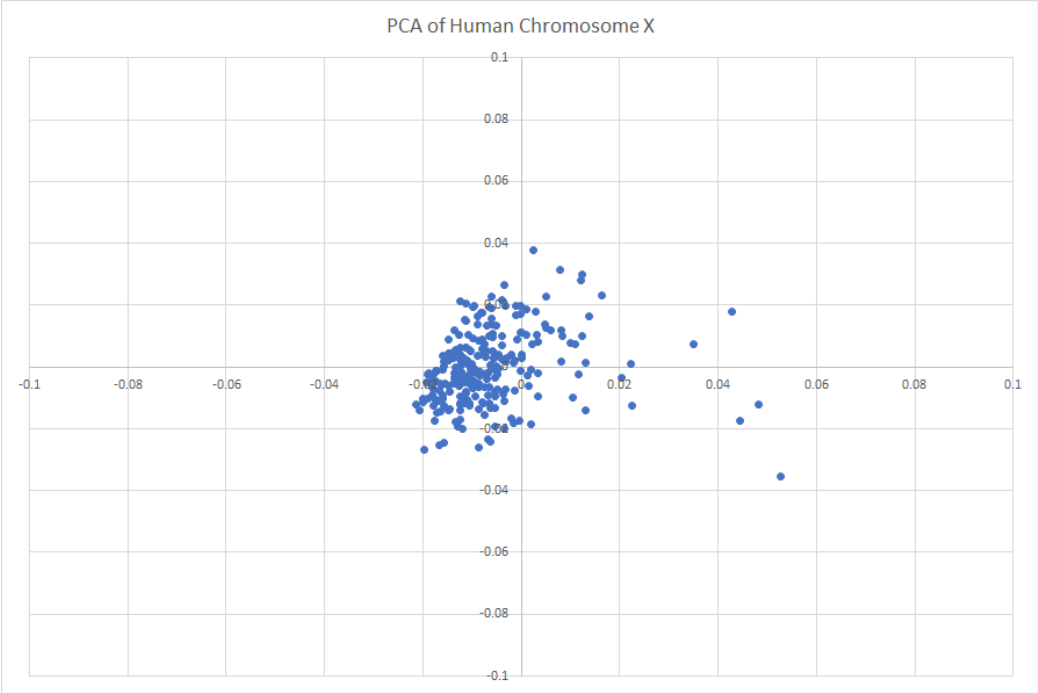

Figure 6.5: Two-dimensional PCA results for the human X chromosome with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 2,000-base length.

chromosome 15. Roughly four distinct clusters can be seen. There is a left one ($X < 0$), a right one ($X > 0.3$), a bottom one ($Y < -0.1$), and a middle one ($X > 0.1$, $-0.1 < Y < -0.5$). These AMI-profile vectors were taken in 50,000-base regions from positions 17,000,000 to 24,950,000. It turns out that the clusters on the graph correspond to distinct areas of the chromosome. The AMI-profile vectors that fall in the middle cluster are from regions at base positions 17,050,000 to 17,500,000. The vectors that fall within the bottom cluster come from base positions 17,500,000 to 18,350,000. The vectors that fall within the right cluster are from regions in base positions 18,350,000 to 19,800,000. The vectors that fall within the left cluster are, finally, from regions that come from base positions 19,850,000 to 24,950,000. According to the UGB, the centromere region begins at base position 17,499,052, the centromere occurs at position 18,355,008, and the centromere region ends at

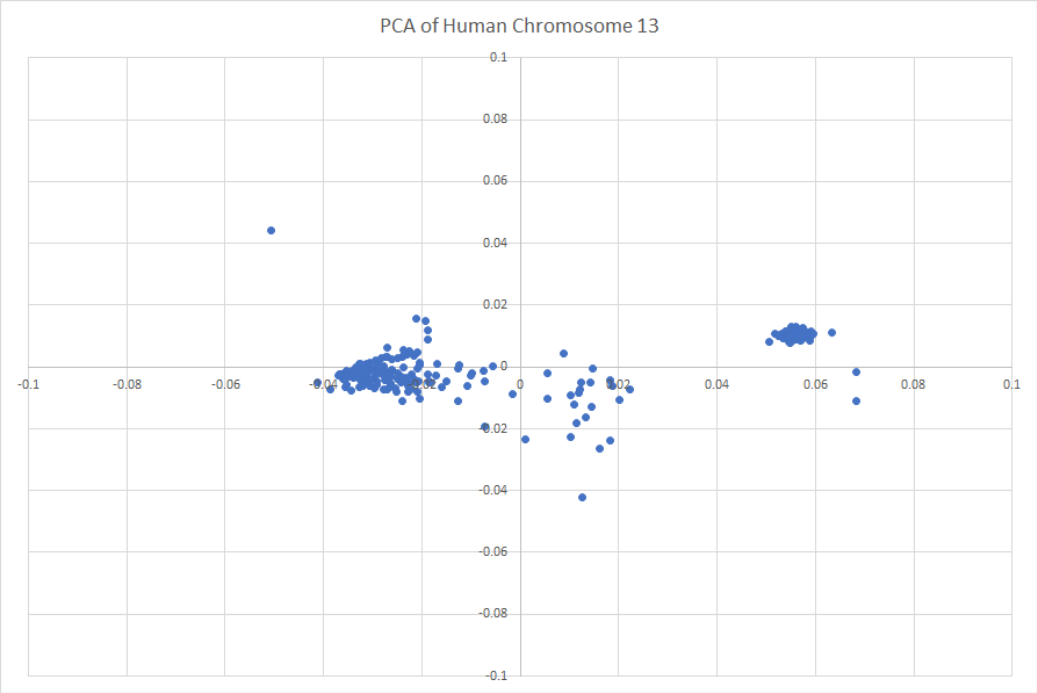

Figure 6.6: Two-dimensional PCA results for human chromosome 13 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 10,000-base length.

19,725,254 (see Figure 6.16). As can be noted, these regions are almost exactly mapped by the clusters. The first "part" before the centromere which is marked on the UGB is from 17,083,674-17,498,951, which corresponds to the middle cluster. The second "part" before the centromere is from 17,499,052-18,355,008 which corresponds to the bottom cluster. The part after the centromere is from 18,355,109-19,725,254 which corresponds to the right cluster. The left cluster falls outside the region of the centromere, after 19,725,254.

To demonstrate the uniqueness of this centromere material, the PCA was once again repeated for chromosome 15, this time including both results that were close to the centromere region and far away from the centromere region. This produces the results shown in Figure 6.17. It can be noted that any AMI-profile vector that does not fall within one of the centromere clusters tends to fall close to the origin of

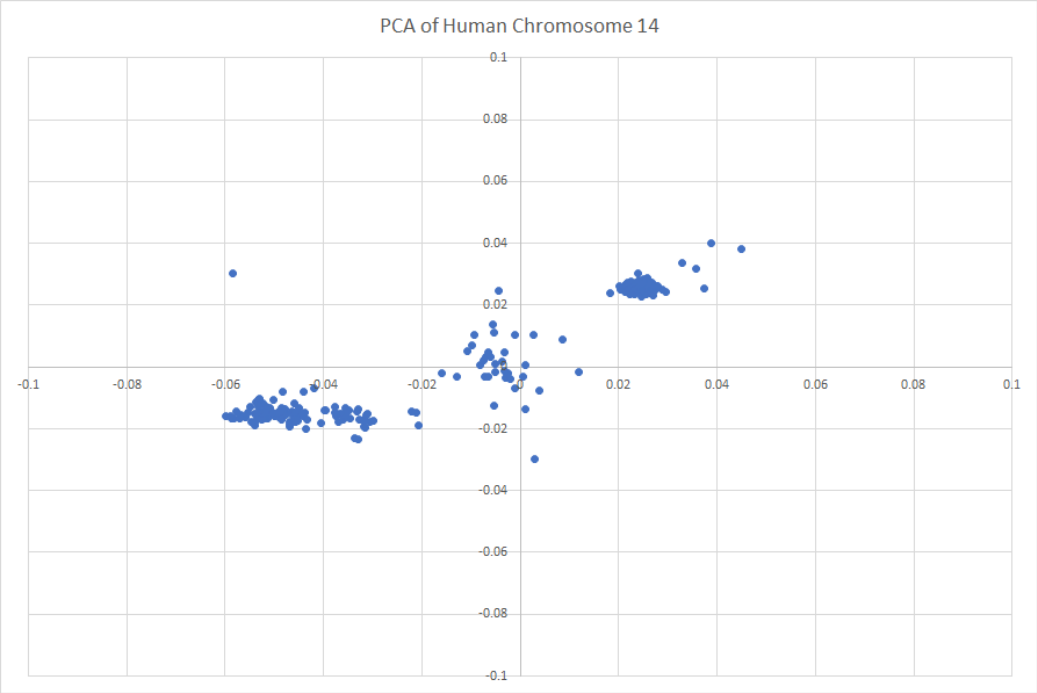

Figure 6.7: Two-dimensional PCA results for human chromosome 14 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 10,000-base length.

this plot in a "blob." This is essentially what was observed for the PCA of other chromosomes that were analyzed that did not include regions close to the centromere. Since it appears that DNA segments from regions not close to the centromere don't fall into distinct clusters, that seems to indicate that the DNA sequences existing around the centromere exhibit some type of special differentiation.

This same analysis can be repeated for human chromosome 13, yielding similar results which can be seen in Figure 6.18. In this figure, three distinct clusters can be seen. There is the left cluster, the middle cluster, and the right cluster. The AMI-profile vectors within the right cluster correspond to base positions 16,300,000 to 17,400,000 on chromosome 13. Those in the middle cluster correspond to positions 17,450,000 to 18,150,000. Finally, those in the left cluster correspond to

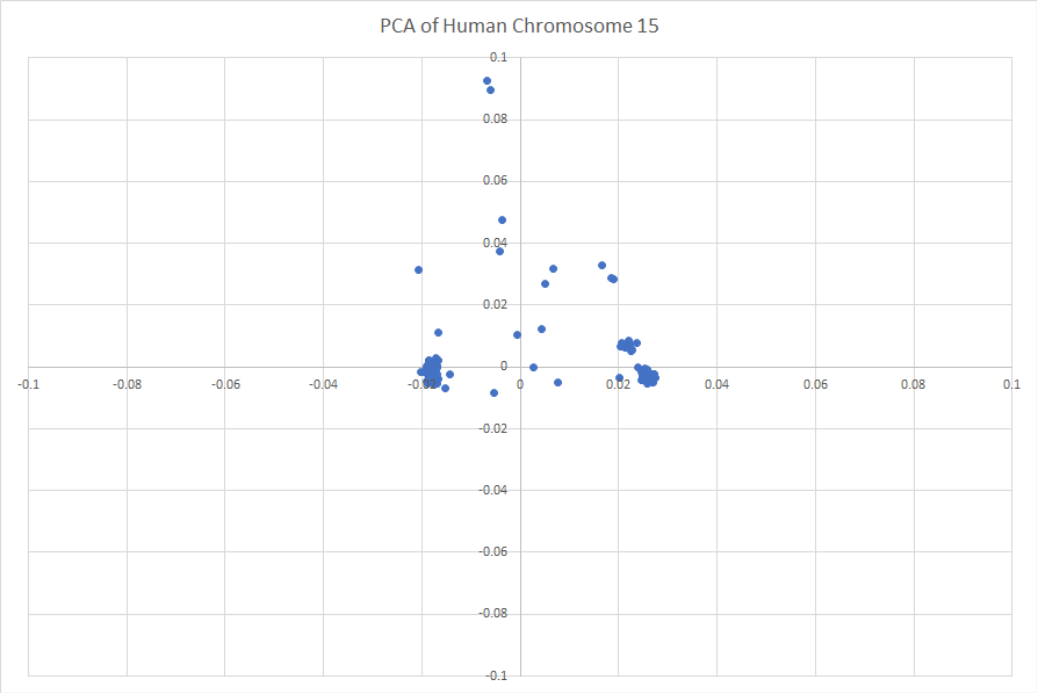

Figure 6.8: Two-dimensional PCA results for human chromosome 15 with AMI-profile vectors of size 30 calculated on DNA sequence regions of a 10,000-base length.

positions 18,150,000 to 40,950,000. According to the UGB, the centromere region starts at position 16,282,174, the centromere occurs at 17,416,384, and the centromere region ends at 18,051,248. Again, it can be noted that the right cluster corresponds to the first part of the centromere, the middle cluster corresponds to the second part of the centromere, and the left cluster (closest to the origin) is everything else. As a control, if PCA is performed on chromosome 13 in a region (base positions 18,200,000 to 43,150,000) that does not include the centromere region, the result in Figure 6.20 can be seen, where no clustering seems apparent.

