

Computer Science Honours  
Research Report  
Compression and Computational Gene Finding

Greg Kempe

1 November 2002

## Abstract

Gene sequences in DNA are punctuated with regions of “junk” that are not used during expression of the gene. Identifying these regions is a complex task and many computational techniques have been devised to solve the gene finding problem. DNA can be described as a sequence over an alphabet of four letters  $\{A, G, C, T\}$  and so a sequence can be considered as a text written in some language. Gene finding methods that use this linguistic approach have been very successful. This research investigated the feasibility of a linguistic approach based on compression.

Compression algorithms have been successfully used in the linguistic analysis of human texts, including the differentiation between texts based on language and author. The useful and junk parts of a DNA sequence can be substantially different and can be viewed as two different languages. The research used a compression-based measure of entropy to attempt to differentiate between coding and non-coding regions in DNA sequences. Additionally, the same measure was used to compare genes from different families and parts of different species' genomes.

The results show that the measure is unable to identify the often subtle variations in genomic data, preventing it from effectively discriminating between different types of DNA sequences. It is reasoned that the measure is not suited to genomic data in general and has limited applications in the field of bioinformatics. Possible reasons for this include the small alphabet and correspondingly limited range of “words” in the sequences. This report details the research performed and discusses the implications of the results.

### **Acknowledgements**

I would like to thank my supervisor, Scott Hazelhurst. His guidance and encouragement are much appreciated. I would also like to thank Anton Bergheim and Lauren Rota for their contribution to my understanding of the biological aspects of the work.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 DNA and Computational Gene Finding . . . . .	2
1.3 DNA as a Language . . . . .	2
1.4 Data Compression . . . . .	3
1.5 Research Overview . . . . .	4
1.6 Results . . . . .	4
1.7 Conclusion . . . . .	5
<b>2 Background and Related Work</b>	<b>6</b>
2.1 Introduction . . . . .	6
2.2 Biological Background . . . . .	6
2.2.1 Cells, Genes and DNA . . . . .	6
2.2.2 The Structure of DNA . . . . .	7
2.3 Computational Gene Finding . . . . .	8
2.3.1 Signal-Based Approaches . . . . .	8
2.3.2 Compositional Approaches . . . . .	9
2.3.3 Hybrid Approaches . . . . .	10
2.4 Language and DNA . . . . .	11
2.4.1 The Linguistics of DNA . . . . .	11
2.4.2 Compression and Linguistic Analysis . . . . .	12
2.5 Data Compression . . . . .	12
2.5.1 The LZ77 Compression Algorithm . . . . .	13
2.5.2 The DEFLATE Compression Algorithm . . . . .	14
2.6 Conclusion . . . . .	14
<b>3 Research Method</b>	<b>16</b>
3.1 Introduction . . . . .	16
3.2 Research Question . . . . .	16
3.3 Motivation . . . . .	17
3.4 Research Method Overview . . . . .	17
3.5 Compression and the Entropy Measure . . . . .	18
3.5.1 Entropy Value Calculation and Use . . . . .	18
3.5.2 The Relative Entropy per Character Measure . . . . .	18
3.5.3 Alternative Compression Measures . . . . .	19
3.6 Experiments . . . . .	20
3.6.1 Cross-Region Comparison . . . . .	20
3.6.2 Independent Region Comparison . . . . .	21

3.6.3	Gene Family Comparison . . . . .	21
3.6.4	Genome Comparison . . . . .	22
3.7	Conclusion . . . . .	22
<b>4</b>	<b>Results</b>	<b>23</b>
4.1	Introduction . . . . .	23
4.2	Cross-Region Comparison . . . . .	23
4.2.1	ACU08131 . . . . .	23
4.2.2	AGU04852 . . . . .	24
4.2.3	HUMATPGG . . . . .	24
4.3	Independent Region Comparison . . . . .	24
4.4	Gene Family Comparison . . . . .	29
4.5	Genome Comparison . . . . .	29
4.6	Conclusion . . . . .	30
<b>5</b>	<b>Discussion</b>	<b>32</b>
5.1	Introduction . . . . .	32
5.2	Cross-Region Comparison . . . . .	32
5.2.1	Discussion . . . . .	32
5.2.2	Additional Results . . . . .	33
5.2.3	Conclusion . . . . .	33
5.3	Independent Region Comparison . . . . .	34
5.3.1	Discussion . . . . .	34
5.3.2	Conclusion . . . . .	34
5.4	Gene Family Comparison . . . . .	34
5.4.1	Discussion . . . . .	34
5.4.2	Conclusion . . . . .	35
5.5	Genome Comparison . . . . .	35
5.5.1	Discussion . . . . .	35
5.5.2	Conclusion . . . . .	35
5.6	Reasons for the Results Obtained . . . . .	36
5.7	Contribution and Future Work . . . . .	36
5.8	Conclusion . . . . .	37
<b>6</b>	<b>Conclusion</b>	<b>38</b>
6.1	Introduction . . . . .	38
6.2	Research Overview . . . . .	38
6.3	Results . . . . .	39
6.4	Conclusion . . . . .	39
	<b>References</b>	<b>40</b>

# Chapter 1

## Introduction

### 1.1 Introduction

Recent surges in the sequencing of various genomes have resulted in large bodies of DNA data that still have to be analysed. Once understood, the real-world applications of such data are far-reaching, ranging from cures for diseases and pest control to modified foodstuffs and improved quality of life. Even without any applications, such knowledge contributes to humankind’s understanding of the complexities of life—an admirable goal in its own right.

Finding methods for storing and analysing such large quantities of biological data have presented a number of challenges, many of which have yet to be met [Batzoglou *et al.* 1998]. One of these is the gene finding problem: given the deoxyribonucleic acid (DNA) of an organism, a potentially huge amount of data, how does one decode its meaning? Which pieces of it are used and for what purposes? Attempts at answering questions like these have led to the development of the fields of computational molecular biology and bioinformatics; combinations of the fields of molecular biology, mathematics and computer science.

This document discusses the results of research into a new method for identifying useful regions in DNA. A sequence of DNA can be considered as a text written in some language. The research applied techniques used for the linguistic analysis of human-language texts to these sequences. More specifically, compression has been successfully used in linguistic analysis and its applications in certain areas of bioinformatics, including computational gene finding, were explored.

The research used a compress-based entropy measure in an attempt to gauge the “closeness” (in informational content) or similarity of bodies of DNA data. It investigated the feasibility of using the measure to identify coding and non-coding regions in DNA sequences, distinguish between genes from different families and identify the similarities between entire genomes. The results showed that the entropy measure is not capable of determining the “distance” in informational content of the sequences with sufficient accuracy to produce meaningful values. The research concludes that the measure is not suited to differentiating between bodies and types of DNA data. Furthermore, the measure does not have the capabilities required to produce informative values for DNA data in general. As a result, it is unlikely it would have applications in any other field of bioinformatics.

This chapter presents a general outline of the research. Section 1.2 gives a brief introduction to the basics of DNA and computational gene finding. The language-based approach to DNA, along with a discussion of compression as a tool for linguistic analysis, is presented in Section 1.3. Section 1.4 discusses data compression in general and Section 1.5 gives an overview of the research. The outcome of the research and the reasons for the results are discussed in Section 1.6. Finally, Section 1.7 concludes and outlines the structure of the rest of the document.

## 1.2 DNA and Computational Gene Finding

The genome of an organism is the total genetic information of that organism and is stored in its cells in the form of DNA. The DNA is broken into sections—genes—that are used by the cells as blueprints for the various products they must produce in order to function, such as proteins. Other regions in the DNA control which genes are used and when, as well as how the multitudes of genes and their products interact.

DNA is comprised of four nucleotides which are represented by the letters A, G, C and T, and a strand or sequence of DNA can be considered simply as a string of these letters [Hunter 1993]. Combinations of nucleotides are recognised by mechanisms in the cell and usually code for some amino acid. Depending on their context, some combinations may instead act as signals to the cell, for example telling it where a gene begins and ends. In organisms like humans and other vertebrates, the useful parts of genes are interspersed with “junk” DNA—sequences, referred to as non-coding regions, that are ignored by the cell and are removed before use. One aspect of the gene finding problem is identifying these coding and non-coding regions [Haussler 1998].

The complexity of genomic data makes this task very difficult. Classical biological approaches require a great deal of time and so over the past two decades computational techniques, which are much faster but somewhat less accurate, have been developed. A number of these techniques have been very successful but the gene finding problem as a whole is far from solved. Indeed, “there is little doubt that Nature can construct an enigma of the kind which human ingenuity may not resolve.” [Pevzner 2000, p. 135]

Two basic approaches to gene finding can be taken. Content-based or compositional approaches study the inherent differences between coding and non-coding regions [Haussler 1998; Guigó 1998]. Since the latter are not used, they are, to a certain degree, exempt from the laws of natural selection and tend to mutate and change without much control. The former region, however, is far more controlled by natural selection. Mutations, especially dangerous ones, tend to be prevented and changes occur less often. As a result, non-coding regions exhibit a far more random distribution of nucleotides. Content-based gene finding methods analyse these differences and use them in an attempt to differentiate between the two regions.

Signal-based approaches attempt to use the natural signals in a DNA sequence to distinguish between the two regions [Haussler 1998; Dong and Searls 1994]. The mechanisms in the cell that read and understand genes use signals to identify the non-coding regions in order to remove them. These signals occur at either end of non-coding regions—the splice junctions—and the range in the signals means that detecting them is not straight forward. Additionally, the same signals may occur elsewhere in non-signalling capacities.

The differences and complexities of gene sequences make neither of these methods particularly successful on their own. Instead hybrid approaches that combine both of the methods, and thereby capture the interactions between them, have been developed with good results [Bat-zoglou *et al.* 1998]. However, in general they do not perform well with atypical DNA sequences or on sequences that have alternative splicing patterns [Haussler 1998]. The more factors that can be identified in a sequence the better the chances of accurately classifying its regions. Developing new methods and improving old ones is therefore an important area of research.

## 1.3 DNA as a Language

One of the more recent and highly successful hybrid gene finding methods is that by Dong and Searls [1994]. They consider a DNA sequence to be a sentence in some language. Valid sentences in the language—genes—are dictated by a grammar which imposes various rules and restrictions, in the same way that English grammar dictates the well-formedness of English phrases, sentences, paragraphs, etc. Dong and Searls [1994] built a flexible grammar and developed an algorithm

that is able to determine if a given DNA sequence matches that grammar and what its constituent parts are.

The success of their method indicates that such a linguistic approach to gene finding has merit. Due to the significant differences between coding and non-coding regions, they could be considered as two different languages. A method for identifying the two regions based on these language may be useful.

A relatively simplistic approach to the linguistic analysis of human language texts is that taken by Benedetto *et al.* [2002]. By compressing a number of texts and analysing the results they were able to identify the authors and language of each text. Their technique involved defining the “distance” between two texts, which is effectively the distance between their informational content. The greater the distance, the less related the two are. Their definition is based on the changes in performance that occur when part of one text is compressed along with the other text. In this way, the difference in informational content is established without the texts needing to be interpreted or understood in any way.

Such an approach may have merit when used to identify the two “languages” of coding and non-coding DNA sequences. It is not the languages themselves that are of interest in such a case, but rather the boundaries between them. Additionally, the approach could be applied to sets of genes to determine how different their informational content is. Genes with similar functions tend to share certain characteristics that the method may identify. Furthermore, when considering a species’ genome in its entirety it may be possible to use the approach to gauge how different two genomes are to each other. It is these aspects of the method that the research explores.

## 1.4 Data Compression

The method used by Benedetto *et al.* [2002] used compression to evaluate the entropy of a sequence, a discussion of which follows.

Data compression, a form of encoding, is used to express a body of data more succinctly than its original form. Often, especially in contexts like text compression, this is achieved by exploiting repetitions in the data. Instead of repeating long sequences that occur more than once, a code is used in lieu of the sequence (obviously the code should be shorter than the sequence it represents). For instance, the code may simply refer back to an occurrence of the sequence earlier in the data.

Compression is a broad field and so only that pertaining to textual data, which is most pertinent to this research (as opposed to, say, image or audio data), is discussed in this document.

There are a number of methods of keeping track of frequently used substrings. One such approach is the so-called dictionary method used in the LZ77 algorithm by Lempel and Ziv [1977]. The method maintains a dictionary of the past substrings encoded and a look-ahead window of the data about to be encoded. During compression it looks for the longest prefix in the look-ahead window that matches an entry in the dictionary. The algorithm only looks for entries that have occurred recently, which improves performance: the most frequently used sequences are encoded using less data than ones with infrequent use. Every string therefore has an optimal encoding which differs from other strings, and it is this difference that is of interest.

The differences between coding and non-coding regions that are identified by content-based gene finding programs may be sufficient for each region to have a different optimal encoding. In terms of LZ77, as the buffer moves out of one region and into the next, their differences make the likelihood of finding recently-used prefixes in the dictionary minimal, resulting in a poor level of compression. As the compressor moves along the new region the dictionary is updated and the algorithm adapts, finding an encoding that is optimal for the new region and compression improves. When the region changes again, performance decreases until the dictionary adapts etc. Similarly, when considering two bodies of DNA data such as genes or

genomes, the characteristics of each individual will result in different optimal encodings. It is these putative differences, and the resulting changes in performance, that the research explored.

## 1.5 Research Overview

The measure of distance defined by Benedetto *et al.* [2002] evaluated the similarity of two texts based on their optimal encodings. Viewed in terms of a DNA sequence, it is known that the body being examined consists of multiple “texts” (regions) but the boundaries between them are not. Instead of identifying the two languages of the texts (in this case coding and non-coding regions), these boundaries must be identified.

When moving from one region into the next, the encoding that is optimal for the first will not be optimal for the second, resulting in a change in compression performance and dictionary usage. The research hoped to identify the regions in a DNA sequence by compressing the sequence and analysing the results. If the two regions have sufficiently different optimal encodings then a change in region may be indicated by a drop in performance. These performance fluctuations could then be used in an attempt to identify coding and non-coding regions in a DNA sequence.

In addition to this primary aim, the research also used the putative differences in optimal encodings to attempt to differentiate between genes with similar functions and genes with entirely unrelated functions. Each of these *gene families* have specific functions and members share certain characteristics and similarities [Bergheim 2002a]. The research investigated the possibility of using a compression-based measure to identify these similarities (or lack thereof) and judge how related two sequences are.