Finally, revisiting one of the longer chromosomes that, under the initial conditions of this experiment, did not exhibit any clustering, confirms the results so far. Human chromosome 1 was originally tested in regions that occur after base position 17,360,000, which is nowhere near its centromere, which occurs at base

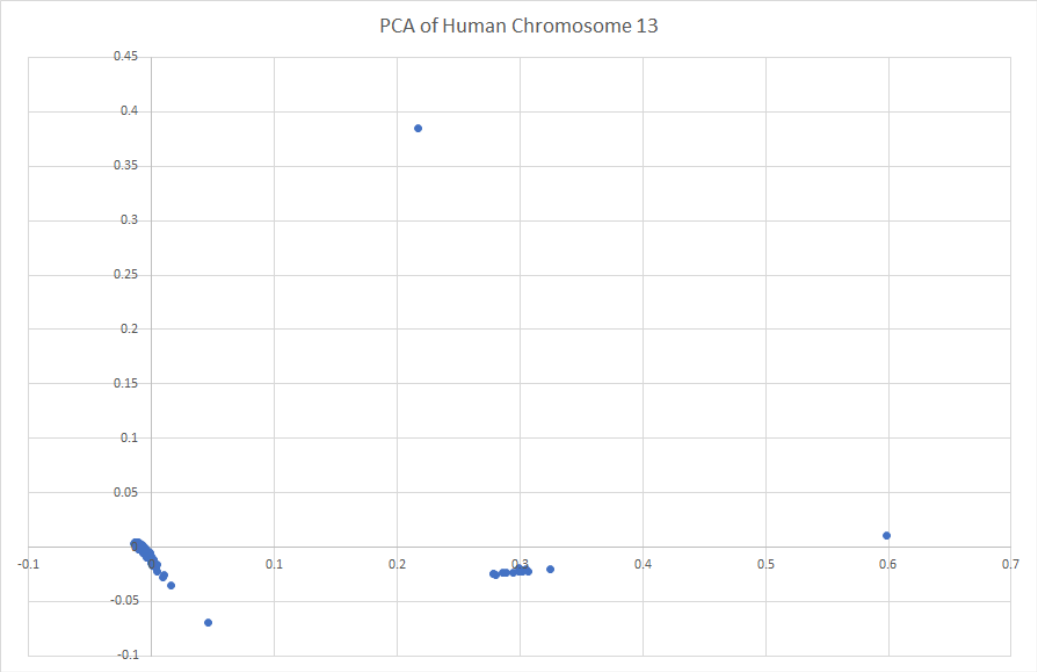

Figure 6.9: Two-dimensional PCA results for human chromosome 13 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length.

position 122,503,248. Producing a PCA for chromosome 1 around its centromere gives the results seen in Figure 6.21. There are *roughly* two clusters. One is below $Y = 0$ and the other above $Y = 0$ and to the left of $X = 0.5$. Those clusters correspond to the chromosome's base positions as follows: the vectors appearing in the bottom cluster come from base positions 119,563,100 to 121,563,100 and also 143,163,100 to 144,513,100. The vectors in the top cluster come from positions 122,013,100 to 125,013,100. Once again, if the UGB annotation is examined from Figure 6.22, it corresponds to this result: the centromere region begins at 122,026,460, the centromere occurs at 122,503,248, and the centromere region ends at 124,785,432. So, even though the clustering here didn't identify the centromere point exactly, it can still be seen that at least the entire centromere region is represented by the top cluster, which extends from roughly the beginning to the end

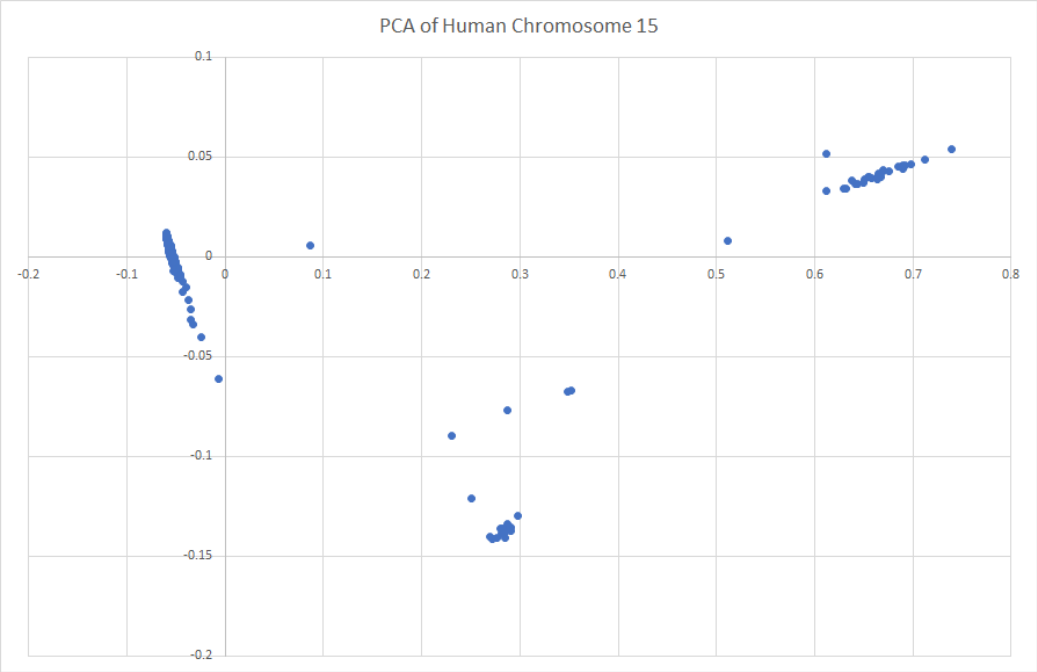

Figure 6.10: Two-dimensional PCA results for human chromosome 15 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length.

of the centromere. The other cluster records vectors that come from regions that fall outside the centromere area.

6.3.4 Explanation

These results were interesting in that they showed definite correlation; however, the question of *why* the DNA sequences found around the centromere were different still remained a mystery. The answer was eventually found in the documentation for the human chromosome assembly found on the UGB. This documentation reveals that the DNA bases represented in the centromere regions are not sequenced from the actual human genome:

> Centromeres are specialized chromatin structures that are required for cell division. These genomic regions are normally defined by long tracts

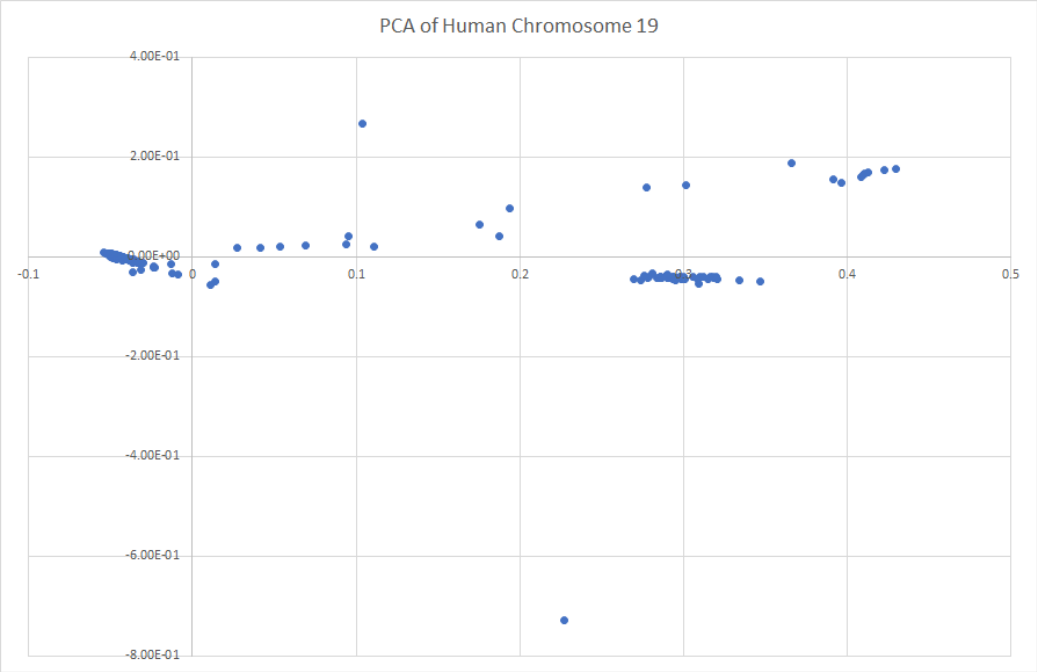

Figure 6.11: Two-dimensional PCA results for human chromosome 19 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length.

of tandem repeats, or satellite DNA, that contain a limited number of sequence differences to distinguish the linear order of repeat copies. The size and repetitive nature of these regions mean they are typically not represented in reference assemblies. Unlike all previous versions of the human reference assembly, where the centromere regions have been represented by a multi-megabase gap, GRCh38 incorporates centromere reference models that provide an initial genomic description derived from chromosome-assigned whole genome shotgun (WGS) read libraries of alpha satellite.

Each reference model provides an approximation of the true array sequence organization. Although the long-range repeat ordering is not expected to represent the true organization, the submissions are expected to provide a biologically rich description of array variants and

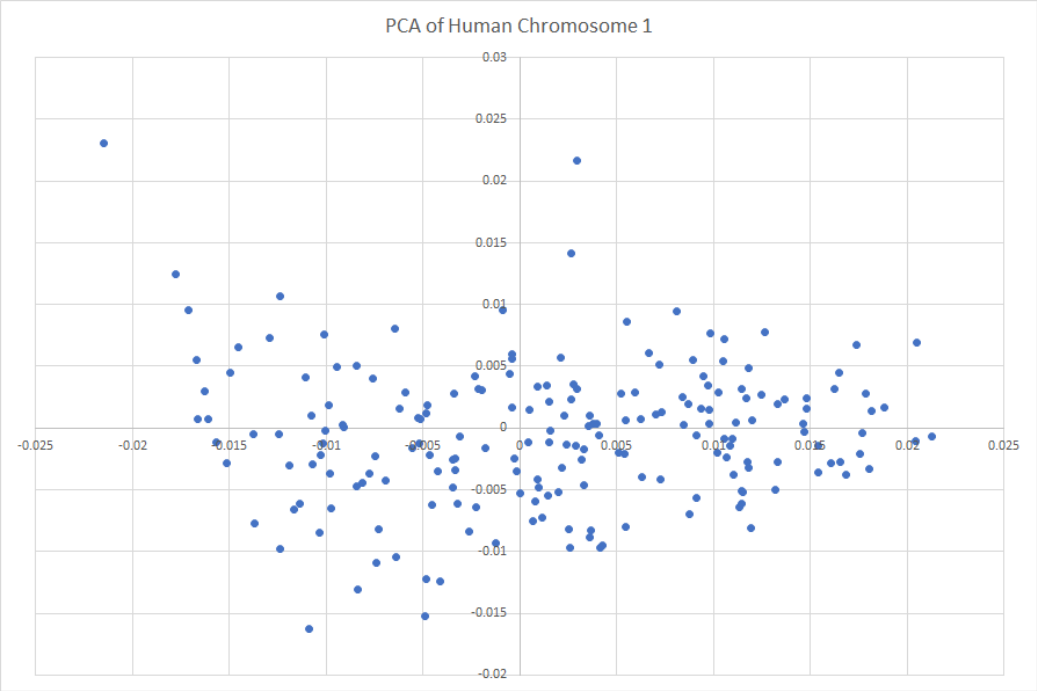

Figure 6.12: Two-dimensional PCA results for human chromosome 1 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length.

> local-monomer organization as observed in the initial WGS read dataset. As a result, these sequences serve as a useful mapping target to extend sequence-based studies to sites previously omitted from the human reference genome.[10]

In other words, DNA that occurs in the region around the centromere is difficult to sequence, and thus the bases which appear in those positions in the human genome assembly studied do not represent actual data sequenced from the human genome; rather, the bases in those positions in the assembly represent a *model* that is created to *simulate* the base patterns in the centromere region. Thus, the correlations that the PCA discovered were real effects of the data that was being analyzed. However, the reason that centromere-region DNA segments were differentiated from non-centromere-region DNA segments by their AMI profiles was

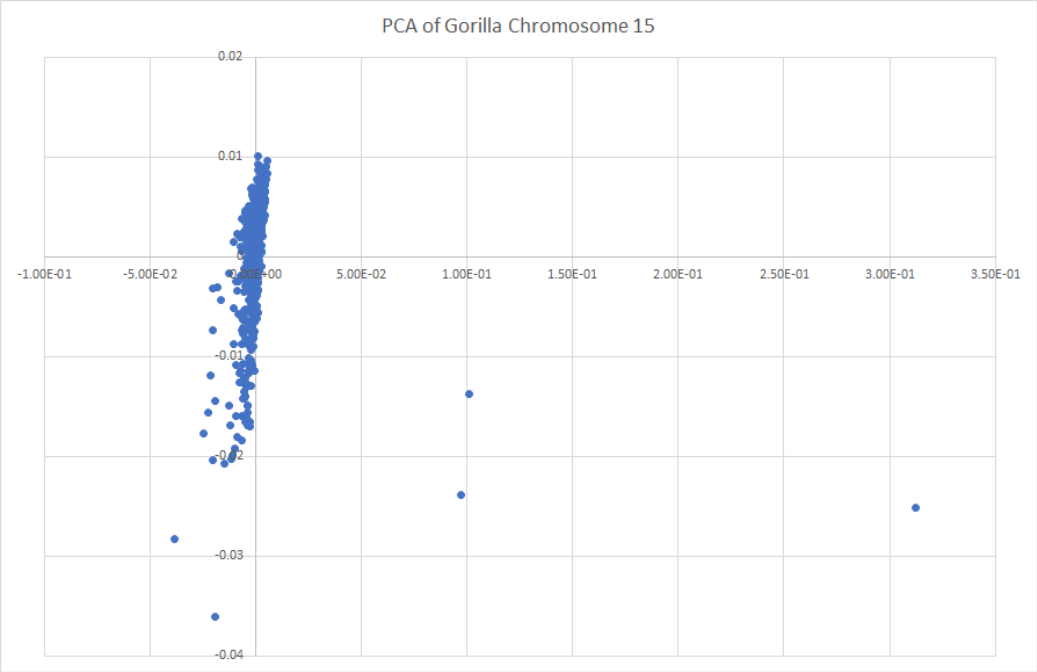

Figure 6.13: Two-dimensional PCA results for gorilla chromosome 15 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length.

because the DNA found in the centromere regions was not real human DNA; it was "synthetic." It is therefore the case that the PCA was able to identify which DNA segments belonged properly to the human genome from those which were synthetic interpolations; however, it was not able to differentiate between different regions of true human DNA. This would be consistent also with the results from the chimpanzee and gorilla DNA that was analyzed which showed no clustering. Those DNA assemblies did not contain synthetic, interpolated centromere-region components, and thus did not cause the PCA to detect any difference in the "type" of DNA it was encountering.

While this result does not confirm the original objective of this exploration, which was to discover a method to segment different types of *human* DNA, it did devise a method by which DNA which is foreign to the human genome can be detected through use of a PCA of various regions of the area in question. Since the

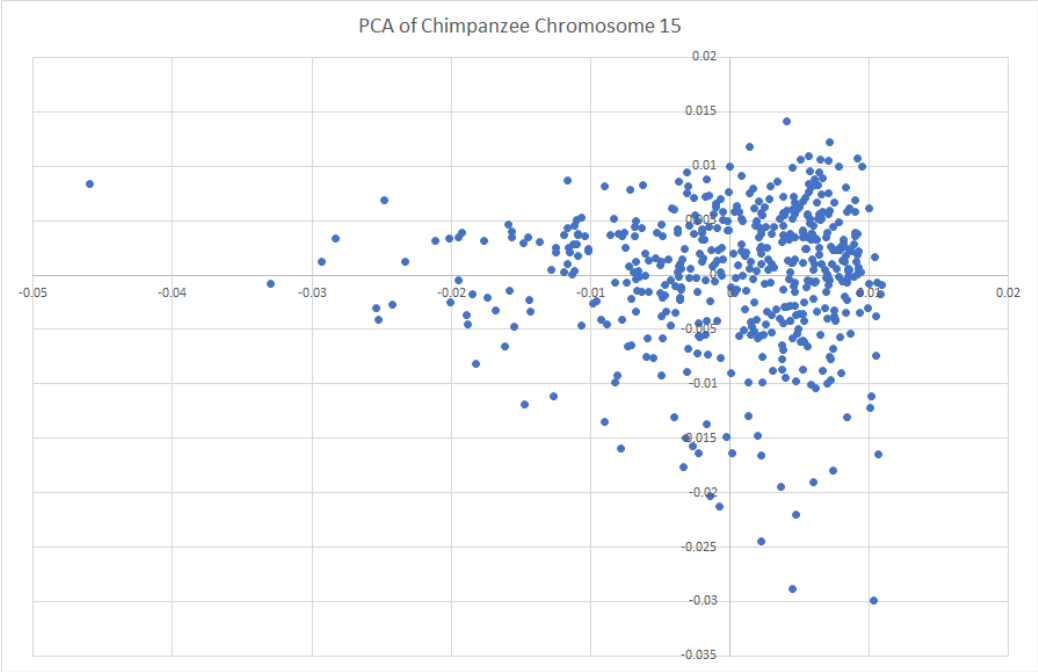

Figure 6.14: Two-dimensional PCA results for chimpanzee chromosome 15 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length.

PCA method was able to differentiate between these types of DNA, it is conceivable that it could, given the right input parameters, differentiate between different types of human DNA (or, more broadly, different types of DNA within a single species), but those conditions were not discovered. The result reconfirmed the analysis of [1], that the AMI profile, upon which the PCA vectors were based, is useful for differentiating between DNA that is native to a species' genome and DNA which is foreign to it.

6.4 Coding versus Noncoding Bacterial Regions

One final exploration with regard to the use of PCA to differentiate between various regions of DNA attempted to shift the paradigm being assumed. In the prior PCA results, DNA regions were chosen arbitrarily and were all of the same length. In

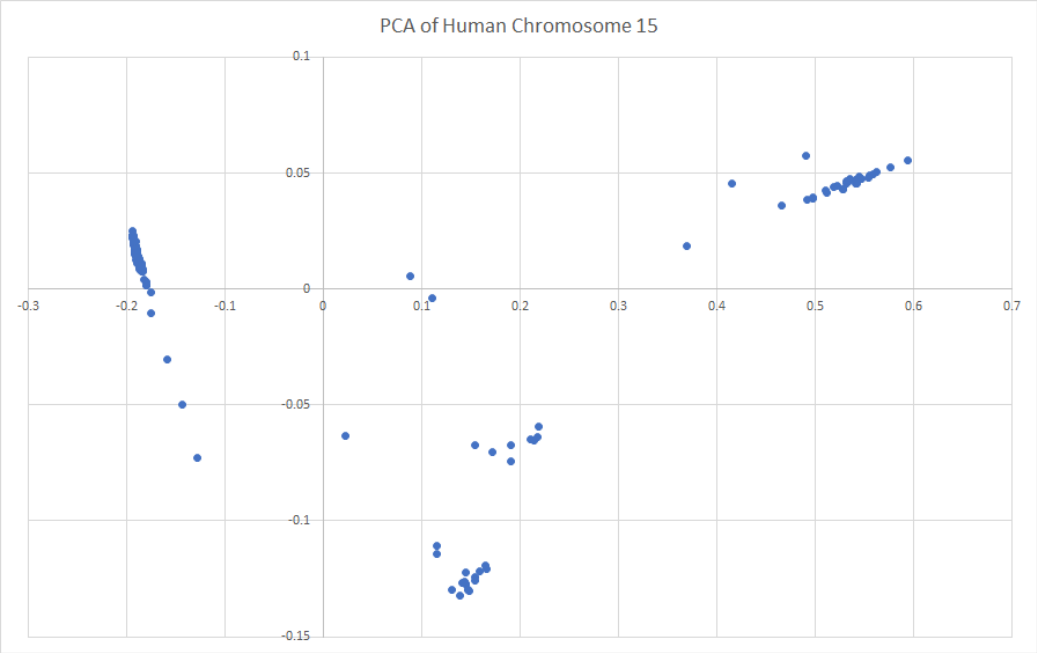

Figure 6.15: Two-dimensional PCA results for human chromosome 15 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length, selected to surround the chromosome centromere.