Finally, also as a secondary investigation along with the gene family comparison, the research investigated using the same measure to judge the distance in informational content between two genomes. Like genes from the same family, the genomes of related species have certain similarities. The measure attempted to identify these similarities and determine how alike the two genomes, and hence the two species, are.

As a basis for comparison, these last two experiments compared genes from the same family, and sections of the same genome. These values were used as ground values as they indicated what results could be expected for similar sequences, giving an insight into the values that could be expected for very different sequences.

## 1.6 Results

The work involved defining a compression-based measure of entropy, based on the measure used by Benedetto *et al.* [2002]. The measure was used to produce a number of entropy values which were compared with expected results as well as analysed for other, unexpected patterns.

The first application of the measure was to the coding and non-coding regions of genes. The resulting entropy values showed none of the expected changes at the boundaries between the regions, neither did they show any other pattern that could be used to identify the boundaries. There were some small differences in the entropy values within the two regions, differences which were confirmed by additional experimentation. However, the differences were small and unreliable. Consequently, the results led to the conclusion that the measure was not suitable for identifying the two regions, either by identifying their boundaries or the patterns of the regions themselves.

The second application of the measure compared genes from different families. The entropy values produced when comparing genes from the same family varied just as much as values for genes from different families. As with the region comparison, there were no additional interpretations of the results that could lead to a decision regarding the relatedness of the genes. Instead of producing two distinct sets of values, the measure produced a wide range of values

for both sets of data with which it was not possible to gauge the relatedness or similarity of the genes.

The final application compared parts of entire genomes. Again, even when comparing parts of the same genome to each other—a procedure that should have produced a set of low entropy values—a wide range of values resulted. Similar values were produced when comparing parts of different genomes. The range in values and lack of any definite pattern indicated that the measure could not be used to determine the similarity of genomes.

The results, when considered as a whole, lead to the conclusion that the measure is not only unsuitable for the tasks attempted, but also for DNA data in general. The measure is not able to capture the subtleties of genomic data, preventing it from identifying the variations in the data needed to separate the different classes of sequences presented to it. Potential reasons for this include the small alphabet of DNA (only four letters), the small range in “words”, and the large window used by the compression algorithm.

The research has narrowed the field of potential bioinformatic tools, albeit by a small margin. In doing so it has shown that the compression-based entropy measure, while applicable to natural language texts, is not suited to the more limited language of genomic data.

## 1.7 Conclusion

The problem of separating the useful and “junk” regions in DNA sequences is not fully solved. A number of methods have been devised that, while successful, are still not entirely reliable especially for uncharacteristic sequences. These methods either identify naturally occurring signals in the DNA or use the differences in content between the two regions to identify them. A more advanced method is to use a linguistic metaphor and apply grammar-based techniques to the sequence. The success of such methods warrant further investigation into the use of linguistic techniques in the bioinformatics field.

The research investigated the viability of a technique based on the optimal encoding of a sequence as determined by a compression algorithm. By comparing the optimal encodings of different bodies of DNA data, the work attempted to identify the different bodies and determine how far apart, in terms of information content, they are. The results indicate that the measure is not able to extract sufficient information from the sequences to capture often subtle differences. Consequently, the research concludes that the measure is not suited to DNA data in general.

The rest of this document discusses the research in detail and is structured as follows. Chapter 2 gives a more detailed introduction to the biological side of the research and reviews existing computational gene finding techniques. It also discusses data compression and the consideration of DNA as a language, and how the two relate. Chapter 3 presents and motivates the formal research question and discusses the method used to answer it. The results produced by the work are presented in Chapter 4 and their meaning and implications are detailed in Chapter 5, along with the contribution of the work and possible areas of future research. Finally, Chapter 6 concludes the document.

## Chapter 2

# Background and Related Work

### 2.1 Introduction

The field of computational molecular biology is involved in finding computer-based solutions to such problems as protein folding, DNA matching and gene finding. This latter problem has been the focus of much research over the last 20 years and, while having produced a number of useful results, is still an open problem.

The methods for solving the problem developed so far have tended to take a couple of approaches, the most modern of which is a linguistic approach. By treating a DNA sequence as a form of language, well-studied tools such as grammars and finite state automata can be applied to it, and have been with much success [Dong and Searls 1994]. This has paved the way for other linguistic-based techniques. Other existing gene finding methods are either content-based, signal-based or a mixture of the two [Haussler 1998]. Content-based approaches analyse the content of the gene itself, while signal-based approaches focus instead on the signals within the genes, such as those at the boundaries between the coding and non-coding regions. The hybrid approaches utilise both techniques and generally yield more accurate and reliable results.

This chapter discusses existing gene finding techniques and their results, and gives a background to compression and its use in linguistic analysis. Its structure is as follows. A brief introduction to the biological background of genes and DNA is given in Section 2.2, followed by a discussion of computational gene finding and a review of existing methods in Section 2.3. Section 2.4 covers the linguistic nature of DNA and the use of compression in the analysis of languages. A more detailed discussion of compression, and of the algorithms pertinent to this research, is given in Section 2.5.

### 2.2 Biological Background

#### 2.2.1 Cells, Genes and DNA

Very generally, life as a whole can be considered in two groups: *prokaryotes* and *eukaryotes*. Prokaryotes, such as bacteria, are single-celled organisms whose cell does not contain a nucleus. As such, they are relatively simple. Eukaryotes are multi-cellular organisms whose cells contain a nucleus in which genetic information is stored. Eukaryotes are a far more complex form of life and range from humans and other vertebrates to plants and fungi [Hunter 1993]; it is these forms of life that the research is interested in.

The DNA in an organism's cell is the store for all the genetic information, or genome, of that organism. All the organism's cells contain the same DNA but different types of cells use different parts, resulting in the many and varied forms of cells found in multi-cellular organisms. DNA is comprised of two complementary strands of molecules that are bonded together and form the well-known double helix shape. The strands are directed, each with a head end and a tail end

that is bonded with the opposite end of the complementary strand. Thus, when interpreting or reading a strand, the complementary strand is always read in the opposite direction [Hunter 1993].

The strands are a sequence of four nucleic acids, called *nucleotides* or *bases*: adenine (A), thymine (T), cytosine (C) and guanine (G). The two strands are joined by bonds between these bases, adenine bonding exclusively with thymine and cytosine exclusively with guanine. Due to this natural pairing, nucleotides are often called *base-pairs*.

A DNA sequence can be well over a million base-pairs long and only parts of it are used by the cell at once. These subsequences, which can roughly be considered genes, produce some form of product which may either be used by other parts of the cell or affect the way in which other subsequences are used. Not all parts of a gene are useful, however, a characteristic unique to eukaryotes. Interspersed among the useful DNA are sequences of “junk” DNA, termed *introns*, that are not used by the cell. These introns must be removed from the surrounding *exons* (the areas that code for something useful and are used by the cell) before use.

To produce the product a gene codes for, a process referred to as *expression* of the gene, the sequence must undergo a number of transformations. The first of these is transcription, in which the cell creates a copy of the gene being expressed resulting in ribonucleic acid (RNA)—a base-for-base copy of the sequence whose only difference from DNA is that uracil (U) takes the place of thymine. The RNA is then processed by the cell’s spliceosomes which remove the introns and splice the resulting exons together, forming messenger RNA (mRNA). Translation then occurs in which the mRNA is used to create chains of amino acids which are manipulated to form the final protein. More detail on DNA, genes and their expression is given by Hunter [1993].

### 2.2.2 The Structure of DNA

One can represent DNA sequences as strings over the alphabet  $\{A, C, G, T\}$ , where each letter corresponds to a nucleotide. Sequences are not considered base-by-base, instead nucleotides are considered in groups of three. Each triplet of nucleotides, called a *codon*, corresponds unambiguously to one of the amino acids used in protein generation. The coding for an amino acid is not unique, however; the amino acid alanine, for example, is represented by the codons GCT, GCC, GCA and GCG. Coding is further complicated by the fact that as nucleotides are considered in triplets, there are three possible places to start parsing a segment of DNA. The sequence ...AATGCGATAAG... could be considered using any of the following *reading frames*: ...AAT-GCG-ATA..., ...ATG-CGA-TAA... or ...TGC-GAT-AAG... Not all reading frames are useful and finding meaningful frames is not a trivial task; indeed, some sequences code for different proteins depending on the reading frame [Hunter 1993].

Not all exons are translated into proteins. The exons at both ends of a gene sequence contain codons that act as signals to the various expression mechanisms. These include start codons (ATG) and stop codons (TAA, TAG and TGA) that delineate the boundaries of the sequence. Additionally, at the start of a gene sequence (termed the 5’ end) there are various promoter signals that are used to control how and when a sequence is expressed, and the end of a sequence (the 3’ end) contains still further non-coding exons. There are also signals within the sequence, such as at exon/intron boundaries (called *donor splice sites*) and intron/exon boundaries (*acceptor splice sites*).

In this research, the definition of a gene is that adopted by Batzoglou *et al.* [1998]: a single, contiguous region of DNA, combined with its accompanying regulatory signals, that codes for some gene product.

## 2.3 Computational Gene Finding

Recent large-scale sequencing efforts have produced huge amounts of genome data for a number of species, including humans [Pevzner 2000]. Ways of analysing and making sense of this data are of great importance, and one aspect of the field of computational molecular biology is focused at finding ways of doing this. After a genome has been sequenced it is only understood at the simplest level: only the structure of the two DNA strands is known. Further understanding requires identifying the individual genes in the genome, what they code for and how they cooperate—a process referred to as *annotation*. Computational gene finding is the application of computers to the location of genes in a genome and the classification of coding and non-coding regions [Hunter 1993; Pevzner 2000].

Due to the complex nature of DNA and its function, finding all the signals and content of which a gene is comprised is not a trivial task. Biological approaches to gene finding, while more accurate and reliable than computational approaches, are often time consuming, expensive and require a great deal of biological expertise. The task is simplified greatly when researchers have access to bioinformatic tools that can perform at least part of the identification for them. Often these tools are used to identify likely gene sequences which can then be confirmed by biological means.

A number of computational methods have been devised to identify genes over the past 20 years, with varying degrees of success. Two basic approaches to the problem have been taken. Signal-based approaches attempt to identify and use the natural regulatory signals that occur in DNA. They effectively take the viewpoint of the biological mechanisms that use the signals and attempt to classify sequences accordingly. They identify introns and exons, for example, by finding splice junctions and stop and start codons. Compositional or content approaches, on the other hand, focus on the regions themselves, attempting to distinguish introns from exons by their differing characteristics and codon frequencies. Each approach has its benefits and drawbacks and neither alone can solve the gene finding problem. Instead, a number of hybrid methods have been developed that attempt to combine the strong points of both, with a much higher success rate.

### 2.3.1 Signal-Based Approaches

In the spliceosomes of a cell, biological mechanisms recognise signals in the gene sequence and use them to identify coding and non-coding regions. Signal-driven approaches attempt to identify these signals computationally and use them to separate introns from exons and identify the beginnings and ends of genes.

The signals focused on include those indicating donor and acceptor splice sites and start and stop codons. For instance, introns often start with GA and end with CG [Bergheim 2002b]. Additionally, there are longer but more weakly conserved (ie. they occur often but not always, or may be slightly modified) sequences of eight nucleotides at the donor splice site and sequences of four nucleotides at the acceptor splice site [Pevzner 2000]. While it may appear that these characteristics should make signal identification easy, all of these sequences may also appear in non-signalling capacities throughout both exons and introns [Stormo and Haussler 1994].

The simplest way of representing one of these signals is with a consensus sequence taken from a collection of sequences [Haussler 1998]. The consensus sequence is often calculated using the majority function and indicates the most frequent nucleotide at each position. This method is very simple and does not take into account the context of the signal or the frequencies of the nucleotides. Profiles, or position weight matrices, are slightly more advanced and assign frequency-based scores to each nucleotide in each possible codon position. If a sequence achieves a sufficiently high score when compared with a profile, it is considered to contain the signal [Pevzner 2000].

Profiles and consensus sequences on their own have had little success, mostly due to the

range and subtlety of signals and their interactions [Dong and Searls 1994; Pevzner 2000]. More success with signal finding has been achieved with machine learning techniques such as neural networks and advanced statistical techniques like Hidden Markov Models (HMMs). Both of these methods combine a number of signal measures and interpret them together, capturing the dependencies between them. An overall score is calculated and if it exceeds some threshold, the site is considered to contain a true occurrence of the signal [Pevzner 2000]. HMMs are a powerful technique that have been used with much success in hybrid approaches and are discussed in more detail in Section 2.3.3.

Choosing how to weight each signal can be complex and both methods involve a learning stage. A set of training sequences are used to guide the neural network or HMM towards optimal weightings. The major drawback of this is that the model can suffer from over-training, in which case it tends only to be useful for the training set. Additionally, the resulting model is often only useful for sequences similar to the training set; sequences that have different splicing characteristics may not be recognised by it.

The complexities of DNA signals mean that good models of them and reliable algorithms for finding them do not yet exist, and the understanding of their biological significance is not yet sufficiently advanced that signal identification alone can be used to find genes [Batzoglou *et al.* 1998; Pevzner 2000].

### 2.3.2 Compositional Approaches

Intron and exon regions tend to develop differently. Introns, being mainly unused, are free to mutate and change without much control or natural selection. However, a small change in an exon can result in significant biological changes and are generally discouraged by natural selection, especially if they are harmful [Batzoglou *et al.* 1998]. Indeed, the dual strands in DNA are a protection against such damaging mutations and cells even attempt to repair damaged sequences [Hunter 1993]. These differences in evolutionary control result in a number of statistically and biologically significant differences in nucleotide patterns between coding and non-coding regions [Hunter 1993; Bergheim 2002b]. Compositional approaches use these differences to characterise sequences and isolate the useful parts from the junk parts [Pevzner 2000].

One of the first content-driven methods for gene finding, still used by many modern algorithms, was the Testcode algorithm by Fickett [1982]. The algorithm measures the tendency of nucleotides in coding regions to “favour” certain codon positions, generally referred to as the *positional asymmetry* of the sequence. For instance, it was shown that T shows a strong positional preference in exons and much less preference in introns. This is biologically significant due to the tendency for the third nucleotide in a codon to change without altering the codon’s meaning, since it is the first two nucleotides that are most preserved [Hunter 1993]. Testcode also quantifies the *hextuple frequencies* of coding versus non-coding regions, which is the measure of how often a string of six nucleotides (comprising two codons) occurs throughout a region. Hextuple frequencies and positional asymmetry are two measures that have met with much success, as is indicated by the number of modern methods that use them.