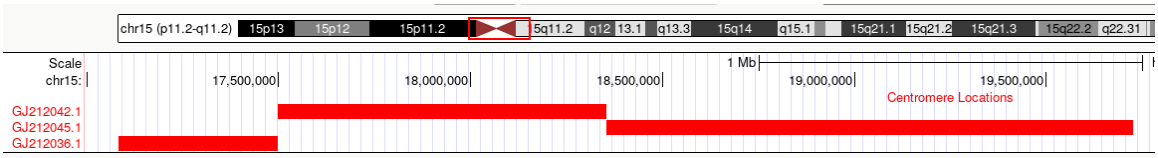

Figure 6.16: Excerpt from the *UCSC Genome Browser* showing the centromere area of human chromosome 15.[7]

other words, DNA regions were chosen without regard to any properties that the DNA they contained might be known to possess. However, due to the results showing that coding and noncoding regions could be successfully differentiated using various machine learning methods from Chapter 5, it was thought that perhaps the status of a DNA region as coding or noncoding could be predicted using a PCA. Because coding and noncoding regions are more apparent and easier to identify on a bacterial genome, *Staphylococcus aureus* was used. Bacterial genomes generally do

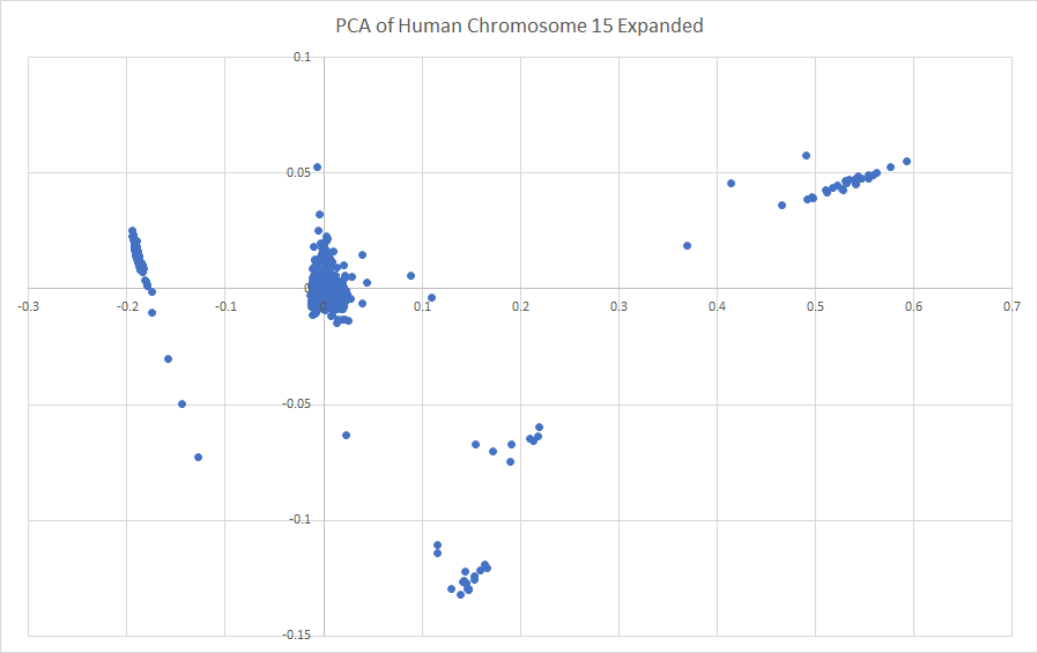

Figure 6.17: Two-dimensional PCA results for human chromosome 15 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length, selected to include both regions that surround the chromosome centromere and regions that do not.

not have multiple chromosomes, and their chromosome does not contain a centromere.

To perform this analysis, coding and noncoding regions from the *Staphylococcus aureus* genome were identified. Coding regions can be found on the annotated reference files associated with the genomic assembly. Any DNA regions not annotated as a "coding region" were considered a noncoding region. To avoid any influence from potentially aberrant behavior at the extremities of the chromosome, noncoding and coding segments which were not within the first 100 identifiable segments (coding or noncoding) of the chromosome were selected. Coding and noncoding regions were only chosen for this analysis if they consisted of at least two hundred bases, and enough segments were sequentially selected such that there were at least 200 of each type of segment. In the first experiment, there were 200

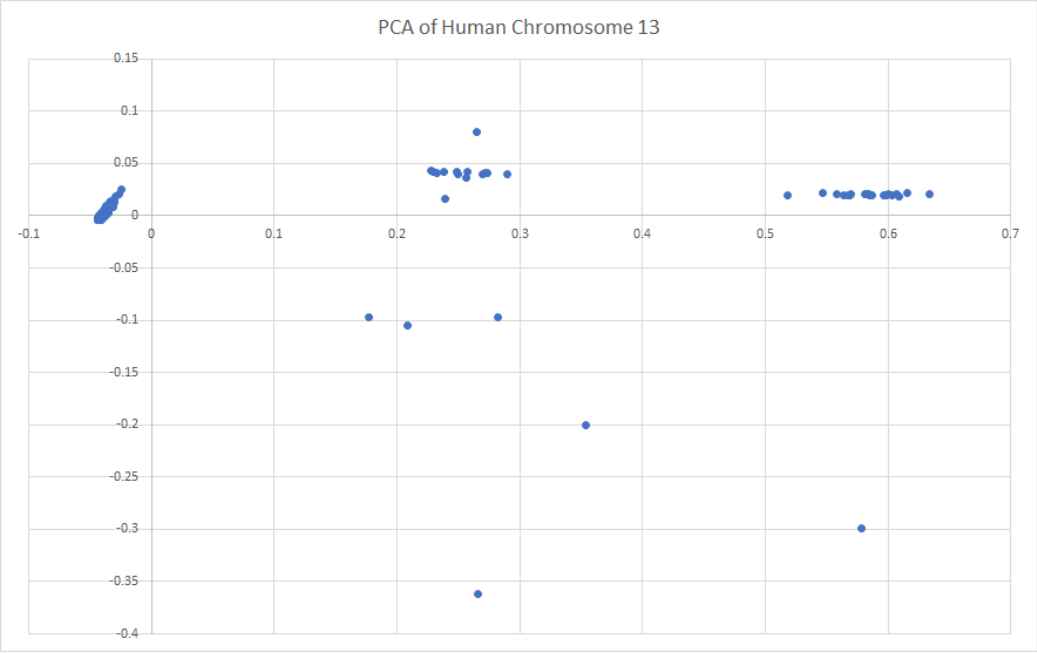

Figure 6.18: Two-dimensional PCA results for human chromosome 13 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length, selected to surround the chromosome centromere.

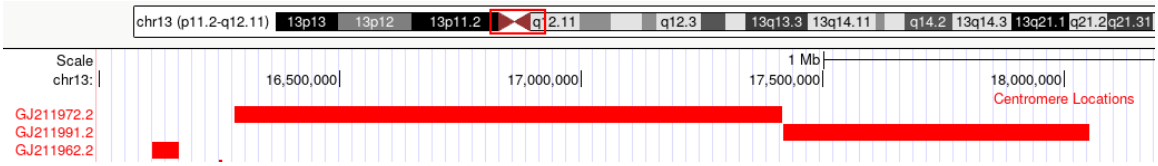

Figure 6.19: Excerpt from the *UCSC Genome Browser* showing the centromere area of human chromosome 13.[8]

noncoding segments and 609 coding segments. This was due to the fact that coding segments were much more plentiful, and thus 609 coding segments had to be encountered sequentially before at least 200 noncoding segments of size 200 bases or more were encountered.

When the AMI profiles (with 200 base positions of lag) of these 809 sequences were supplied as vectors to a PCA, the results were insignificant. The PCA did not reveal an identifiable clustering of coding and noncoding regions with respect to each other. This can be seen in Figure 6.23 where coding and noncoding regions

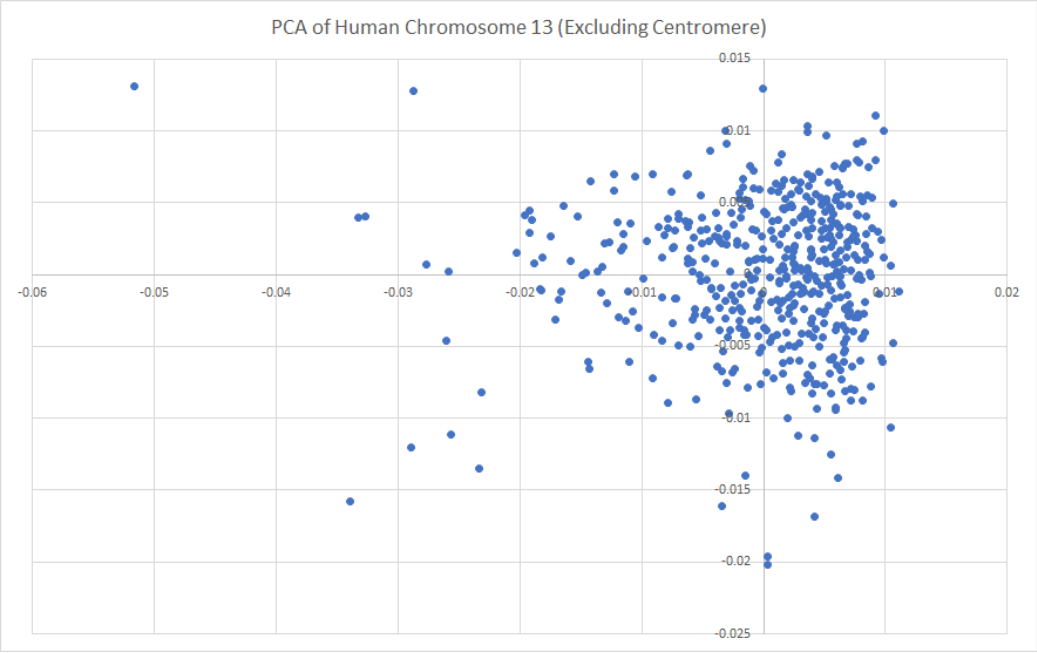

Figure 6.20: Two-dimensional PCA results for human chromosome 13 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length, selected to exclude regions around the chromosome centromere.

overlap each other close to the origin of the graph. This same experiment was repeated with different coding and noncoding regions that were chosen by starting after the first 500 identifiable segments instead of the first 100. The results shown in Figure 6.24 are virtually indistinguisable from those of Figure 6.23.

Finally, coding and noncoding sequences of at least 500 bases in length were considered without regard to their location on the *S. aureus* genome and with the stipulation that at least 150 of each type of sequence be included. This resulted in a total of 150 noncoding sequences and 1,841 coding sequences being considered in the PCA. As the results in Figure 6.25 show, once again, the coding and noncoding sequences cannot be differentiated. It can be observed that the noncoding sequences all lie to one side of the projected area; however, this is uninteresting since the noncoding sequences are intermingled with and therefore not separable from the coding sequences.

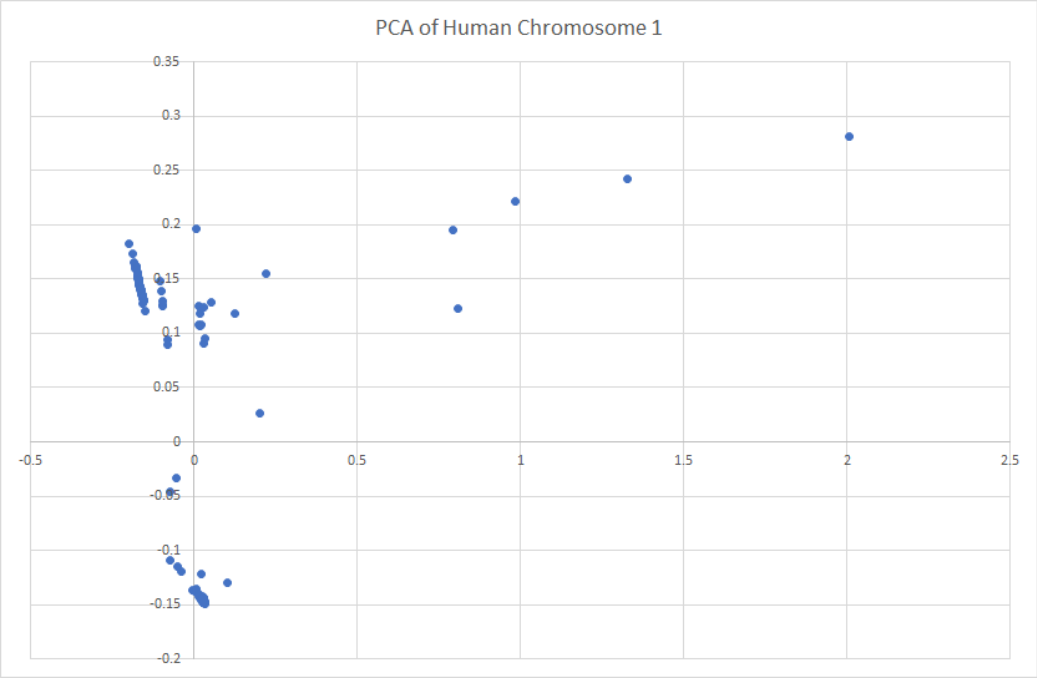

Figure 6.21: Two-dimensional PCA results for human chromosome 1 with AMI-profile vectors of size 200 calculated on DNA sequence regions of a 50,000-base length, selected to surround the chromosome centromere.