The main drawback of the above statistical measures is their dependence on long sequences. In order to exhibit sufficiently different frequencies sequences must be at least 200bp (base-pairs) long [Fickett 1982]. This presents problems with groups of organisms like vertebrates which typically have exons of only 130bp [Pevzner 2000].

Another distinguishing feature between introns and exons is the difference in codon usage. Introns are far more likely to develop a random distribution of nucleotides than exons [Guigó 1998]. There are  $4^3 = 64$  possible codons so the probability of any one codon occurring in an intron, assuming a random distribution, is approximately  $\frac{1}{64} \approx 0.0156$ . The probability of the same codon occurring in an exon is very different and this bias in codon usage is often used as a content sensor.

Sequences common in specific coding regions may also be identified. For example, some

content sensors identify long sequences of the ALU codon that are common in human introns, but not exons. Others identify so-called *CG islands*, clusters of CG dinucleotides or CxG trinucleotides (where x represents any nucleotide) that tend to be more frequent at the start of genes than elsewhere [Haussler 1998]. These differences only apply to certain genomes, however, and are not always reliable.

As with signal sensors, HMMs have been used to bring multiple content sensors together with much success. Instead of grouping splice sites together, they are used to model the probability of one type of region (such as an exon) following another (such as an intron) given the surrounding content and signals; the score produced is used to classify the regions of a sequence [Pevzner 2000]. Neural networks have been similarly applied, also with a high degree of success, although the same training-related problems experienced with signal sensors still occur [Haussler 1998; Batzoglou *et al.* 1998].

As the above tests indicate, content measures can either be model-dependent or model-independent [Guigó 1998]. The former require that the results of the measure be compared with those of a sample DNA sequence that is already annotated. For instance, identifying exons based on high ALU repetitions requires that the number of repetitions in a new sequence is compared with those for a known sequence. To use these measures, DNA representative of the sequences being investigated must exist. Model-independent content measures, however, do not need such comparisons. They rely instead purely on the “universal” relative differences, such as positional asymmetry, that are the result of biological differences between coding and non-coding regions. These are more suited to new genomes for which comparison sequences are not yet known. An extensive review and comparison of both types of measure is given by Guigó [1998].

### 2.3.3 Hybrid Approaches

Often, there are several different features that can be used to classify sequences, none of which are sufficiently reliable on their own. For example, compositional approaches are good at identifying regions in general but tend to produce only low-resolution results for the boundaries between them. Signal finding approaches, on the other hand, return very definite boundary indications. Almost all modern computational gene finding methods combine both approaches to produce far more accurate and reliable results than are attainable using each approach on its own.

Hidden Markov Models, successful with both content and signal finding, are also very useful in bringing the two methods together [Kulp *et al.* 1996]. A Hidden Markov Model is a form of finite state automaton with probabilities associated with each state and transition. The probabilities determine the likelihood of being in a state or taking a transition from one state to the next, given some input. Using these, an HMM can generate the probability of a sequence being modelled by the HMM’s states and transitions.

For example in a gene finding context, the start state may correspond to the occurrence of a start codon, followed by a number of exon-intron pairs with donor and acceptor site signals indicating transitions, terminated by a final exon and a stop codon. The transition probabilities can be used to make the model more specific, for instance favouring a certain number of exons after a certain signal has been identified. This allows the interactions between signals and content to be captured, allowing for a more rounded and reliable prediction model [Kulp *et al.* 1996; Pevzner 2000].

Dong and Searls [1994] were the first to advocate a linguistic approach to the problem, also with a good degree of success. They use formal grammars to structure a “typical” gene in a manner similar to HMMs, using both content and signal indicators. A parser then processes an input sequence to see if it matches the gene grammar. The rules of the gene grammar determine how certain parts fit together. For example, if a G is encountered in the sequence it makes a decision on whether it is part of a GT donor splice signal or in the third position of the ATG start codon (among other possibilities). By attaching probabilities to the signals and content combinations, degrees of certainty can be calculated for different possible gene structures. The

range of statistical indicators used by the method help to overcome the unreliability of each when considered individually, and includes measures like exon size, donor and acceptor site consensus and splice quality.

As with HMMs and neural networks, the system must undergo training sessions before use to establish weightings and other parameters. Stormo and Haussler [1994] detail a general method for finding both optimal and a range of sub-optimal weightings given a set of inputs. Using a dynamic programming approach, their algorithm builds up local optimal weightings from a set of sequences and measures and combines them to obtain an optimal overall result.

As the body of sequenced DNA grows, the likelihood of the product of a new gene resembling or even matching that of an existing one increases; some gene finding methods attempt to take advantage of this or use it to validate their results. One method is as follows. Once a group of potential exons in a sequence has been established, a target protein is identified that is a *homolog* of the protein the gene codes for. That is, the coded proteins of the two genes are the same or they share similar exons. The algorithm then combines the putative exons in such a way as to resemble the homolog as closely as possible. Provided accurate homologs can be identified this method works very well, although newer and less understood genes tend to suffer from a lack of them [Haussler 1998; Batzoglou *et al.* 1998]

Despite the advances in the field and the intelligence of modern gene finding systems, users must still have a good understanding of the weighting and evaluation procedures in order not only to understand the results but tweak the system to get the most accurate results, especially for atypical data.

## 2.4 Language and DNA

### 2.4.1 The Linguistics of DNA

The representation of DNA as a string of letters naturally leads to its interpretation as a body of text written in some language. Metaphorically, one could consider this to be the language of life but the interpretation also has some more prosaic scientific merit.

The language-based approach to gene finding by Dong and Searls [1994] was very successful. In a manner similar to the way human language texts can progressively be broken down into paragraphs, sentences, words, nouns, verbs etc., so can DNA sequences: introns, exons, codons and their related signals can be considered the grammatical components of the DNA language. If the way in which these components relate to each other and fit together can be understood, the meaning of a DNA sequence can be extracted.

This interpretation is also considered by Batzoglou *et al.* [1998] who compiled a dictionary of DNA words—portions of DNA that were (in this case for simplicity) 11 nucleotides in length. Instead of associating a meaning with each word, each dictionary entry listed the members of the DNA database they used in which the word occurred. A similar dictionary was also constructed for amino acids. The primary purpose of their work was to provide a means for finding protein homologs, and their tests indicated that the dictionary method was more accurate than simple frequency analysis.

Batzoglou *et al.* [1998] make a slightly different language-related comparison: they liken introns and exons to two “scripts” in the same alphabet. The exon script is understood by biological mechanisms and eventually translated into amino acids and proteins (or some other gene product), while the intron script is nonsense—to the cell at least—and is ignored. Finding a method to distinguish between the two would be incredibly useful, albeit not on its own. To paraphrase the authors, one cannot expect to understand the meaning of words just by identifying them, especially when they may appear with different meanings in different paragraphs [Batzoglou *et al.* 1998, p. 657].

## 2.4.2 Compression and Linguistic Analysis

Linguists have been able to determine much about bodies of text by performing such simple analyses as comparing word frequencies [Batzoglou *et al.* 1998]. These methods have a great deal in common with the statistical and compositional approaches to gene finding, particularly the method by Benedetto *et al.* [2002]. They define a metric to measure the “distance” between two bodies of text, interpreted as the relative difference in their informational content or entropy. The metric is based on the relative compression ratios of the two bodies and application of it was very successful. Their work showed that if two bodies have significantly different information content or are written in a different language or by a different author, their relative compression ratios will be different. This difference affects the “distance” between them and is what they used to perform various linguistic analyses and comparisons.

The method employed by Benedetto *et al.* [2002] works as follows. Given two bodies of text,  $A$  and  $B$ , extract a short sequence  $a$  from  $A$  and  $b$  from  $B$ . By appending  $b$  to  $A$  and  $a$  to  $B$  and compressing the resulting sequences  $Ab$  and  $Ba$ , the short foreign sequences are effectively compressed using the optimal encoding of the full-length sequences. Let  $|x|$  be the length of the sequence  $x$  and  $L(x)$  the length of  $x$  after compression. Then  $\Delta_{Ab} = L(Ab) - L(A)$  is the length of  $b$  when encoded using the optimal encoding of  $A$ . The authors then define the relative entropy per character between  $A$  and  $B$  as  $S_{AB} = (\Delta_{Ab} - \Delta_{Ba})/|b|$ .

If the two bodies are “distant” in terms of information content, then their optimal encodings will differ, decreasing the effectiveness of the compression and increasing  $S_{AB}$ . If they are similar then the effect during compression, and the resulting entropy value, will be less. For instance, they were able to identify the language of an unknown text with great accuracy simply by calculating the distance between it and a number of known bodies and then choosing the pair with the least entropy per character. The technique was also used to identify the different families of languages across the world. It is important to note that the method did not interpret the text at all; it was treated simply as a stream of characters.

Considering the exons and introns of a DNA sequence to be written in two different languages, this research applied a similar method in an attempt to differentiate between them. The compression tools used by Benedetto *et al.* [2002], which were also used in this research, are discussed in the following section.

It is worth noting that Goodman [2002] criticised the work by Benedetto *et al.* [2002]. One of the criticisms was that the technique was not novel and better methods (in that they are slightly more accurate and quite a bit faster) of measuring the relative entropy of texts exist. However, the technique is still applicable and useful in terms of the research. This first step was to determine if the method had any value. If successful, further investigation into improving performance and accuracy was to be conducted.

## 2.5 Data Compression

Compression can be considered as a specific form of encoding in which the compressed encoding should require less space than the original encoding [Lelewer and Hirschberg 1987]. The compression should be accomplished in such a way as to preserve the informational content of the original data. For visual and audio data, parts of the data can be excluded without losing any information because the user will not notice the difference. Compression of textual data, on the other hand, usually requires that all information be preserved because any loss is significant. As DNA can be expressed as simple text sequences, only compression techniques aimed at text data are considered by this review.

The brevity of the compressed form of a sequence of letters is limited by its informational content, or entropy. Generally, one can consider a sequence that contains large amounts of redundancy, usually in the form of repetition, to have less informational content than a sequence

of the same length that contains little or no repetition. It is this fact that compression algorithms use to compress data into encodings that are as close as possible to being optimal.

Huffman encoding is a method that comes close to the optimal for an entire sequence [Wood 1993]. For each character in the sequence, the shortest unambiguous coding is calculated in such a way that the most frequent characters have the shortest codes. For small sections of the sequence, however, such an encoding may not be optimal. A more efficient approach is to consider the sequence in small windows and calculate the optimal encoding for each window, an approach termed *dynamic* or *adaptive* encoding (normal Huffman encoding is termed *static*) since the encodings change over the length of the sequence. This can be further improved by encoding entire subsequences with a single Huffman code, as opposed to single characters. Thus, optimal (or close to optimal) encodings can be determined for small subsequences, and these result in an optimal encoding for the sequence as a whole.

The compression algorithm must somehow determine the best encodings for repetitions in a sequence. The longer and more frequent the repetition, the more it will benefit from a short encoding. However, there is a trade off: the longer a sequence the less likely it is to be repeated. Compressors generally solve this problem by imposing some maximum limit on the length of the sequence, as well as on how far back or forward they will search for a repetition.

The LZ77 compression algorithm by Lempel and Ziv [1977] is an example of an algorithm that uses dynamic Huffman codes and a sliding window to find repetitions and compress data. It is the basis for the DEFLATE compression algorithm which is implemented by widely-used compression tools such as gzip and z-lib. These two algorithms are discussed below.

### 2.5.1 The LZ77 Compression Algorithm

Lempel and Ziv [1977] developed a general-purpose compression algorithm still in use today whose performance is comparable with that of more application-specific algorithms.

The algorithm works by considering the data to be compressed (the input stream) in a moving window of  $n$  characters. Part of the window, conceptually the right-hand side of it, covers data that has yet to be encoded: the lookahead buffer. The rest of the window covers data that has already been encoded (but is considered in its non-encoded form). The algorithm searches the window for the longest match with the beginning of the lookahead buffer and outputs a pointer to that match. Since it is possible that not even a one-character match may be found, the output cannot contain only pointers. To solve this problem, after the pointer (which may be null and point to nothing) the algorithm outputs the first character in the lookahead buffer after the match.

Suppose the input stream is a sequence of characters  $S = s_1s_2s_3\dots$  over some alphabet. If the window is at position  $p$  and all characters up to  $k$  have been encoded, then it covers the substring  $s_p s_{p+1} \dots s_k \dots s_{p+n-1}$ . The algorithm looks for the longest string in the window, starting at  $j \leq k$  and of length  $m$ , that matches the first  $m$  non-encoded characters. That is, it finds  $j$  such that  $m$  is maximal and  $s_j s_{j+1} \dots s_{j+m-1} = s_k s_{k+1} \dots s_{k+m-1}$ . Notice that the substring may extend past  $k$  and include some non-encoded characters (ie.  $j + m - 1$  might be greater than  $k$ ).

Once a match has been found, the pointer to it is the pair  $\langle j, m \rangle$ , where  $j$  is the position of the match in the window and  $m$  is the length of the match. This pointer is encoded and added to the output stream. The character just after the match,  $s_{j+m}$ , is then written to the output stream, the window is moved  $m + 1$  characters forward and encoding continues. If no match can be found, the pair  $\langle 0, 0 \rangle$  followed by  $s_j$  is encoded and the window is moved one character forward.

Decoding, not of importance to the research, is performed simply by reversing the encoding process.

The worst case performance of this algorithm occurs when the data contains either no repetitions at all or repetitions of single characters [Lempel and Ziv 1977]. In this case, the coded

form of the data is actually larger than the original because the (comparatively) lengthy pairs  $\langle 0, 0 \rangle$  and  $\langle p, 1 \rangle$  are used to encode a single character. This limitation is removed by the slightly modified implementation of LZ77 used by the DEFLATE compression algorithm discussed in the next section.

## 2.5.2 The DEFLATE Compression Algorithm

The DEFLATE compression algorithm, documented by Deutsch [1996], uses a combination of a slightly modified LZ77 algorithm and Huffman encoding to perform compression that is comparable with the best general-purpose compression methods. The algorithm is implemented by gzip [Gailly and Adler 1993] which was the tool used by Benedetto *et al.* [2002] and in this research. The description of DEFLATE below is summarised from the work of Gailly and Adler [1993] and Deutsch [1996].

DEFLATE makes some minor adjustments to LZ77 that improve compression, but the two are otherwise the same. The maximum length of a match is 258 bytes<sup>1</sup> and the search for a match can go only as far back as 32,768 bytes. This makes the window size effectively 33,026 bytes. Additionally, to overcome the limitations of LZ77, null pointers are not used if no match can be found. Instead, the current character in the input stream is written and encoding continues.

While not part of the algorithm itself, the manner in which the implementation of DEFLATE searches for matches is important. A chained hash table is used to store all previously seen strings of three or more characters and is referred to as the dictionary. Such a system consists of a hashing function  $f(x)$  that operates on a string, and an array indexed by the output of the hashing function that forms the table. The function  $f(x)$  is not unique for its inputs and when multiple strings hash to the same value, they are all stored under the same array entry and chained together (the first pointing to the second, the second to the third, etc.) [Baase and Van Gelder 2000].

To search for the longest match of a string  $S = s_1s_2s_3 \dots s_{258}$  during encoding, DEFLATE calculates the hash of the first three characters,  $f(s_1s_2s_3)$ . It uses the result to look up the corresponding chain of strings in the table and finds the longest match. When a match is found a pointer to it is written to the output stream, as per LZ77, and the window is moved the corresponding number of characters forward. If no match is found (which might happen even for the first three letters alone since  $f$  is not unique), only  $s_1$  is written and the window is moved forward one character.

For performance reasons, entries are never removed from the hash table; instead a match that is too old (ie. the string occurred more than 32 768 characters ago) is simply ignored. Before the table is checked for the longest match, the string  $s_1s_2s_3$  is always hashed and added to the table in such a way that recent strings always precede older ones in the chain, ensuring that the most recent matches are always used.

The research used DEFLATE's ability to identify and exploit repetitions in its input stream in an attempt to distinguish between the different repetitions and patterns of introns and exons in DNA sequences. The research helped to shed light on whether or not the regions have sufficiently different patterns to distinguish between them.

## 2.6 Conclusion

Sequences of DNA can be broken into basic functional parts—genes—which can be broken further into signals, introns, exons etc. Only certain parts of a gene are used and the “junk” regions must be removed before the product the gene codes for, usually a protein, can be expressed. Identifying genes and separating these coding and non-coding regions is a complex

---

<sup>1</sup>A byte can be considered to be a single character of text, or in the case of the proposed research, a letter representing a nucleotide.

task which must be performed before biologists can hope to understand how the gene and its product are used.

Computational techniques for gene finding generally take two approaches. Signal-based approaches search for the signals embedded in the DNA, such as those that separate introns and exons. Content-based approaches use the structural and statistical differences between the two regions to identify them. On their own, however, they cannot capture the interaction between the signals and content and so hybrid methods, which are far more effective, have been developed. Despite their success, these methods are still not perfect.

Hybrid methods that use grammars and other language-related techniques to combine content and signal sensors have been very successful. Investigating further applications of linguistic analysis methods may prove fruitful. One such method, investigated by this research, involves using compression to differentiate between texts based on their different optimal encodings. Compression algorithms such as DEFLATE and LZ77 identify repeated sequences in texts and replace them with pointers to the previous occurrence; thus the optimal encodings for sequences with varying patterns differ. The research attempted to use the different optimal encodings of DNA sequences to differentiate between them. The method and results are described in the next few chapters.

# Chapter 3

## Research Method

### 3.1 Introduction

The gene finding methods reviewed in Chapter 2 are successful only to a certain extent. Many of them cannot be generalised effectively and often there are no similar known sequences to compare new sequences against to check results. The greater the number of sensor and content measures contributing to a decision, the greater the likelihood of the decision being accurate and reliable.

Previous work has shown that linguistic approaches to gene finding can be successful. The research investigated the suitability of applying similar methods to those used by Benedetto *et al.* [2002] to DNA sequences. The primary focus was on the identification of the two different “languages” of exons and introns. Additionally, the method was used to differentiate between genes from different families and genomes from different species. Overall, the research explored the feasibility of applying a compression-based entropy measure to genomic data.

This chapter presents and discusses the research question and research method. The formal research question is given in Section 3.2, its motivation is discussed in Section 3.3 and the strategy for investigating it is outlined in Section 3.4. Section 3.5 discusses the entropy per character measure and its application and Section 3.6 describes the experiments and the data sets used in the research.

### 3.2 Research Question

The aim of the research is to investigate whether or not a compression-based entropy measure can be successfully applied to DNA sequences in an attempt to capture the differences between parts of a gene, gene families and genomes in general. The measure is based on differences in the optimal encodings of the data being considered.

The formal research question is as follows:

*Can the compression-based entropy-per-character measure, when applied to DNA sequences, identify the differences and similarities between introns and exons, genes from different gene families, and entire genomes, with sufficient accuracy to discriminate between them?*

The primary focus of the research is the application of the entropy measure to the gene finding problem: identifying coding and non-coding regions—the first of the three parts of the research question. Each part targets a slightly different application of the entropy measure in the bioinformatics context, each one viewing genes from a more general view, abstracting out more detail than the last. The first deals with the main components of a gene—the introns and exons. The second considers the gene as a whole, comparing genes based on their characteristics and use. The third part investigates the genome itself, genes, introns, exons and all.

In each of these cases two types of data have been identified, either different parts of a gene, different groups of genes, or different genomes. The question asks whether or not the entropy measure can be used to identify the two types based on their optimal encodings and resulting entropy values, and with what accuracy.

### 3.3 Motivation

The success of previous language-based methods to the gene finding problem, such as those by Dong and Searls [1994] and Batzoglou *et al.* [1998], indicated that such an approach may have merit. This, coupled with the successful application of compression to linguistic analysis by Benedetto *et al.* [2002], suggested that it may also be useful in a gene finding context, as well as in the broader context of genomes in general.

Benedetto *et al.* [2002] used a reasonably simplistic way of measuring the performance of their compression algorithm as it moved from one region into the next. Since they already had the two as separate sequences, calculating the entropy per character was relatively simple. In this research, however, the boundaries between the regions were not necessarily known and instead had to be determined. In all four experiments, the basis for differentiating between each of the two regions lay in their different optimal encodings. In order for the entropy measure to show a change, part of one region had to be encoded by the compressor using the optimal encoding of the other region. The entropy measure would then produce a value based on how different the two encodings were. The further apart the two bodies of data were in terms of content, the greater the entropy value.

The choice of the DEFLATE algorithm, and hence gzip as a compressor, was partially due to Benedetto *et al.* [2002], whose work suggests it may be useful, and partially due to the characteristics of the algorithm. In order for different regions to have different optimal encodings, the compressor had to use a dynamic encoding scheme—one whose encoding changes as the data being encoded does. DEFLATE achieves this with its window-based dictionary.

The compressor was used “as is” without any tailoring towards the problem. The work could have potentially added another model-independent content measure to the repertoire of existing measures in an attempt to further improve the accuracy and generalisability of computational gene finding methods. It could also have resulted in measures for the other facets of bioinformatics included in the research question: gene families and genome similarities.

### 3.4 Research Method Overview

The nature of the research question required a number of separate experiments, each of which applied the entropy measure to a slightly different problem. One experiment targeted introns and exons, another genes from different families, and a third compared random parts of entire genomes.

The general approach of the research method was to compress the two bodies of data in such a way that part of one body was compressed using the optimal encoding of the other. If the two bodies are alike, from the same family, or have similar purposes, then their optimal encodings should have certain similarities. The entropy measure is an attempt to identify these similarities (or lack thereof) and produce an output indicating how alike or how “close” the two bodies are. The measure uses the lengths of the compressed sequences to make the judgement. The closer the compressed sequences are in length, the more similar they are deemed to be, and vice versa. The first step in the research was developing the measure itself based on the work by Benedetto *et al.* [2002]. It is described in detail in the next section. Once the measure was developed, it was applied to the data in a number of different ways.

For the coding/non-coding region experiment, since the regions are intertwined, an approach was taken that was slightly different from that for the other experiments. Each position in the

input sequence was potentially a splice junction and so for each position an entropy value was calculated. For the other experiments, entropy values were calculated only for certain positions in the sequences and the results averaged. Once calculated, the entropy values were analysed to determine if they could be used to differentiate between the two bodies of DNA data involved in each experiment.

In addition to the three experiments described above, an experiment was performed to determine if the entropy values of introns and exons exhibit any major differences: instead of comparing the two regions together, each was considered individually. The resulting entropy values were used to gauge differences within each region. Exons should share patterns and similarities to some degree, while introns, naturally being more random, should have less commonality.

The research question considered genomic data from a number of standpoints. Each of the experiments targeted one of these standpoints and the combined effect was the application of the entropy measure to DNA data in a number of different scenarios. The results, when considered as a whole, allowed an answer to the research question to be formulated.

## 3.5 Compression and the Entropy Measure

### 3.5.1 Entropy Value Calculation and Use

The relative entropy per character measure, based on the definition by Benedetto *et al.* [2002], was used to measure the distance between each pair of bodies being considered. The measurement process works as follows. At each position in the input sequence, a small subsequence—call it  $s$ —is extracted. This  $s$  is then compressed using the optimal encodings for two larger subsequences; one preceding the small subsequence ( $A$ ) and one following it ( $B$ ). If  $s$  belongs to either  $A$  or  $B$  then they should have common characteristics. Provided  $A$  and  $B$  have sufficiently different optimal encodings, compression using the encoding of the “owner” sequence should be better than compression using the encoding of the other sequence. This should lead to an entropy measure for  $s$  that is greater than if  $s$  did not specifically belong to either  $A$  or  $B$ . The measure is defined formally in the next section.

This method can be applied in two ways. If the boundary between the two sequences being compared is not known, it can be applied at a number of positions in the sequence. The areas with high entropy values are then potentially on or near the boundaries between the two sequences. Alternatively, if the two sequences are already separate, then  $A$  and  $B$  can be concatenated and  $s$  becomes a subsequence at the start of  $B$ . In this case only one calculation is performed and a judgement on the relatedness of the two sequences can be made.

During experimentation, the lengths of  $A$ ,  $B$  and  $s$  were chosen primarily with two limitations in mind. Firstly, if  $A$  and  $B$  were too short it would not be possible to establish their optimal encodings. If  $s$  was too short, then the optimal encoding would have little or no impact. Secondly, if any of the sequences were too long, they may have had encodings that were too general. Some of these problems did arise during the research and their effects are discussed further in Section 5.6.

### 3.5.2 The Relative Entropy per Character Measure

To calculate entropy values, optimal encodings for subsequences on either side of a position  $x$  in the input sequence  $S = s_1s_2s_3\dots$  are generated. A short portion of the subsequence to the right of  $x$  is then compressed using both encodings, resulting in an entropy per character value.

The relative entropy per character measure is defined as follows. Let  $|s|$  denote the length of a string  $s$ ,  $L(s)$  the length of the compressed version of  $s$ , and  $S(x, d) = s_x\dots s_{x+d-1}$  the subsequence of  $S$  that starts at  $x$  and has length  $d$ . Let  $A$  and  $B$  be two sequences of the same length and  $s$  a significantly shorter sequence. Then

$$\Delta_{As} = L(As) - L(A)$$

is the length of  $s$  when encoded using the optimal encoding of sequence  $A$ . The relative entropy per character between  $A$  and  $B$  is given by

$$E_{AB} = \frac{\Delta_{As} - \Delta_{Bs}}{|s|}$$

The three sequences are constructed as follows. For each position  $x$  in the input sequence  $S$ , let  $A = S(x - w, w)$ ,  $s = S(x, d)$  and  $B = S(x + d, w)$  with  $d \ll w$ . So  $AsB$  is a contiguous subsequence of  $S$  of length  $w + d + w$ , as depicted in Figure 3.1. Compressing the sequence  $As$  results in  $s$  being compressed using the optimal encoding of  $A$ . Similar results apply to  $Bs$  and the final value of  $E_{AB}$  can be calculated.

It is important that  $s$  is appended to both of the other two sequences. If, instead,  $As$  and  $sB$  were compressed, the results would be incorrect. In order for  $s$  to be compressed using the optimal encoding of  $B$ , that encoding must first be generated by the compressor. That is,  $B$  must be compressed first, followed by  $s$ . Thus, compressing  $Bs$  instead of  $sB$  gives the value required by the measure.

### 3.5.3 Alternative Compression Measures

A slightly different, more low-level approach was initially considered. The encodings of the two regions should result in different dictionary usage characteristics. For example, certain entries (possibly corresponding to single codons or pairs of them) would be used more frequently in some regions than in others. For instance, the range of dictionary entries used in introns is likely to be greater than in exons, due to the former's more random codon distribution. Due to these differences, monitoring dictionary usage during compression may allow for region identification. However, this method would require modifications to an existing gzip implementation and is somewhat more complex than the method outlined above. A number of different variables would have to be considered, such as the window across which usage frequencies are measured. Another issue is the fact that both regions may show large ranges in usage within themselves. Due to these problems, as well as the fact that it is closer to the method used by Benedetto *et al.* [2002], it was decided that the more naive, but less complex entropy-based approach would be used instead.

It is unlikely that this more complex approach would have produced results that were any more useful than those obtained. As discussed in Chapter 5, the most likely reasons for failure are not due to the entropy measure itself but more to the characteristics of DNA sequences and compression algorithms in general; changing the manner in which the compression performance is monitored is unlikely to make a difference.

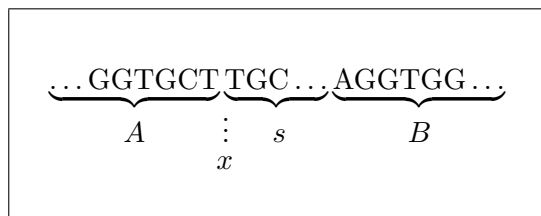


Figure 3.1: The sequences used by the entropy measure. The current position is  $x$ ,  $|A| = |B| = w$  and  $|s| = d$ . The  $A$  sequence starts at position  $x - w$  and  $B$  at  $x + d$ . The value of  $d$  is chosen so as to be less than  $w$ .