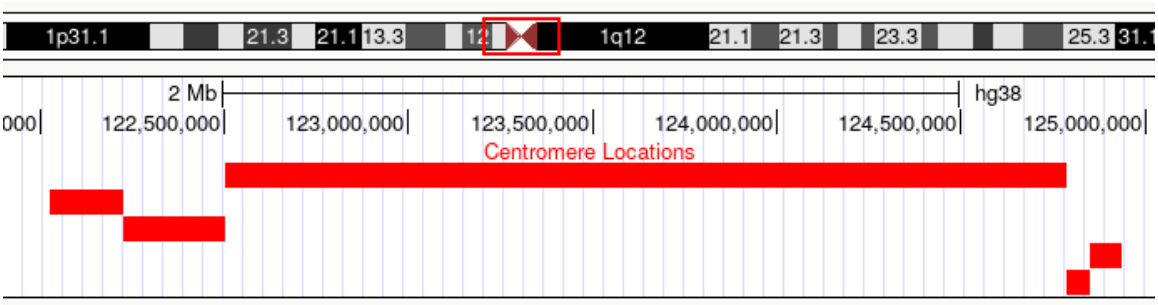

Figure 6.22: Excerpt from the *UCSC Genome Browser* showing the centromere area of human chromosome 1.[9]

6.5 Conclusion

This foray into the use of PCA to assess the relatedness of regions of DNA sequences proved only successful in an unintended way. In general, PCA based upon the AMI profile was unable to differentiate between differing regions of DNA sequences, whether or not those sequences came from the human genome or a bacterial

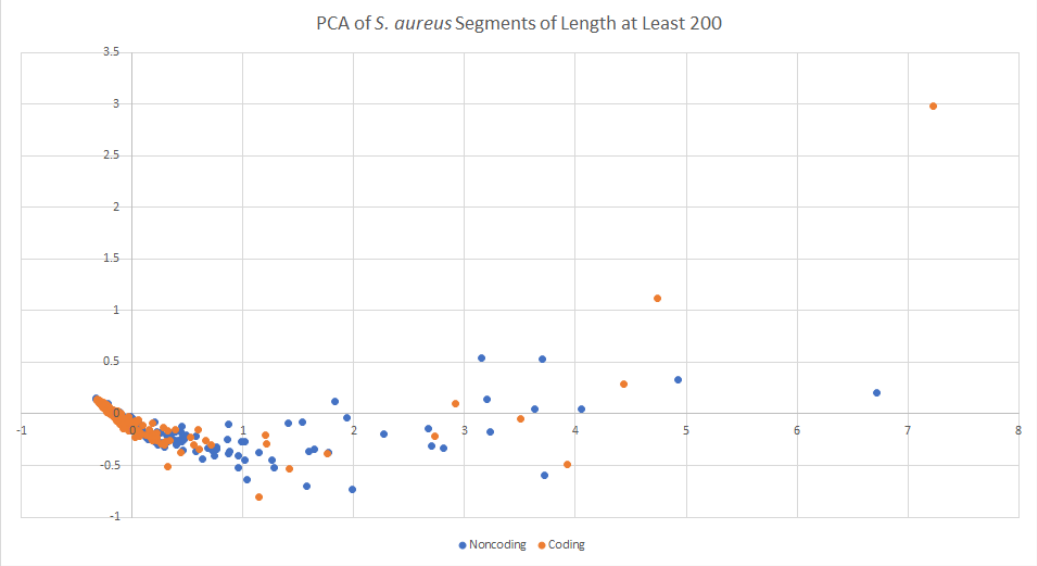

Figure 6.23: Two-dimensional PCA results for *S. aureus* with AMI-profile vectors of size 200 calculated on coding and noncoding DNA sequences of at least a 200-base length, selected from after the first 100 sequences encountered.

genome. This was even the case when preexisting knowledge of the various regions (whether coding or noncoding) was taken into consideration with the selection of the regions tested. It was discovered that PCA was useful at differentiating real human DNA sequenced from the human genome from approximated human DNA generated from models, particularly for the centromere region. Since it was demonstrated that PCA can differentiate between human and nonhuman DNA, it is likely that it would be able to differentiate between DNA sequences from differing species, similar to the results reported by Bauer, Schuster, and Sayood.[1]

Future work in this area could include a study of other types of known regions to see if using qualities that are already known can result in a differentiation. Other metrics related to the standard AMI profile could also be employed in producing the input vectors for the PCA analysis, such as the single-base AMI profiles discussed in Chapter 3 and the triplet AMI profiles explored in Chapter 4. Additionally, the PCA could be extended to three or four dimensions and more quantitative

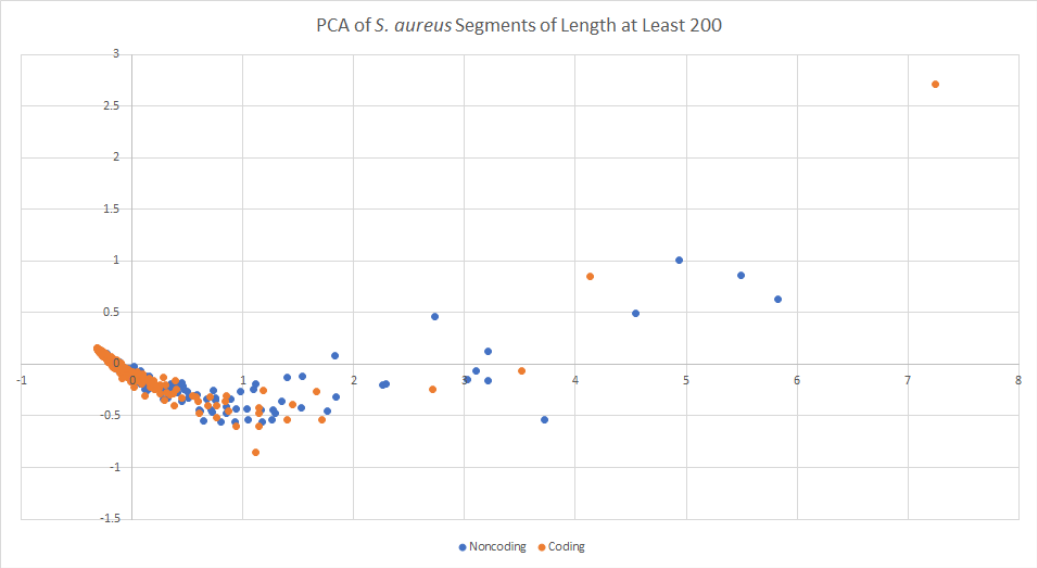

Figure 6.24: Two-dimensional PCA results for *S. aureus* with AMI-profile vectors of size 200 calculated on coding and noncoding DNA sequences of at least a 200-base length, selected from after the first 500 sequences encountered.

clustering methods, rather than the qualitative observations used in this analysis, could be employed to ascertain if regions can be differentiated.

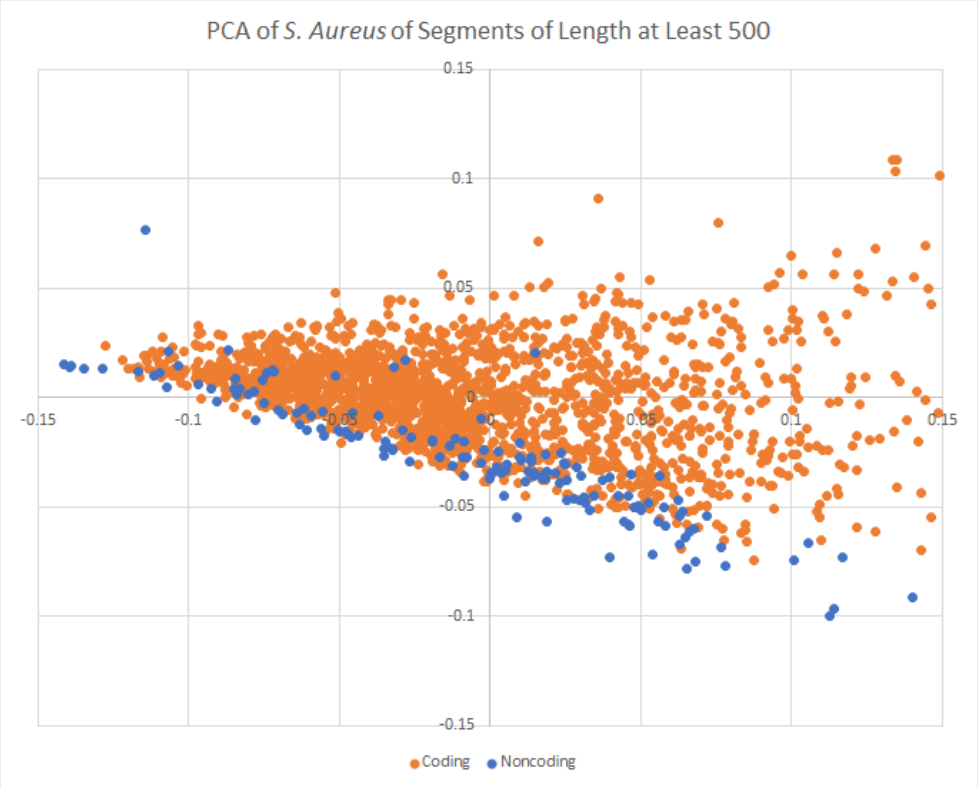

Figure 6.25: Two-dimensional PCA results for *S. aureus* with AMI-profile vectors of size 200 calculated on coding and noncoding DNA sequences of at least a 500-base length.

Chapter 7

Arithmetic Coding as a Means of DNA Sequence Compression

7.1 Introduction

In every preceding chapter of this investigation with the exception of the preliminary explorations in Chapter 2, the AMI profile or a variant of it has been used to help quantify the information structure of any DNA sequence under examination. However, for purposes of the material in this chapter, the AMI profile, the thread that has been tying every one of these experiments together, will be abandoned as a metric by which to quantify the content of DNA sequences. The original aim for the use of the AMI profile was for it to aid in the formulation of a compression technique for DNA sequences.

Instead, this chapter will approach compression more directly, and the relative frequency with which groups of bases appear in a particular DNA sequence will be used to facilitate the compression. Groups of bases in a DNA sequence are generally known as "k-mers," denoting a group of k bases in a row. In order to compress data, something about the structure of the data must be known. For several data compression techniques, the encoding of an entity relies on the "context" in which that entity occurs. In terms of DNA sequences, the order of bases in a sequence is

highly significant, and thus it can be noted that data compression techniques that use context-based approaches could be made to encode DNA bases based on the context in which they appear. In other words, the context would consist of the k-mer preceding the base being encoded.

7.2 Arithmetic Coding

Arithmetic coding is a data compression technique whose encoding and decoding algorithms do not require unique binary codes to be generated for each symbol that could be encountered. Arithmetic coding relies on the skewed nature of the probability distributions of the symbols which will be encountered relative to each other, and it partitions an interval into segments whose width is based on their relative probability of appearance.[3] Thus, arithmetic coding generally works well if some symbols encountered are very common whereas others are very rare. Context-based arithmetic coding extends this idea by not only considering the probability that a symbol will appear, but rather it considers the probability that a symbol will appear *in a given context*.