Sequence	Length	Exons (Mean Length)	Introns (Mean Length)	Length Pairs Used
ACU08131	5392	6 (184)	5 (525.4)	(30, 198), (30, 240), (60, 198), (60, 240), (90, 198), (90, 240), (90, 300), (90, 360), (120, 198), (120, 240), (120, 300), (120, 360), (150, 198), (150, 240), (150, 300), (150, 360)
AGU04852	10984	5 (207.8)	4 (856.25)	(60, 240), (120, 300), (150, 360)
HUMATPGG	13683	22 (142.45)	21 (472)	(120, 300), (150, 360)

Table 3.1: Sequences used in the Cross-Region and Independent Region experiments. Each length pair  $(d, w)$  corresponds to the  $d$  and  $w$  variables used in the entropy measure as described in Section 3.5.2.

## 3.6 Experiments

### 3.6.1 Cross-Region Comparison

The cross-region comparison experiment was the primary focus of the research. It used the entropy measure in an attempt to distinguish between coding and non-coding regions in a gene.

Initially a single gene sequence, ACU08131, was used with a range of subsequence ( $A$ ,  $B$  and  $s$ ) lengths. Once those results had been analysed, two other sequences were used with selected sequence lengths. All three sequences were taken from the data used by Burslet and Guigó [1996] in a comparison of gene finding programs<sup>1</sup>. The data set contains 570 sequences (each comprising a single gene) and contains only vertebrate protein coding sequences, with all anomalous sequences (such as those not beginning with a start codon) removed. The removal of anomalous sequences was important; if the method proved to be successful it was to be applied to more atypical data. The three sequences are summarised in Table 3.1 where each length pair  $(d, w)$  corresponds to the  $d$  and  $w$  variables used in the relative entropy per character measure as described in Section 3.5.2.

The lower bound on the  $w$  values is slightly lower than the minimum value of 200bp required by Testcode [Fickett 1982] which was used as a guide. The upper bound was dictated by the lengths of sequences in the data set as well as by the lengths of their introns and exons. Benedetto *et al.* [2002] state their results to be successful with their equivalent of the  $s$  sequence as short as 1KB (approximately 1,000 nucleotides), although they do not mention any lack of success below that.

Before each sequence was used, the 5' end (ie. everything before the start codon) was removed. Then, for each length pair  $(d, w)$ , the entropy values for the sequence were calculated starting at position  $w + 1$  (since a sequence of  $w$  basepairs to the left of the position was needed). The position was incremented in steps of three to reduce the number of calculations required, producing a third of the total possible values without affecting the characteristics of the results. The final position was  $w + d$  basepairs from the 3' end of each sequence, since consecutive sequences of length  $w$  and  $d$  had to be extracted to the right of the final position. On both sides, therefore, there was a small margin that contained no entropy values.

For interpretation the values were plotted alongside the known splice junctions. The layout of the entropy values across the gene, especially near splice junctions, gave an insight into whether or not they could be used to identify either the two regions or the splice junctions themselves.

The aim of this experiment was to determine if the entropy measure could be used to identify introns and exons based on their splice junctions and the areas around them. By partitioning

<sup>1</sup>The data is available online at [www.imim.es/GeneIdentification/Evaluation/Index.html](http://www.imim.es/GeneIdentification/Evaluation/Index.html)

the sequence based on the positions that incurred maximum entropy values, the coding and non-coding regions might have been identified. Once an attempt had been made to identify the regions, they were to be compared with the original annotations to determine the level of accuracy, as well as with the results of other gene finding programs to gauge relative levels of success. However, it was not possible to distinguish between coding and non-coding regions based on the entropy values. The comparison with the actual regions, as well as with the results from other methods, was therefore not performed.

### 3.6.2 Independent Region Comparison

The second experiment was conducted to determine if coding and non-coding regions had different entropy characteristics. The cross-region comparison performed in the first experiment attempted to identify splice junctions. Instead, this experiment calculated entropy values for each region independently in an attempt to identify characteristics that could be used to distinguish between the two. The focus was therefore on the nature of the two regions and not the boundaries between them.

For each of the three sequences, the introns and exons were extracted and joined to form two new sequences, one composed entirely of introns and one composed of exons. Both of the sequences were then subjected to the same entropy calculations as in the cross region comparison, using selected sequence lengths. Finally, the mean and standard deviation for each set of entropy values was calculated.

The coding regions of a single gene have similar characteristics, while the non-coding regions are more random and less distinctive. Consequently, the entropy values for exon sequences should be relatively uniform, display a lower mean and deviate less from that mean. The values for the intron sequences, however, should have a wider range: a higher mean with a greater deviation. If significant enough, such a comparison may be used to distinguish between the two regions, albeit probably quite roughly.

This experiment allowed a slightly different part of the research question to be answered, namely that of whether the entropy measure could be used to differentiate between regions based on their entropy values (as opposed to the splice junction entropy values). It was also used as a benchmark test for the cross-region comparison. If the measure failed to produce results consistent with those expected for this experiment, it seemed unlikely that the measure would be sufficiently powerful to differentiate between the two regions, now matter how it was applied.

### 3.6.3 Gene Family Comparison

Many organisms have a number of genes in common that perform similar tasks. The genes themselves may be identical or slightly different, but they are said to belong to the same gene family [Bergheim 2002a]. For example, a myotubularin gene in a human and in a mouse share many similarities but are not necessarily identical. In this experiment the entropy measure was applied to determine if it could be used to link genes from the same families. It considered the genes in their original forms; the 5' and 3' ends were left intact and introns were not removed.

The data, taken from the NCBI website [NCBI], consisted of human (*Homo sapiens*) and mouse (*Mus musculus*) myotubularin genes; human, mouse and rat (*Rattus norvegicus*) heat shock genes; and human and rat zinc finger genes. Since the genes were of different lengths, it was not possible to use each gene in its entirety. Instead, parts of each gene were compared against one another using a range of sequence lengths. The human, mouse and rat genomes were chosen because all three have large, readily available bodies of data and are active areas of research.

Since in each case the two bodies of data were already separate, it was only necessary to calculate a single entropy value. For each pair of lengths ( $d$ ,  $w$ ), subsequences of length  $w$

were taken from random positions in both genes. The  $d$  nucleotides preceding this sequence in the second gene were taken as the  $s$  sequence and the entropy measure was applied. This was performed three times for each sequence with each length pair, and the results averaged to give the final entropy values. These values were then compared. The length pairs used in this experiment were reasonably short as the genes ranged in length from only 1,700bp to approximately 4,800bp.

To determine a basis for the comparison, the same procedure was applied while using genes from the same family for both sequences. The resulting values give an indication of what the entropy values should look like for genes from the same family, and hence for ones from different families. Significant differences between the basis values and the other values would suggest the measure could be useful in determining (very roughly) if two genes were related or not.

### 3.6.4 Genome Comparison

The genome comparison experiment had an objective similar to the gene family comparison experiment. Instead of comparing gene families, though, it compared sections of entire genomes. Different genomes are known to have different characteristics (such as common CG islands [Hausler 1998]) and it was hoped that the entropy measure would be able to use these characteristics to determine the similarity of different genomes.

The three genomes used, again taken from the NCBI website [NCBI], were the human X chromosome containing 476,889 basepairs, the mouse X chromosome containing 246,848 basepairs and the wheat (*Triticum aestivum*) chloroplast genome containing 134,545 basepairs. The species were chosen because all had lengthy bodies of data that were easily available and are currently being researched. Parts of each genome, taken as-is without modification or extraction of non-coding regions, were compared against one another using a range of sequence lengths in a method similar to the one used for the gene family comparison. Since the genomes differed in length, they could not be used in their entirety. Instead, sequences were taken from three random locations in each genome and the resulting entropy values were averaged for each length pair.

A wider range of length pairs were used than in the previous experiment as the genomes ranged in length from 10,000bp down to 500bp. For the longer mouse and human genomes, larger lengths of 100,000 were also used. Again, basis values were established by comparing different parts of the same genome with each other. By comparing these basis values with the other values it was possible to make a decision regarding the effectiveness of the measure.

## 3.7 Conclusion

Computational gene finding methods are by no means perfect and the problem of finding further measures and methods to make them more accurate is still open. This chapter outlined the research method and how it contributed to the effort to find new measure. The general aim of the research was to determine whether or not a compression-based entropy measure would be useful and to what extent.

Using an entropy per character measure based that used by Benedetto *et al.* [2002], the research attempted to break DNA sequences into coding and non-coding regions based on the level of entropy identified at each point in the sequence. The method used the text compressor gzip to compress a short subsequence of the DNA strand twice, once using the optimal encoding of the sequence preceding it and once using the optimal encoding of the sequence following it. The lengths of the compressed texts were used to calculate a measure of entropy at each point in the sequence. The measure was also applied to selected parts of entire genomes, as well as genes from separate families, to more widely investigate its applicability. The next two chapters present and discuss the results of the research.

# Chapter 4

## Results

### 4.1 Introduction

This chapter presents the results of each of the four experiments. While they are described briefly, a full discussion of their interpretation is left until Chapter 5. The results of the cross region comparison are given for selected sequence lengths across all three sequences used. The lack of any patterns identifying splice junctions is highlighted and some other, unexpected results are identified. The results of the independent region comparison show that introns often display a higher range of entropy values, although the difference is not great. Finally, the gene family comparison results show no difference between genes from different families, and the genome comparison has a similar outcome.

Results of the cross-region comparison are presented in Section 4.2 and the independent region comparison in Section 4.3. Section 4.4 presents the results for the gene family comparison and finally Section 4.5 presents those for the genome comparison.

### 4.2 Cross-Region Comparison

#### 4.2.1 ACU08131

The entropy values are easiest to interpret when plotted alongside the sequence's splice junctions. Figures 4.1, 4.2 and 4.3 show the entropy values and splice junctions for the ACU08131 sequence taken with increasing sequence lengths.

Each point is the entropy value for that position in the sequence. The vertical dashed lines indicate splice junctions, except for the rightmost one which indicates the end of the sequence. The leftmost region, starting at position one, is an exon which is then followed by an intron, then an exon etc. Note that the empty margins on either side of the sequence are the areas for which no entropy values were calculated. The y-axis has been moved up slightly for clarity. The large, rightmost region in all the figures is not an intron but the 3' end of the gene. It is interesting to note that almost all of the 3' ends contain regions with high entropy values.

The striated nature of the entropy values is due to the measure: the compressed sequence lengths generally differed by only up to ten characters (ie.  $\Delta_{As} - \Delta_{Bs}$  was an integer value ranging from 0 to a maximum in the region of 10). As a result, there are only a small number of entropy values.

As can be seen in the figures, as the sequence lengths increase the entropy values do not gain any new characteristics. Instead, the layout of the entropy values (their spikes and troughs) remain relatively constant—each figure has similar characteristics. For example, the peak between positions 400 and 1,000 is present in all three figures.

### 4.2.2 AGU04852

Figures 4.4, 4.5 and 4.6 show the entropy values for the AGU04852 sequence with similar sequence lengths to the previous three figures. Note that the first exon in the sequence begins at the first nucleotide and is only three basepairs long. Thus the leftmost region in the figure is an intron.

Similar comments to the above apply to these three figures. The regions of high values are relatively similar in all three and the values lack distinguishing characteristics near the splice junctions. The exons in the AGU04852 sequence are much smaller than in the ACU08131 sequence (barring the final one) and as a result the sequences used by the entropy measure would have occasionally overlapped multiple splice junctions. The impact of this is minimal, judging by the similarity of the figures for the broad exon compared with the thinner two. If overlapping multiple regions had affected the results, the values for the thin exons would be different to those of the broad one.

### 4.2.3 HUMATPGG

Figures 4.7 and 4.8 show the entropy values for the HUMATPGG sequence, a much longer sequence with more regions. The first splice junction is very near the start of the sequence, so the first discernible region in the figures is an intron, as are all the larger regions. Again, the characteristics of the values do not change significantly with the change in sequence lengths, and there is no definitive pattern around the splice junctions.

For all of the sequences, there is no significant and reliable change in the values at the boundaries of the regions. One might expect the entropy values to peak close to splice junctions as the *B* sequence moves into the new region and the *s* sequence belongs more to *A* than *B*. Similarly, as the position moves past a splice junction both the *s* and *B* sequences are in the same region while *A* is in a different one. Again, high entropy values are expected. Such features are not present in the figures, however. There is also no other characteristic of the values that correlates with the splice junctions.

One of the more remarkable features of the figures is the range in entropy values in introns. A number of the non-coding regions show either a large number of positions with high values, or a lack of them. However, this difference may be due to the average length of introns, which is greater than that of exons. The independent region comparison sheds some light on this issue and it is discussed further in Section 5.2.

## 4.3 Independent Region Comparison

The median and standard deviation of the entropy values, calculated for the coding and non-coding regions independently, are shown in Table 4.1. Of the fourteen length pairs, eleven show a higher mean for the introns than for the exons (marked with an asterisk), although the difference is quite small. Additionally, eleven (marked with a dagger—the same eleven barring two) also show a higher standard deviation for the introns.

While evident, these differences are not as marked as the figures in the previous section would suggest. The entropy values were calculated by considering each of the two regions independently. While the subsequences used in the cross region comparison occasionally spanned multiple splice junctions (and thus regions), those used in this experiment did not. It measured the similarity of the two regions to themselves instead of to each other.