For DNA sequences, as the size of k-mers increases, certain k-mers become fairly commonly encountered, while other k-mers are are almost never encountered if at all. This makes arithmetic coding an ideal data compression technique for this situation since arithmetic coding does not require that codes be generated for all k-mers which could be encountered, rather only considering the ones which actually *are* encountered.

There are two types of arithmetic coding which were explored. The first type was adaptive arithmetic coding. With an adaptive approach, the frequency of occurrence of each base in the context of a particular k-mer is generated as the

encoding algorithm progresses. Thus, nothing is known about the frequency of occurrence of any k-mers within the DNA sequence at the outset of the encoding process. Contrary to that, the second type of arithmetic coding implemented was omniscient arithmetic coding. With the omniscient approach, the frequency of occurrence of every k-mer of a particular size is calculated up front, so that the probability distributions for encountering a particular base in the context of a particular k-mer are known.

The major benefit of an adaptive approach is that the context tables can be learned as both the encoding and decoding algorithms progress, and thus the context tables do not have to be stored within the compressed output in order for decoding to work correctly. For the omniscient approach, by contrast, since the context tables are determined up front before the encoding process begins and are thus fully known by the encoder, they must be somehow stored with the compressed output because the decoder must also have access to those same context tables in order to accurately decode the compressed content. However, the major benefit of an omniscient approach is that it can know whether certain bases will ever be encountered in certain contexts with certainty, and thus it does not have to allot any of its compression "space" to the possibility that such a base in such a context might occur. Thus, an omniscient approach can more beneficially distribute its intervals such that greater compression is possible. Adaptive approaches, on the other hand, must assume that any base could be encountered in any context, and thus the leeway the algorithm has to partition its intervals in a beneficial manner becomes more limited, thus reducing the amount of compression that can be achieved.

7.3 k-mer Analysis

It was hypothesized that context-based arithmetic coding using k-mers of various sizes in DNA sequences would be able to sufficiently describe the information structure within these sequences and would thus be amenable to arithmetic coding. This hypothesis was based on an analysis of the frequency of k-mer occurrences in various DNA sequences. When calculating what would serve as the context tables for an arithmetic approach, it was observed visually that these tables exhibited a type of periodicity with respect to which k-mers occurred. In other words, the sparsity in these context tables was not randomly distributed but was rather "structured" in how it appeared.

For the analysis contained in this chapter, bacterial genomes were studied for several reasons. Among these were that bacterial genomes consist of a single chromosome, meaning that a genome-level compression can be attempted. Further, bacterial DNA is generally regarded as simpler than the DNA of other more complex organisms. Various features of human chromosomes that became relevant factors of the experiments performed in previous chapters could be avoided by using bacterial genomes. These included the aberrant behavior of the DNA contained in chromosome telomeres along with the interpolated DNA around the centromeres as well as the fact that not all DNA in more complex organisms can be sequenced, leading to regions of various chromosomes where the bases are undetermined (and thus represented by an "N" in the genome assemblies). In bacterial genomes, it was observed that the frequency of undetermined bases is very low, and this made these types of sequences more desirable in that such complexity-increasing nonidealities could be avoided.

The bacterial sequences studied in this analysis were chosen simply if they appeared to be well-studied and well-represented in the NCBI databases. The genomes studied included the following: (Each species name is listed along with its NCBI accession number and its abbreviation in the tables that follow.)

- *Staphylococcus aureus* [NC_007795.1] ("*S. aur.*")
- *Escherichia coli* [NC_000913.3] ("*E. coli*")
- *Mycobacterium tuberculosis* [NC_000962.3] ("*M. tub.*")
- *Porphyromonas gingivalis* [NZ_CP011995.1] ("*P. gin.*")
- *Pseudomonas aeruginosa* [NC_002516.2] ("*P. aer.*")
- *Streptococcus pyogenes* [NZ_LS483338.1] ("*S. pyo.*")
- *Thermus thermophilus* [NZ_AP019794.1] ("*T the.*")
- *Acaryochloris marina* [NZ_AP026075.1] ("*A. mar.*")

Figures 7.1, 7.2, 7.3, and 7.4 show a sampling of the periodicity observed with regard to the frequency of appearance of each k-mer. Each image represents a visualization of the context "space." Each pixel in the image represents a particular k-mer of size 12. Exactly enough pixels are allocated to each image to display exhaustively all the possible k-mers. The k-mers are displayed with the order of preference of bases being A, C, G, T and beginning in the upper left corner of each image then moving right and then down. Thus the top left pixel represents the 12-mer AAAAAAAAAAAA, the next pixel to its right represents AAAAAAAAAAAC, and so on. A purely black pixel represents a k-mer that does not appear at all, and a purely white pixel represents a k-mer that appears most

frequently compared to all the others in the genome under consideration. Pixels are graded from black to white based on the frequency with which its corresponding k-mer occurs. Surprisingly periodic, non-random patterns can be readily observed in the distribution of the frequency of these k-mers.

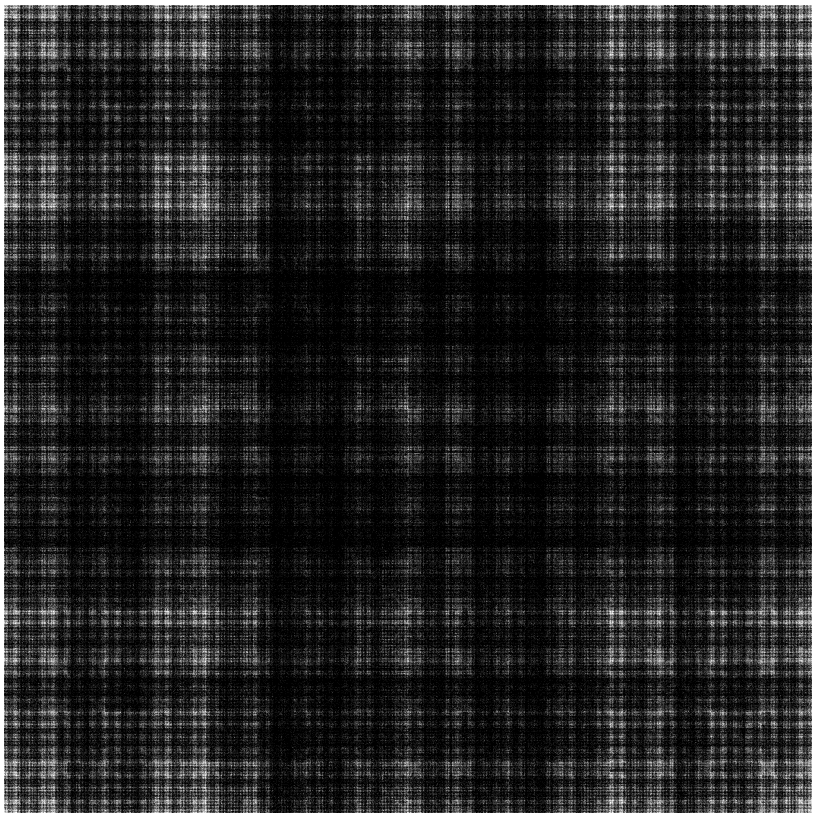

Figure 7.1: Image of the context space showing the relative frequency of the appearance of k-mers of size 12 for *Staphylococcus aureus*. Black pixels represent no appearance of a particular k-mer; white pixels represent frequent appearance of a particular k-mer.

7.4 Adaptive Arithmetic Coding Results

An adaptive arithmetic coder was designed to use a k-mer of specified length as its context and to encode bases encountered in the sequence in light of that k-mer

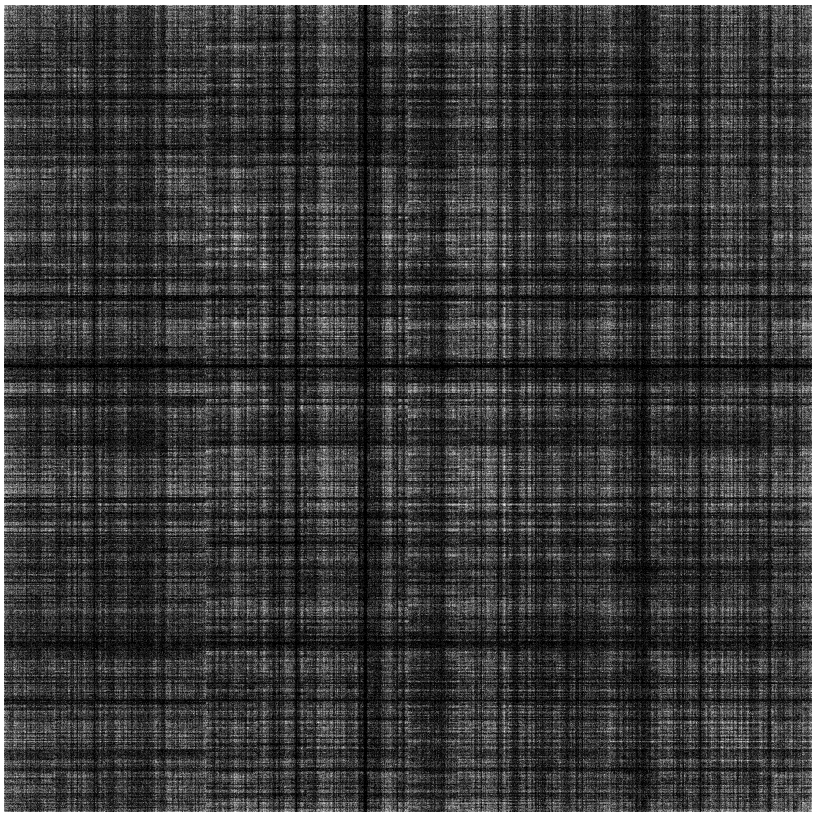

Figure 7.2: Image of the context space showing the relative frequency of the appearance of k-mers of size 12 for *Escherichia coli*. Black pixels represent no appearance of a particular k-mer; white pixels represent frequent appearance of a particular k-mer.

context. To start, each k-mer in the context space had to be given a nominal count of 1 in order to allow for the possible appearance of any k-mer.

The compressed sequence produced by the arithmetic coder was given in terms of bits. To store a DNA sequence directly in terms of bits, without any compression, only two bits per base are required since DNA is only a four-letter alphabet and a basic code can be employed as follows: A - 00, C - 01, G - 10, T - 11. Thus, in order to be successful with compression, the arithmetic coder must be able to improve over this two-bit-per-base metric. Thus, compression results are given in terms of a "bit rate," which is the number of bits in the compressed sequence divided by the number of bits that would be required by the uncompressed sequence just described.