The fact that exon entropy values have a lower mean and deviate less from that mean indicate that the exons in each sequence are relatively similar. The introns, on the other hand, tend to differ more as the wider range in entropy values show. However, these differences are not very

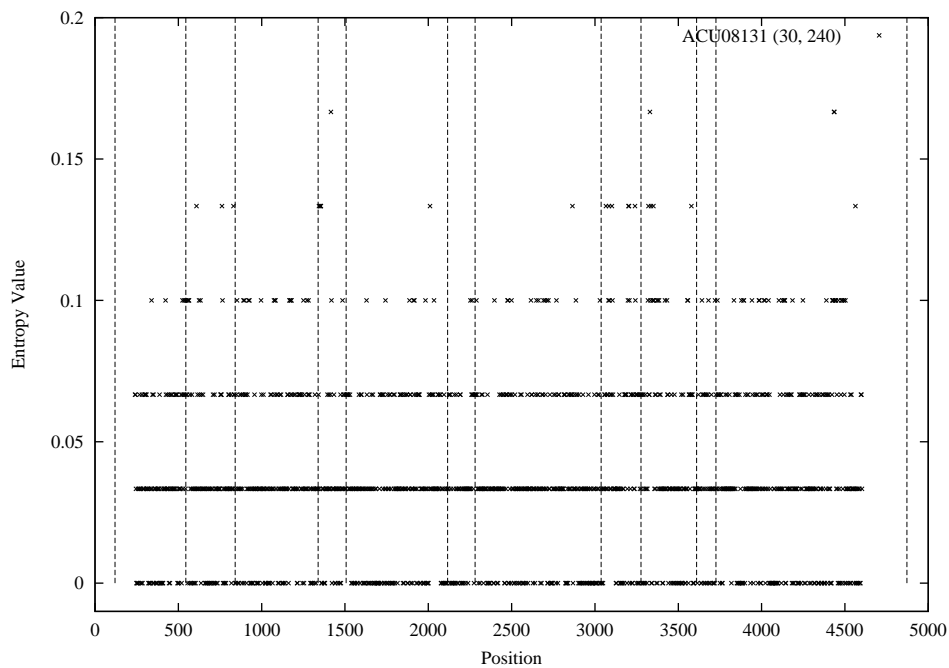


Figure 4.1: ACU08131 sequence with lengths (30, 240). Vertical dashed lines indicate splice junctions, except for the rightmost one which indicates the end of the sequence. The leftmost region, starting at position one, is an exon which is then followed by an intron, then an exon etc. The y-axis has been moved up slightly for clarity.

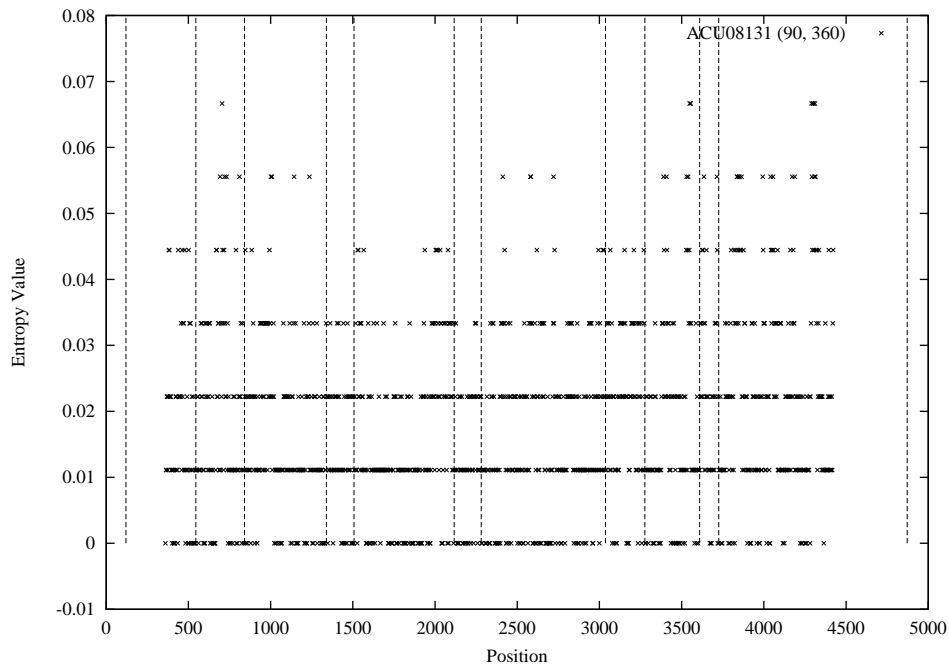


Figure 4.2: ACU08131 sequence with lengths (90, 360).

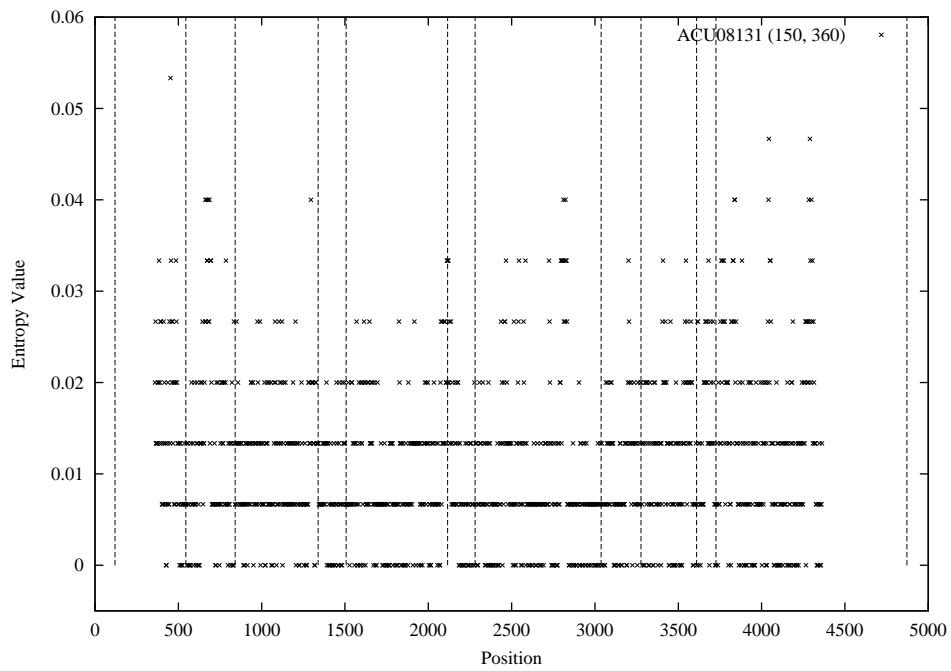


Figure 4.3: ACU08131 sequence with lengths (150, 360).

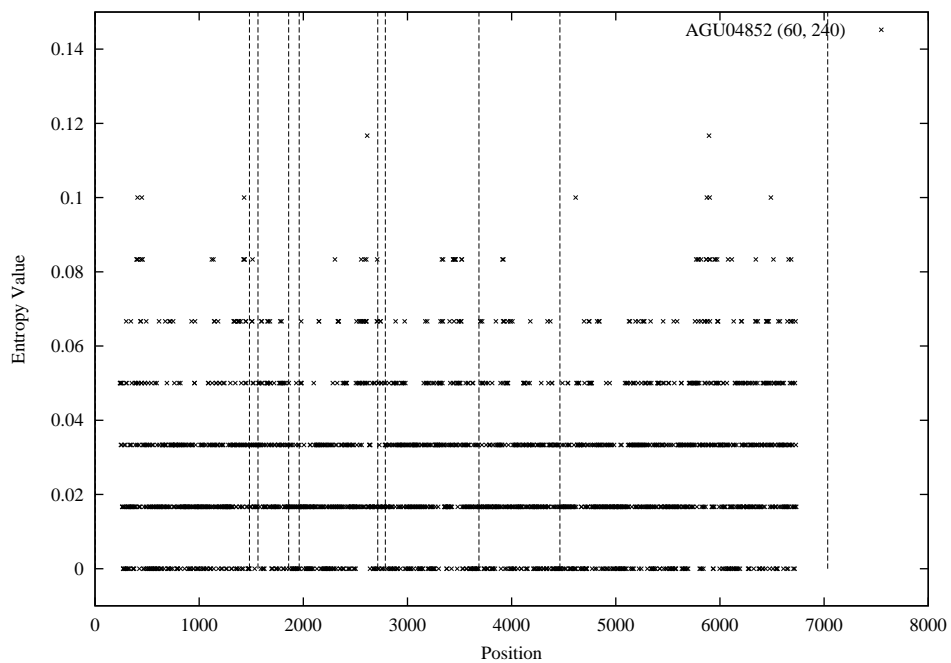


Figure 4.4: AGU004852 sequence with lengths (60, 240). The first exon in the sequence begins at the first nucleotide and is only three basepairs long, making the large leftmost region an intron.

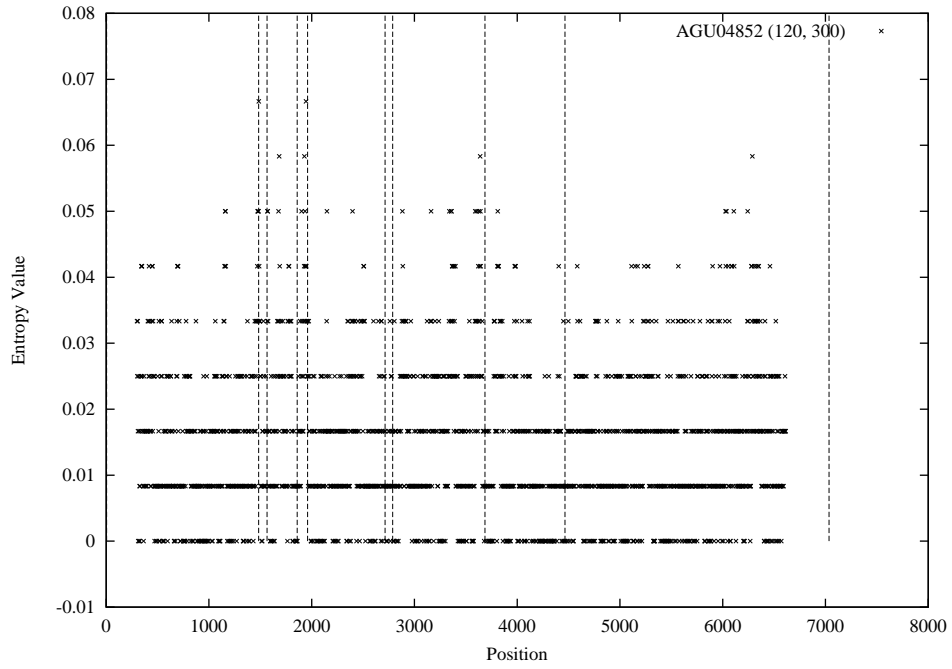


Figure 4.5: AGU004852 sequence with lengths (120, 300).

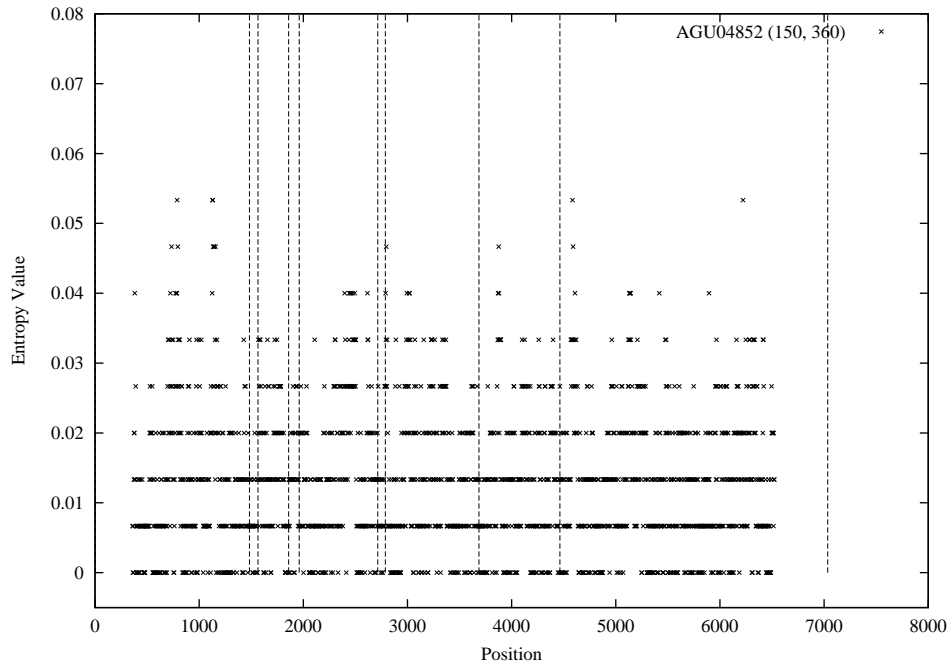


Figure 4.6: AGU004852 sequence with lengths (150, 360).

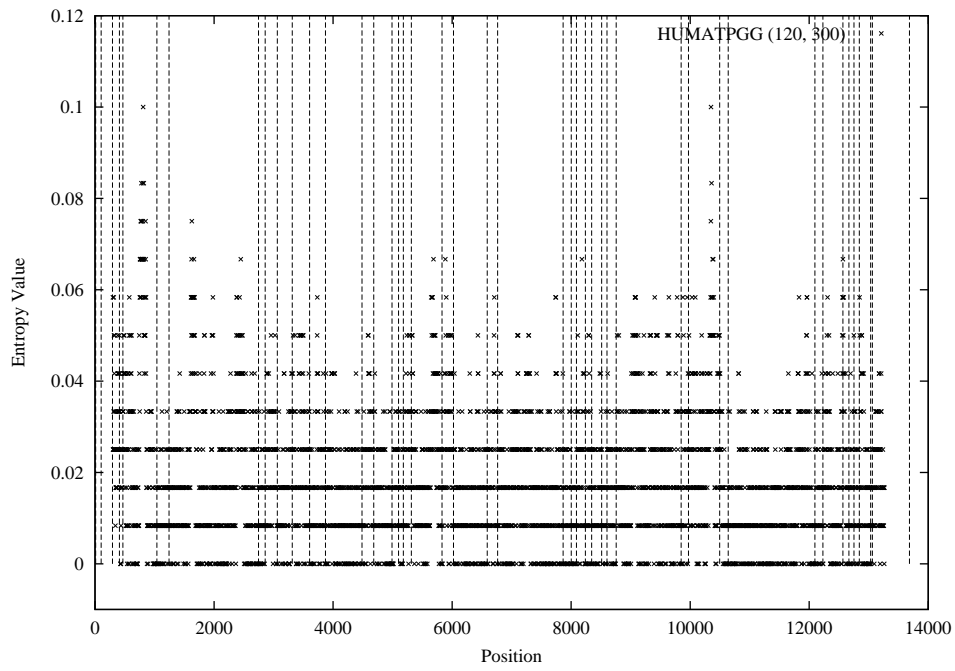


Figure 4.7: HUMATPGG sequence with lengths (120, 300). The first splice junction is very near the start of the sequence, so the first discernible region is an intron, as are all the larger regions.