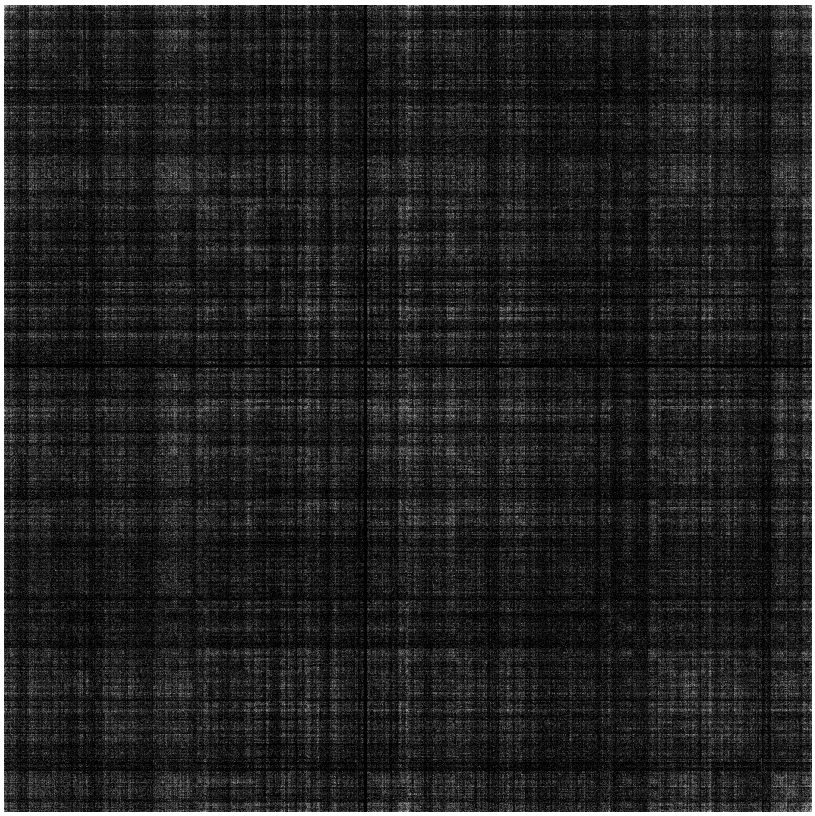

Figure 7.3: Image of the context space showing the relative frequency of the appearance of k-mers of size 12 for *Porphyromonas gingivalis*. Black pixels represent no appearance of a particular k-mer; white pixels represent frequent appearance of a particular k-mer.

Although undetermined ("N") bases were rare in the bacterial genomes studied, when an occasional undetermined base was encountered, it was simply replaced with A in order to reduce the complexity of the algorithm designed. In practice, undetermined bases would either have to be considered as an additional "base" and added to the k-mer context tables (which would increase their size by a whole dimension) or the positions of undetermined bases would have to be simply recorded by the encoder and re-added manually by the decoder.

Applying the adaptive approach to the *S. aureus* genome for k-mers of sizes 2 to 12 produced the results displayed in Table 7.1. As can be seen, the best compression

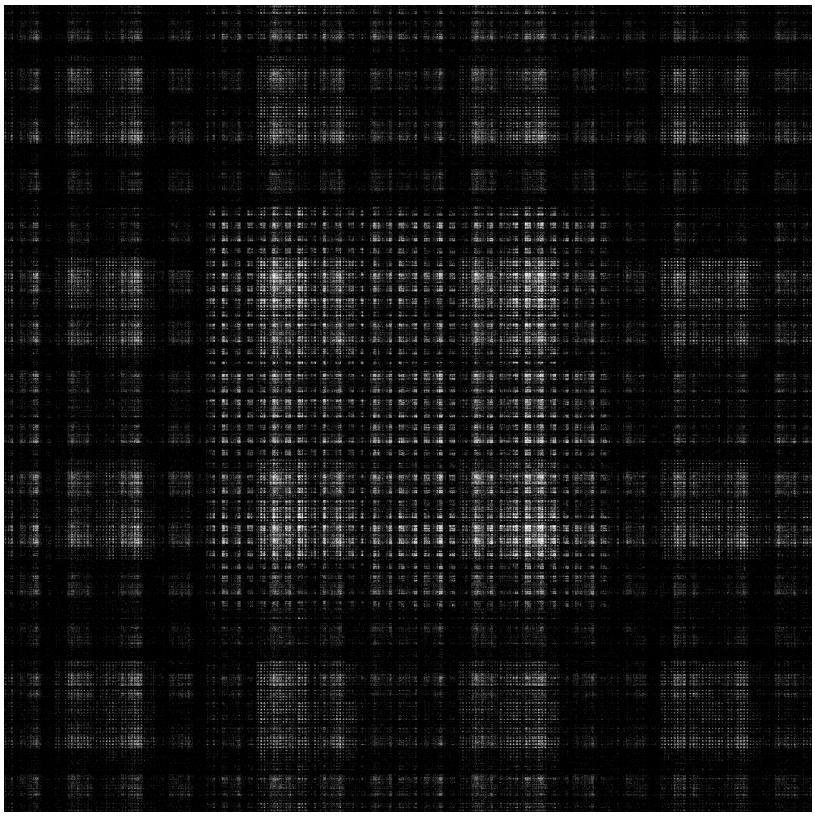

Figure 7.4: Image of the context space showing the relative frequency of the appearance of k-mers of size 12 for *Thermus thermophilus*. Black pixels represent no appearance of a particular k-mer; white pixels represent frequent appearance of a particular k-mer.

bit rate was 0.95 bits per bit, and the best compression ratio observed was 5.56%. Overall, these do not represent high rates of compression. The reason compression results were lackluster is likely due to the main drawback of the adaptive approach itself: since every possible k-mer might possibly appear, the context tables have to allow for every possible k-mer and thus cannot use any knowledge of k-mers that are completely absent to "widen" the compression interval. This is why the compression actually degrades as the k-mer size is increased. While the sparsity of the context tables for large k-mers would initially seem to help the compression, it actually becomes a liability for the adaptive method since the context space increases

k-mer Size	Bits	Ratio	Bit Rate
2	4,468,573	4.92%	0.95
3	4,451,363	5.29%	0.95
4	4,444,904	5.43%	0.95
5	4,443,557	5.46%	0.95
6	4,451,192	5.29%	0.95
7	4,491,571	4.44%	0.96
8	4,566,318	2.84%	0.97
9	4,645,616	1.16%	0.99
10	4,691,660	0.18%	1.00
11	4,692,439	0.16%	1.00
12	4,677,597	0.48%	1.00

Table 7.1: Results of adaptive arithmetic coding for k-mers of size 2 to 12 on the major portion of the *S. aureus* genome before the first 'N' base is encountered, which consisted of 2,350,011 bases, requiring 4,700,022 bits uncompressed.

four-fold when the k-mer size adds an additional base, making many more k-mer possibilities which need to be accounted for.

7.5 Omniscient Arithmetic Coding Results

An omniscient arithmetic coder was also designed to use a k-mer of specified length as its context and to encode bases encountered in the sequence in light of that k-mer context. However, before encoding began, the k-mer context tables were fully generated. Besides this adjustment, the omniscient arithmetic coder worked identically to the adaptive arithmetic coder, and the same methods used to evaluate the compression results of the adaptive arithmetic coder (bit rate, etc.) were also used to evaluate the results of the omniscient approach.

The performance of the omniscient approach, as can be expected, was drastically improved. The results of omniscient arithmetic compression for all eight bacterial genomes examined can be seen in Table 7.2. As the size of the k-mers increases, so

k	*S. aur.*	*E. coli*	*M. tub.*	*P. gin.*	*P. aer.*	*S. pyo.*	*T. the.*	*A. mar.*
2	0.95	0.98	0.95	0.99	0.94	0.97	0.9	0.99
3	0.95	0.98	0.94	0.98	0.93	0.97	0.89	0.99
4	0.95	0.97	0.94	0.98	0.92	0.96	0.88	0.98
5	0.94	0.97	0.93	0.98	0.91	0.96	0.87	0.98
6	0.94	0.96	0.93	0.97	0.9	0.96	0.85	0.98
7	0.93	0.96	0.92	0.96	0.89	0.95	0.84	0.98
8	0.91	0.94	0.91	0.93	0.88	0.91	0.81	0.97
9	0.83	0.89	0.86	0.82	0.84	0.78	0.75	0.93
10	0.66	0.72	0.72	0.55	0.75	0.53	0.64	0.8
11	0.41	0.42	0.5	0.25	0.6	0.26	0.5	0.5
12	0.2	0.18	0.28	0.08	0.4	0.1	0.33	0.21

Table 7.2: Bit rate results of omniscient arithmetic coding for k-mers of size 2 to 12 on the eight bacterial genomes.

does the sparsity of the k-mer context tables (because large numbers of possible k-mers do not appear on the sequences being compressed), and thus the compression as quantified by the bit-per-bit rates is impressive. For k-mers of size 12, the best sequence compression is exhibited by *S. pyogenes* at 0.1 compressed bit per raw bit. All bacterial genomes tested exhibited basically the same behavior in this regard.

However, these excellent compression results obtained with the omniscient approach are not directly comparable to the results obtained for the adaptive approach. There is a bit of a "sleight of hand" going on here. Because the omniscient approach assumes that the context table is known up front, the information contained within that table is *not* stored in the compressed result and thus is not reflected in the bit rates reported in Table 7.2. Thus, it will take extra storage (more bits) to transmit both the compressed sequence *and* the k-mer context tables to the decoder. This is not something the adaptive approach has to worry about as its k-mer context tables are "baked into" its compressed sequence.

Methods for storing the k-mer context tables and recovering them for the decoder were explored. One of those involved simply saving the k-mer count tables

as compressed image files similar to those shown in Figures 7.1 to 7.4. The difficulty in doing this was that storing images that contain the exact k-mer context tables needed to be lossless, and thus .png files were used. However, this wiped out the gains made in the compression algorithm. The more compression that could be obtained in the encoding phase meant a larger k-mer context table that had to be stored and transmitted to the decoder.

While, in order to recover the sequence accurately, the k-mer context tables used by the encoder must be the exact same ones used by the decoder, the k-mer context tables did not have to *precisely match* the sequence data when used by the encoder. This insight led to an attempt to compress the k-mer context tables using a *lossy* image compression technique such as .jpg. The idea here was that, before encoding even began, k-mer context tables would be determined and then saved as a .jpg image. This .jpg image would then be immediately converted back into the k-mer context tables. However, these reconverted tables were not exactly equal to and were only an approximation of the original k-mer context tables due to the lossy nature of .jpg compression. However, as long as the *encoder* used these recovered tables from the .jpg image to compress the DNA sequence, the .jpg format, which was more compressed than .png, could then be used to store the k-mer context tables since the decoder would use the same recovery process from the .jpg image as used by the encoder. This ensured exactly matching k-mer context tables at both the encoder and the decoder. However, in practice, it was observed that compression of the .jpg image beyond that of the .png image was traded off in degraded compression performance of the sequence itself because the k-mer context tables were not as "fine tuned" as they originally were.

Another way to overcome the storage problem for the k-mer context tables used by the encoder was the idea of "cross pollination." In other words, perhaps the

k-mer context tables could be *standardized* for all sequences (or at least all sequences of a certain classification) that are encountered by the encoder. If this could be successfully done, then the k-mer context tables could just be considered part of the compression algorithm itself and would thus not have to be compressed and stored with the compressed sequence output.