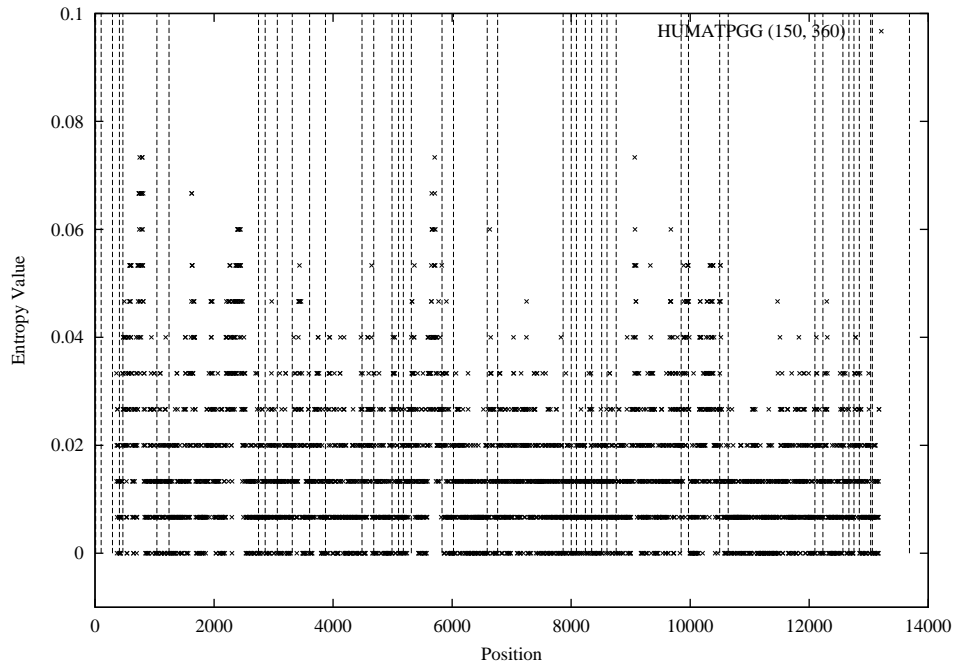


Figure 4.8: HUMATPGG sequence with lengths (150, 360).

Lengths	Sequence	Exons		Introns	
		Mean	Std. Deviation	Mean	Std. Deviation
(30, 240)*†	ACU08131	0.048148	0.035207	0.041000	0.034323
(60, 240)*†	ACU08131	0.024911	0.021642	0.024126	0.020112
(60, 240)	AGU04852	0.023952	0.018827	0.027763	0.021641
(90, 240)	ACU08131	0.016479	0.013543	0.018411	0.014304
(90, 300)	ACU08131	0.014895	0.011151	0.017690	0.014504
(90, 360)	ACU08131	0.016553	0.010677	0.017832	0.015697
(120, 300)	ACU08131	0.013542	0.009547	0.014856	0.012198
(120, 300)	HUMATPGG	0.013654	0.011098	0.017646	0.014343
(120, 300)*	AGU04852	0.017446	0.011194	0.014838	0.011506
(150, 300)	ACU08131	0.009266	0.008281	0.012665	0.009900
(150, 360)	ACU08131	0.010256	0.008108	0.011589	0.009469
(150, 360)	AGU04852	0.009825	0.007908	0.015560	0.012932
(150, 360)†	HUMATPGG	0.012221	0.009792	0.012584	0.009351
(150, 420)	ACU08131	0.008947	0.008416	0.012657	0.010511

Table 4.1: Median and standard deviation of the exon and intron entropy values. Those marked with a \* have an exon mean greater than their intron mean. Those with a † have an exon standard deviation greater than their intron standard deviation.

large—using them to identify the two regions would most likely be inaccurate and unsuccessful. These results and their implications are discussed further in Section 5.3.

## 4.4 Gene Family Comparison

For this comparison it was only possible to compare certain pairs of families and with certain sequence lengths, due to the lengths of the sequences involved. Figure 4.9 shows the entropy values for three length pairs, calculated for genes from the same family and genes from different families. A total of 18 same-family and 22 different-family calculations were made.

Entropy values for genes from the same family (the basis values), which should be relatively low considering the genes are similar, are neither exclusively high nor exclusively low. The highest values for two of the length pairs are actually for genes in the same family. The values for genes from different families also have a large range and include, unexpectedly, very low values. Overall, the values lack any features that would indicate that some genes belonged to the same families, while others were from different families. The results are discussed in more detail in Section 5.4.

## 4.5 Genome Comparison

The results of the genome comparison experiment are presented in Figure 4.10. Of the lengths in the figure, the pairs up to (5,000, 10,000) were used for all of the genomes. The three pairs involving 100,000 basepairs were used only for the mouse and human genomes since they were the only two that were sufficiently long. In total, 21 same-genome and 22 different-genome calculations were made.

A wide range of entropy values resulted for both the basis values (calculated using the same genome for both sequences) and the values for different genomes. As in the previous experiment, basis values were expected to be low, while the other values were expected to be higher. Again, there is no distinguishing characteristic that sets one set of values apart from the other. Instead, both sets have a wide range of values and show no pattern. The results include values for the

same-genome set that are high, and values for the different-genome set that are low. This is contrary to the low and high values, respectively, that were expected.

A more detailed discussion of the results is given in Section 5.5.

## 4.6 Conclusion

Each of the four experiments targeted slightly different aspects of the entropy measure and each produced similar results. The cross-region comparison did not reveal any pattern that could be used to identify splice junctions, although the layout of the entropy values has some unexpected characteristics which are discussed further in the next chapter. The independent region comparison shows that while a small difference in the mean and standard deviation of the two regions does exist, it is not definitive and it is unlikely to be significant. The gene family and genome comparisons fail to show any discrimination between their two sets of data; the range of values is unexpected and yields little useful information. The results of all four experiments, and their implications, are discussed in detail in the next chapter.

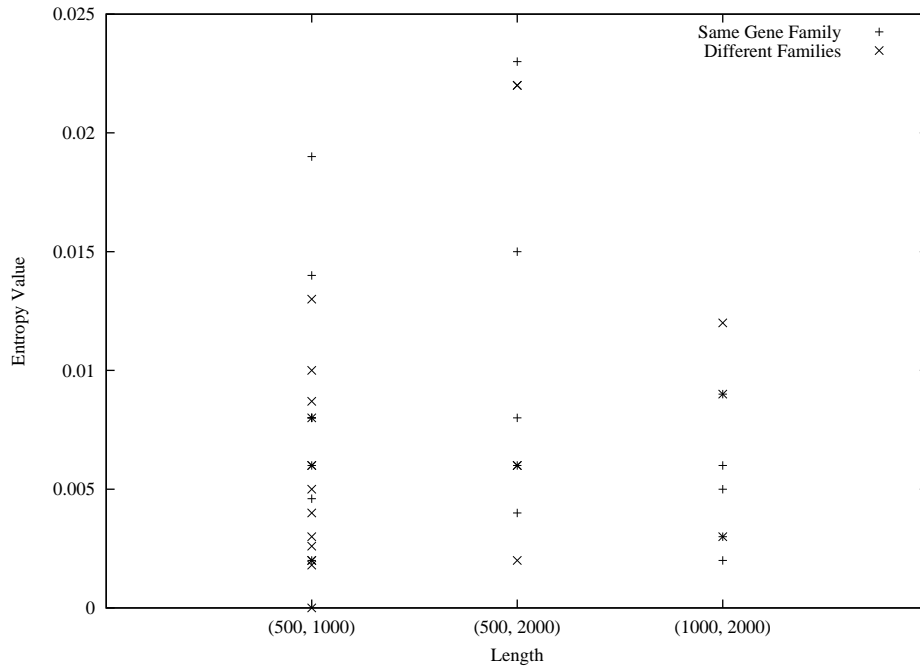


Figure 4.9: Entropy values from the gene family comparison.

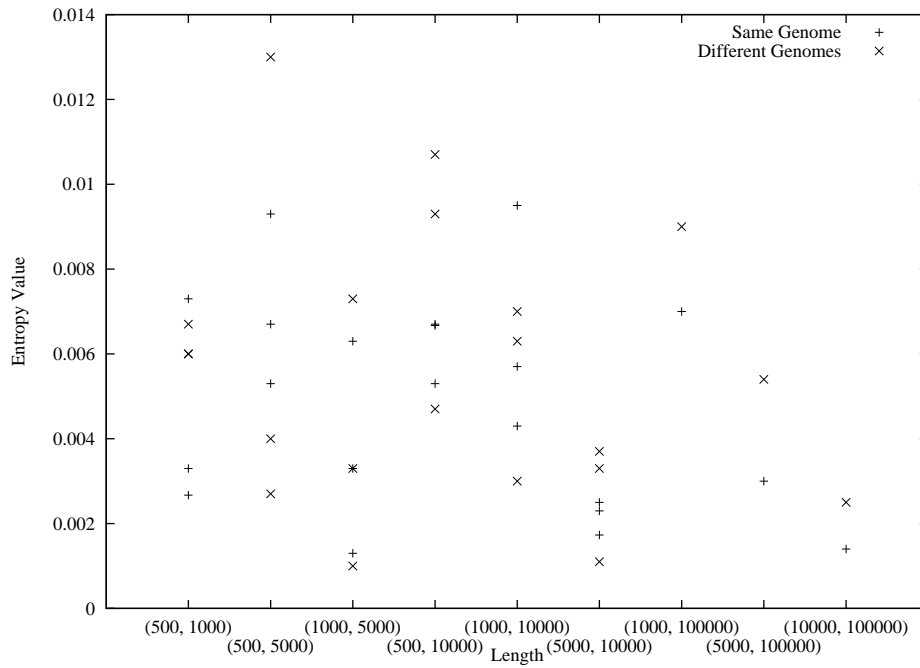


Figure 4.10: Entropy values from the genome comparison.

# Chapter 5

## Discussion

### 5.1 Introduction

This chapter discusses in depth the results presented in Chapter 4 and how they allow the research question to be answered. The results are analysed and reasons for the unexpected ones are discussed. The difference between the expected results (for which reasoning is provided) and the actual results are compared and a discussion of the results of the research in general is presented. Possible areas of future research are also described. The results lead to the conclusion that the entropy measure is not suited to differentiating between different bodies of DNA data. Reasons for this may include that small alphabet and small number of possible “words” (codons).

Section 5.2 discusses the results of the cross-region comparison and Section 5.3 covers the results of the independent region comparison. The results of the gene family and genome comparisons are analysed in Sections 5.4 and 5.5 respectively. Possible reasons for the lack of usefulness of the entropy measure are presented in Section 5.6 and the contribution of the research, and areas of future work, are outlined in Section 5.7.

### 5.2 Cross-Region Comparison

#### 5.2.1 Discussion

The cross-region comparison is the primary focus of the research, centered more on the splice junctions than on the introns and exons. Under the assumption that coding and non-coding regions would have sufficiently different optimal encodings, it was hoped that splice junctions would be indicated by large  $E_{AB}$  values. The reasoning behind this was that at a splice junction,  $A$  is in one region and  $s$  and  $B$  in the other, so  $s$  is compressed better using the encoding of  $B$  than using the encoding of  $A$ . In other words,  $s$  would display a greater entropy when compared with  $A$  than with  $B$ , since its content is more similar to that of  $B$ . Therefore,  $\Delta_{As}$  would be greater than  $\Delta_{Bs}$  and  $E_{AB}$  will be significantly greater than 0. Conversely, if  $x$  was not a splice junction then  $s$  would display similar compression levels for both sequences, making  $\Delta_{As} \approx \Delta_{Bs}$  and  $E_{AB}$  would be close to 0.

As the method generates entropy values for positions closer and closer to a splice junction, the more different the two optimal encodings should become, since more and more of the two sequences will lie on different sides of it. This should lead to a slow increase in the entropy values. When the splice junction passes through  $s$ , the values should peak and then begin to drop as it moves on through  $A$  and finally out of consideration entirely. At that stage the entire sequence should ideally fit into a single region and entropy values will be close to 0 again.

The results presented in Section 4.2 do not display such characteristics, however. There is no marked rise or drop of entropy values around the splice junctions; the areas do not display any pattern, be it like that discussed above or otherwise, that could be used to identify them.

Indeed, there is no discernible pattern around the splice junctions at all—the areas have values similar to the rest of the sequence. Instead of increasing, reaching a peak and then decreasing again, the values generally remain at an average level across the splice junctions, occasionally displaying lower or higher values. These are characteristics shared with the rest of the sequence.

It is the lack of any identifying features that leads to the conclusion that the entropy measure is not useful for identifying splice junctions.

### 5.2.2 Additional Results

There are also some other features that are not directly related to splice junctions. For all of the sequences, the bulk of the entropy values are low (within the first three bands of values). Spikes and troughs may occur at some positions, though, and one of the features of the results is the layout of these features. Despite some minor changes between the results of different length pairs, these features tend to be consistent and even become slightly more marked as the length pairs increase. The cause of these features, especially the spikes which are more noticeable, does not seem to be related to the splice junctions at all. However, there does seem to be a correlation between spikes and non-coding regions. This may be due to the predominantly greater lengths of the introns. They make up the bulk of the sequences, a fact which may simply give the appearance of high values purely because the high values have a greater chance of falling within an intron than an exon. This feature is analysed and discussed further in the next section.

As can be seen in the figures some of the regions, especially the exons, are very short. Due to this, the subsequences used by the entropy measure may have spanned multiple splice junctions, a situation which the method hoped to avoid. As a result, the entropy values may differ from those that would result if only one boundary was spanned. However, it is unlikely that these values have had much negative impact on the results—if removed or calculated differently it is highly unlikely that they would shed any additional positive light on the measure and its results. Additionally, the shape of the entropy values tend to show more definition for longer length pairs, suggesting that, at least for the sequence lengths used, longer pairs provide better results. If the pairs were chosen so as to fit inside all exons, both length values would be under 100bp in some cases. This is well under both the 200bp limit suggested by Fickett [1982] for Testcode and the 1,000bp limit implied by Benedetto *et al.* [2002]. As a result, it is unlikely that such lengths would produce entropy values with any meaning.

Another feature of the results is that the 3' end of each sequence tends to have high entropy values, especially when the region is quite large. This may be due to the fact that the end contains information about the gene and how it should be used, rather than the gene's actual data. There may be less redundancy and commonality within the 3' end than in the gene itself, causing higher entropy values. The region was not part of the research scope, however, and investigating the results is a possible avenue of future work.

### 5.2.3 Conclusion

The results indicate that the entropy measure cannot be used to identify splice junctions. The expected pattern of low values within a region and higher values at the region boundaries is not present. The lack of any pattern surrounding them—expected or otherwise—suggest that it does not differentiate between the different characteristics of the two regions sufficiently accurately to mark region boundaries. This leads to an answer for the first part of the research question: the entropy measure is not suitable for identifying introns and exons by identifying the boundaries between them.