This idea was tested by applying the k-mer context tables taken from one bacterium genome to all the other ones. The results of this approach can be seen in Table 7.3, where the k-mer context tables from *S. aureus* were applied to all the bacterial genomes, and Table 7.4, where the tables from *P. gingivalis* were used. As can be seen from these results, the compression quality in doing this degrades considerably and seems to actually make matters *worse* in many cases when the *P. gingivalis* tables are used. The best bit rate encountered by any cross-pollination attempt was 0.98 bits per bit. These results seem to be close to those obtained with the adaptive approach, and that is because a similar problem is here lurking in the background. Even though an omniscient approach is utilized when doing cross pollination, the k-mer context tables can no longer be as precisely fitted to the sequence which they are used to compress. For example, if the tables of *S. aureus* are used as the standard, they may not contain k-mers that will be encountered when trying to encode *T. thermophilus*, and thus the assumption must be made once again that *any* possible k-mer might be encountered during encoding. As with the adaptive approach before, this limits the amount of compression that can be achieved because now any k-mer can be expected and cannot necessarily be ruled out from the start.

k	*S. aur.*	*E. coli*	*M. tub.*	*P. gin.*	*P. aer.*	*S. pyo.*	*T. the.*	*A. mar.*
2	0.95	1.04	1.13	1.04	1.14	0.98	1.18	1.03
3	0.95	1.04	1.12	1.04	1.12	0.98	1.16	1.03
4	0.95	1.04	1.13	1.04	1.13	0.98	1.16	1.03
5	0.94	1.05	1.15	1.05	1.15	0.98	1.18	1.04
6	0.94	1.05	1.14	1.05	1.14	0.98	1.17	1.04
7	0.93	1.07	1.17	1.07	1.17	0.99	1.2	1.06
8	0.91	1.09	1.15	1.09	1.15	1.01	1.15	1.08
9	0.88	1.06	1.06	1.07	1.05	1.03	1.05	1.06
10	0.86	1.03	1.01	1.03	1.01	1.02	1.01	1.03
11	0.88	1.01	1	1.01	1	1.01	1	1.01
12	0.93	1	1	1	1	1	1	1

Table 7.3: Bit rate results of omniscient arithmetic coding for k-mers of size 2 to 12 on the eight bacterial genomes using only the k-mer context tables obtained from *S. aureus* to compress all bacterial genomes. (The *S. aureus* results do not match those from Table 7.2 because these results represent context tables where *S. aureus* was "cross-pollinated" with itself since every k-mer now had to be treated as possibly encounterable by the encoder.)

k	*S. aur.*	*E. coli*	*M. tub.*	*P. gin.*	*P. aer.*	*S. pyo.*	*T. the.*	*A. mar.*
2	1	1	1.01	0.99	1.01	1	1.01	1
3	1.01	1	1.01	0.98	1.01	1	1.02	1.01
4	1.01	1.01	1.02	0.98	1.02	1.01	1.04	1.01
5	1.01	1.01	1.03	0.98	1.02	1.01	1.05	1.01
6	1.01	1.01	1.03	0.97	1.02	1.01	1.05	1.02
7	1.02	1.02	1.04	0.96	1.03	1.02	1.06	1.03
8	1.06	1.05	1.07	0.93	1.06	1.05	1.09	1.06
9	1.07	1.07	1.07	0.87	1.06	1.07	1.07	1.07
10	1.03	1.03	1.02	0.85	1.02	1.03	1.02	1.03
11	1	1	1	0.9	1	1	1	1
12	1	1	1	0.94	1	1	1	1

Table 7.4: Bit rate results of omniscient arithmetic coding for k-mers of size 2 to 12 on the eight bacterial genomes using only the k-mer context tables obtained from *P. gingivalis* to compress all bacterial genomes. (The *P. gingivalis* results do not match those from Table 7.2 because these results represent context tables where *P. gingivalis* was "cross-pollinated" with itself since every k-mer now had to be treated as possibly encounterable by the encoder.)

7.6 Conclusion

As with almost every data compression technique, the context-based arithmetic coders explored here came with trade-offs. While the k-mer context tables, which quantify the frequency with which certain k-mers occur in a sequence, do likely represent some key part of the information structure of a DNA sequence, k-mer counts *alone* could not seem to provide an avenue for significant compression. Every method and variation attempted came with trade-offs that seemed to limit any potential gains those variations might have offered. For the omniscient approach, the drawbacks of any one adjustment made to increase compressibility ended up undercutting the benefits of the omniscient approach as a whole. While *some* degree of compression can be achieved, and this is not insignificant, something more needs to be added to the mix, to the known information given to the encoder, which will describe the information structure of DNA accurately enough to allow it to be exploited for the benefit of substantial compression results.

Chapter 8

Conclusion

This thesis represents an exploration. It is known that DNA contains information. This information is used by every cell in every living organism to function properly. Due to advances in communication systems and data compression, many methods for quantifying information have been discovered and used to great benefit. It would be logically assumed, therefore, that these methods could be used to quantify and understand the information contained in DNA sequences. The major question explored in this thesis, then, was this: *how*?

There are many fundamentally biological means to understand what DNA contains. The endless annotation files provided with any genome attest to the fact that there are ways to discover components of the information contained in DNA. The question pursued was whether methods which relied *only* upon mathematical models—models of probability and correlation—could be used to quantify and understand the way that DNA's information was structured without recourse to biological methods and techniques.

When constructing the models by which to quantify the information in DNA, it was quickly realized that there were simply too many variables which could be adjusted and tuned such that an exhaustive search would be impossible. At times it was difficult to know which dials to turn and which could result in the most

beneficial outcomes. Therefore, this investigation took on a meandering nature, trying this technique and this model and that technique and that model, usually guided by past knowledge and intuition, in a attempt to see what the results would yield and whether anything of significance would be observed. This study leaves many stones unturned, from which much future work could be derived.

One particular observation does rise to the top when considering the results that have been obtained throughout this investigation: the biological information stored in DNA is not simple. Though biology has made great strides in understanding the nature and function of the DNA found in the genomes of many organisms, DNA remains a complex entity with a difficult structure to identify. The key to unlock the secrets to its information structure remains elusive.

This investigation did demonstrate, however, that some inroads, minor though they may be, could be made. As a baseline, estimation theory showed that even basic probability models could predict bases in DNA sequences with a nominal accuracy that was better than random guessing. Though basic estimation theory models were not successful at accurately predicting DNA bases, they showed that some level of correlation between bases in DNA sequences was quantifiable and exploitable. Attempts to predict DNA bases using the AMI profile, a technique shown to be relevant in analyzing DNA sequences, showed again that some quantifiable structure was present in DNA sequences, and with more sophisticated means, this structure could be exploited, but models based purely on joint probabilities, regardless of the sophistication, were not enough to substantially quantify the behavior of the information stored in DNA's sequences.

Employing *some* information known from the biological study of DNA sequences was shown to be useful. Since it is known that DNA is biologically interpreted in triplets, when the analysis takes account of the DNA code *qua*

triplets, the AMI profile can identify some unique features and correlations not detected by other means. Furthermore, using various machine learning methods, coding and noncoding regions could be differentiated, indicating that the AMI profile, as a probabilistic model, does capture enough of the information pattern to be able to indicate the coding or noncoding function of a segment of DNA. The AMI profile, along with principal component analysis, was also useful for identifying which DNA segments were naturally part of the human genome and which were created by various models. The AMI profile as a metric has the potential, when combined with other machine learning techniques and some biologically relevant information, to expose correlations that are due to the structure of the information contained in the various sequences.

Finally, using arithmetic coding as a method of quantifying the information structure of DNA sequences absent the use of the AMI profile demonstrated that DNA sequences contain structure that is quantifiable by the grouping of particular bases (k-mers) and which can be minimally exploited to form a compression technique for DNA sequences. This showed that, while DNA sequences generally contain features and redundancy that can be exploited, their complexity is such that merely considering the order of bases in k-mer groups was not sufficient in and of itself to provide the catalyst for significant compression and thus, by extension, significant grounds for thinking that the information structure of DNA sequences has been deeply grasped.

The results of this investigation leave ripe opportunities for future work. As previously discussed, the results based on estimation theory and the results from prediction of bases using the AMI profile could be further examined to determine the effect that the distribution of bases within a sequence has on prediction ability. This could include examination of regions where the high marginal probability of a single

base causes these methods to be unable to predict any other base. These results could also be further extended to explore the potential directional relationships between the positions of bases, where the specific order is taken into account as opposed to *merely* the separation distance between bases. The estimation theory, base prediction, and triplet AMI profile results could also be studied in combination to assess the effectiveness of a multiple-model approach to base prediction, rather than testing each method in isolation. Using all of these metrics together may increase the chances for accuracy in base prediction. Additionally, the reasons for the presence of unique features in the triplet AMI profiles for human chromosomes could be examined using more sophisticated sequence alignment techniques.

The results stemming from machine-learning techniques could also be further explored. Certainly, the sophistication of the machine-learning methods could be increased. Neural networks that contain more layers of neurons, support vector machines that have different kernels, and more dimensions in the principal component analysis as well as other machine-learning methods such as so-called "deep learning" could be employed to study whether more sophistication in these techniques can capture the correlation necessary to describe the information structure of DNA and differentiate between its various types of regions. This could be applied to coding and noncoding regions, as was done extensively in this study, but other types of regions (e.g., exons and introns in eukaryotic coding regions) could be explored as well. Other types of AMI profiles besides the standard one, such as the single-base and triplet AMI profiles, could also be used as inputs to all of these methods to see if extra information and model complexity yields more in terms of accuracy.

Lastly, the arithmetic coding procedure could be supplemented by studying more creative means to compress the k-mer context tables. Since context-based

arithmetic coding is highly successful for compressing a DNA sequence given the sparsity present in k-mer tables, means to exploit that sparsity in compressing them would be valuable. This could include an investigation of whether a type of run-length coding could compress the k-mer tables. Furthermore, not only were the k-mer context tables sparse, but it was also observed that they exhibited a periodicity upon visual inspection. The existence of this periodicity could be further explored as to both explaining its cause and also exploiting its presence for compression and recovery of the k-mer context tables. Finally, all the studies contained in this thesis could always be tested on more DNA sequences from other organisms to determine whether or not the results presented in this investigation, mostly focused on human and bacterial DNA, are representative of other known genomes.

As mentioned, despite the results that indicated that the mathematical and probabilistic models employed in this thesis did capture at least *some* measurable amount of the structure of the information contained in DNA sequences, the key to deeply unlock the information structure of DNA remains elusive. The conclusions of this investigation are but hints pointing in that direction.

Bibliography

[1] M. Bauer, S. M. Schuster, and K. Sayood, "The average mutual information profile as a genomic signature," *BMC Bioinformatics*, vol. 9, Jan. 2008.

[2] C. E. Shannon, "The mathematical theory of communication," *Bell System Technology Journal*, vol. 27, Jul. 1948.

[3] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann, 4th ed., 2012.

[4] L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley, 1991.

[5] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H. Seidel, "Image compression with anisotropic diffusion," *Journal of Mathematical Imaging and Vision*, vol. 31, Apr. 2008.

[6] F. J. M. Mojica, C. Díez-Villaseñor, J. García-Martínez, and E. Soria, "Intervening sequences of regularly spaced prokaryotic repeats derive from foreign genetic elements," *Journal of Molecular Evolution*, vol. 60, Feb. 2005.

[7] UCSC Genome Browser, "Human (grch38/hg38) chromosome 15." `https://www.genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chr15%3A17000000%`

2D21500000&hgsid=2031449294_jYoTe4LdjEqnGL9Y2gmzPmZTw8j7. [Online; accessed 16-March-24].

[8] UCSC Genome Browser, "Human (grch38/hg38) chromosome 13." https://www.genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chr13%3A16000000%2D20000000&hgsid=2031505218_9Wiky0EsagSBRVtkdBkkh94kEqr9. [Online; accessed 16-March-24].

[9] UCSC Genome Browser, "Human (grch38/hg38) chromosome 1." https://www.genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chr1%3A119563094%2D127421397&hgsid=2031543884_gw8ZvEUTpmTFKLsfu4vjMzTyFunS. [Online; accessed 16-March-24].

[10] UCSC Genome Browser, "hg38 centromere locations (gj212036.1)." https://www.genome.ucsc.edu/cgi-bin/hgc?hgsid=910333933_6wTYa5qZ6LXGmaM2YPwWbWANQX6A&c=chr15&l=16999999&r=21500000&o=17083673&t=17498951&g=centromeres&i=GJ212036.1, 2014. [Online; accessed 16-March-24].