## 5.3 Independent Region Comparison

### 5.3.1 Discussion

The results of the cross-region comparison suggest that there may be a correlation between the range of entropy values and the type of region: many of the non-coding regions seem to contain high entropy values. This comparison sheds further light on this feature of the entropy values by considering the two regions independently, thereby eliminating the bias of region size. The difference in values is not unexpected considering the less stringent biological controls imposed on introns. The more random nature of the data does suggest higher entropy values since it is less predictable.

The results of the comparison confirm that there is often a greater range in entropy values in non-coding regions. The mean entropy value is often higher than in exons and there is a greater deviation from that mean. However, the difference is neither very large nor guaranteed and it is not sufficient to make any decisions regarding region type. The results from the cross-region comparison include some exons that display high entropy values (such as the second exon in Figure 4.5), and there are some full sequences for which the trend does not hold.

A large number of positions in introns have values within the first three bands, as do exons. If introns produced persistently high entropy values along with very few low ones (ie. a high mean with a low standard deviation), or conversely if exons always produced low entropy values, the difference would possibly be more indicative of region type. The sequences do not display such characteristics, however, and the results are not sufficiently informative to be useful.

### 5.3.2 Conclusion

While there is a small increase in the mean and standard deviation of the entropy values for non-coding regions, the difference is not sufficiently marked or persistent to make it useful in identifying coding and non-coding regions. Thus, the partial answer to the research question made in Section 5.2 can be expanded upon: the entropy measure is not suitable for identifying introns and exons either by identifying the boundaries between them or by identifying the regions themselves. This completes the answer to the part of the research question pertaining to introns and exons.

## 5.4 Gene Family Comparison

### 5.4.1 Discussion

The gene family comparison considered the genes as a whole, including all non-coding regions. The basis figures were calculated by comparing genes from the same families to each other. Since genes in a particular family have similar uses and share certain characteristics their optimal encodings should be similar. The entropy measure should recognise this “closeness” by producing low entropy values. When genes from different families were compared, their entropy values were compared with the basis values in an attempt to gauge the accuracy of the measure. As the families have different uses and characteristics so do their genes, thus two genes from different families would have less similar optimal encodings and produce high entropy values.

The results, however, do not contain two such sets of values. The same-family values vary as greatly as the different-family values. Instead of being consistently low, the former are spread out across a range of both high and low values; even the two highest values are from the same-family comparison. The different-family results also show a range in values. The expected pattern of high values for different families is not present—indeed, a number of the different-family results are low.

These trends are present for all three of the length pairs. Even the longest pair, (1,000, 2,000), which should produce the most informative results judging by the success of Benedetto

*et al.* [2002], does not show any difference in the two sets of values. The differences which were expected, and which are required if the measure is to be useful, are not present.

### 5.4.2 Conclusion

In order to be used to gauge the relatedness or similarity of genes based on gene families, the measure would have to produce entropy values that are markedly different for the two sets of data (genes from the same family and genes from different families). The results do not contain any significant difference, expected or otherwise. Consequently, the second part of the research question can be answered: the entropy measure does not produce values that can be used to discriminate between genes from different families.

## 5.5 Genome Comparison

### 5.5.1 Discussion

The final genome comparison considered the genes as part of entire genomes, including introns. As in the gene family comparison, some values were first calculated to give a basis for the interpretation of later results. In this experiment, the basis values were calculated by comparing genomes to themselves. Certain characteristics of a species' genome tend to be present throughout the genome, which should make the optimal encoding of one portion of the genome similar to any other. As a result, the basis entropy values should be relatively low and uniform, in a manner similar to the basis values used in the gene family comparison. Conversely, two genomes from different species should display more differences and have less similar optimal encodings. When compared, they should produce higher entropy values than the basis values.

The distribution of the values in the results does not conform to these expectations. The basis, same-genome values range from low to high across all the length pairs, as do the different-genome values. There is no unexpected partitioning of the values either. The exclusively low values indicating similar genomes are not present and many of the different-genome values are low. While the three greatest length pairs do show the expected results, they are not sufficiently prevalent in the other results to make their occurrence truly informative. More calculations at the greater lengths were not possible due to the lengths of the genomes used, although it is doubtful that they would produce values any more informative than those produced for the shorter length pairs. It is unlikely that the increase in lengths would introduce such a definitive pattern into results that previously held no pattern at all.

### 5.5.2 Conclusion

The results of the genome comparison were entropy values that varied too greatly for a correlation between genome similarities to be the cause. The basis values were not the consistently low values that were expected, and the same-genome values ranged between both the lower and upper extremities. The results lead to an answer to the final part of the research question: the entropy measure does not produce values that can be used to differentiate between genomes of the same and different species.

This completes the answer to the research question. To reiterate:

*The compression-based entropy per character measure does not produce values that identify the similarities and differences of introns and exons, genes from different families, and entire genomes, with sufficient accuracy to discriminate between them.*

## 5.6 Reasons for the Results Obtained

DNA sequences certainly have certain statistical characteristics which can differ between introns and exons or between genes, gene families and genomes. The success of gene finding methods such as Testcode indicates this. The failure of the entropy measure to produce any useful results therefore almost certainly lies with the measure itself and not with the data on which it was used. That is, it is not that the different bodies of DNA data lacked distinguishing characteristics, but that the entropy measure was not able to capture those differences sufficiently to produce meaningful results.

There are a number of possible causes for the shortcomings of the measure. Firstly, the structural differences between introns and exons lie primarily in the distribution of basepairs within the codon—it is the preference certain basepairs have for certain positions that Testcode measures. The dictionary method used by gzip, however, does not consider three characters at a time: matches anywhere up to 258 bytes are possible. As a result, the measure may be too general to capture something as subtle as the positioning of a character within a three letter “word”.

Secondly, the four letter alphabet may be too small, especially coupled with the fact that all 64 possible codons have a biological meaning and hence are used. It is unlikely that the work by Benedetto *et al.* [2002] would have been successful if the natural language texts consisted entirely of three letter words, almost all of which were used somewhere in the text, that had been constructed from a set of four characters. The result is that the optimal encodings developed by the measure are not sufficiently specific to be useful. Instead of the compressed sequences differing by 50 or even more characters, they differ only by up to 10 characters. This, along with the great length of the dictionary matches, may also result in encodings that are attuned to the peculiarities of the sequence being compressed and not to the general pattern of which it is a part.

Thirdly, the differences in the “vocabularies” of all the pairs of data (introns and exons, gene families and genomes) were not large. The natural language texts used by Benedetto *et al.* [2002] were in entirely different languages, each with their own specific vocabulary. If one considers a codon to be a word then the bodies differ more in the frequency of each word than in the differences of the words themselves. This leads to encodings that are not very different and entropy values that are not very informative. The independent region comparison showed that the measure identifies some differences between coding and non-coding regions, at least in a limited way. These results indicate that the method is sensitive to each region’s characteristics, but in a way that is not very specific.

Lastly, the length of the sequences is potentially a contributing factor, especially in terms of the intron-exon comparison. Sequence lengths of only a few hundred basepairs (or less) are possibly not enough to compose a sufficiently detailed encoding. If each region was instead a couple of thousand basepairs more useful values might have been produced, although it is unlikely considering the effects of the factors discussed above. Each of them contributes to the measure’s success (or lack thereof) in some way. Added together, they prevent the measure from producing results that are useful.

## 5.7 Contribution and Future Work

While unsuccessful in terms of finding a new method for differentiating between types of DNA data, the research has successfully narrowed the possibilities for further measures, even if only by a small degree. It has shown that the method, while successful when applied to the diverse subject matter of natural language texts, does not have the accuracy and fidelity to gauge the differences between bodies of DNA data. The method is not suitable for discriminating between introns and exons, grouping genes based on their family, or comparing the similarity of different

genomes. Furthermore, it seems the measure and DNA data are not compatible: it lacks the facilities to make meaningful interpretations or comparisons of such data. Due to this, it is unlikely that the measure would be useful in any other area of bioinformatics.

Despite these difficulties, there are some avenues of future work that it may be informative to investigate. Firstly, the spikes and troughs found in the cross-region comparison results are potentially important. They seem to be present for all length pairs, suggesting that they are caused by a characteristic of the sequence and not the measure. The reason for them, and why the entropy measure interprets them as it does, may be interesting.

Secondly, the high entropy values found at the 3' end of the sequences are curious. Is the information in the region so well encoded that it produces high entropy values, or does it simply have a more random structure than the rest of the sequence? Does the 5' end have similar properties?

Finally, adjusting the compression method may produce more informative results. A number of the problems identified in the previous section are due to, or compounded by, the nature of gzip. Using an alternative measure, or attempting to remove some of the limitations gzip imposes, may be beneficial. However, it is unlikely that the results would be drastically altered. The characteristics of genomic data and the nature of the compressor seem too different to be reconciled.

## 5.8 Conclusion

The separate parts of the research question required four different experiments before a suitably complete answer could be formulated. The results from the cross-region comparison show that the measure is not sufficiently accurate to discriminate between introns and exons. The results show none of the expected patterns that would lead to the method being useful. Additionally, they show no other patterns that could be interpreted in a similar way.

The independent region comparison results confirm that there is a slight difference in the range of entropy values for intron regions. However, the region is neither great nor particularly marked and as such is not of much use. The results of the gene family comparison fail to exhibit the low same-family, high different-family results that were expected. Indeed, the results for both types of data range from low to high values. Similar results were found for the genome comparison: values for data from the same genome ranged just as much as values from the comparison of different genomes.

The lack of any useful patterns leads to the conclusion that the entropy measure is not sufficiently accurate to be used to differentiate between the different types of DNA data considered by the research. While producing a negative answer to the research question, the work has contributed to the body of data surrounding gene finding methods.

Possible future work includes the investigation of the peaks and troughs found in the results of the cross-region comparison, as well as the high entropy values found in the 3' end of many of the sequences. Additionally, adjusting the compression method or using an entirely new one may produce different results, although any large improvement is unlikely.

# Chapter 6

## Conclusion

### 6.1 Introduction

Finding solutions to the gene finding problem—identifying which parts of a gene are meaningful and which are not—is an active area of research. Modern techniques use computers to aid in our understanding of genomic data, providing insight into the world around us and helping to find cures for diseases. Existing techniques are not perfect, however, and results must still be verified biologically. Adding a new tool to the collection, or refining existing ones, would have many benefits.

The complexity of genomic data makes the task of understanding it greatly challenging. The signals embedded in genes that are used by mechanisms in an organism’s cells are varied, and mankind’s understanding of them are imperfect. The information and processes over which the signals have control are equally diverse and the many levels of interaction between them have yet to be fully unravelled.

In an effort to aid the attempts at doing so, this research investigated the feasibility of using compression to identify characteristics in DNA sequences. An entropy-based measure was used in an attempt to distinguish between different types of genomic data, based on each type’s optimal encoding as established by a compression algorithm.

This report documented the research; its aim, method, results and outcome. This chapter concludes by reiterating the more salient points of each of these aspects.

### 6.2 Research Overview

The primary aim of the research was an attempt to identify a new content-based technique for gene finding. In addition, it also explored the usefulness of the technique in relation to other types of DNA data.

An entropy measure was defined, based on the optimal encodings of DNA sequences produced by a compression algorithm, that attempted to determine the “distance” between two bodies of data. This distance value was a measure of the similarity of their informational content. The nature of the introns and exons found in genes, as well as the success of methods like Testcode, suggested that the two regions would exhibit sufficiently different optimal encodings for the measure to be able to distinguish between the two. By applying the measure along the length of the gene sequence, an attempt was made to identify the boundaries between the two regions based on the changes in the entropy values. As the area covered moved from one region into the next, the values were expected to fluctuate and exhibit a certain pattern, allowing the regions to be identified. This process was applied to three different genes using a range of sequence lengths.

To investigate the peaks and troughs identified in the results of this experiment, a second comparison was made that calculated entropy values for each of the two regions independently.

Instead of using the values to identify splice junctions, this comparison was aimed at using the values to identify the two regions themselves. In a manner similar to that for the first experiment, entropy values were calculated for the introns and exons of three sequences. The mean and standard deviation of each set of values were then calculated and compared.

Genes from different families were also compared using the entropy measure. The similarity in purpose of genes from the same family suggested that they would exhibit similar characteristics and hence similar optimal encodings. The measure was applied to determine if the relatedness of the genes could be established. Firstly by comparing genes from the same family to determine basis values, and then by comparing genes from different families.

Finally, a similar procedure was followed to compare parts of entire genomes. The entropy measure was used to calculate the “distance” between subsequences from certain genomes and subsequences of both the same and other genomes. This comparison ignored the distinction between coding and non-coding regions of genes, and instead considered the larger picture, approaching the genes within their context.

## 6.3 Results

The results of each comparison had similar implications for the measure. The region comparison results lacked any pattern at the region boundaries, either expected or otherwise. They did, however, contain slightly higher values for the non-coding regions. The second comparison confirmed this difference, although it was found to be neither great nor very consistent. The results from the region comparison also showed high values for the 3' ends of most genes—an area that did not fall under the scope of the work—the cause of which may prove interesting under further analysis.

The results for the gene family and genome comparisons were similar to the above. The expected pattern of values was not present; instead both experiments produced a wide range of values. The basis values that were expected to be low had just as much of a range as the rest of the values. Furthermore, no other pattern that would have allowed for the identification of the two types of data was observed.

These results led to the conclusion that the compression-based entropy per character measure was not compatible with DNA data. Where natural language texts have a large alphabet and exhibit a wide range of words, genomic data has more limited variations that the measure cannot reliably identify.

## 6.4 Conclusion

The results indicate that the measure is not able to extract sufficient information from the sequences to capture their often subtle differences. Consequently, the research concludes that the measure is suited to neither the application of gene finding nor to use with genomic data in general.

Possible areas of further work include investigating the causes of the peaks and troughs evident in the cross-region comparison and the cause of the high entropy values at the 3' end of the genes used.

While unsuccessful in terms of finding a new gene finding technique, the work has still slightly reduced the gene finding search space. While other, more successful, entropy-based measures may be found it is doubtful that they will be compression based. The subtle peculiarities of genomic data make the rather heavy-handed approach taken by compression methods unsuitable.

# References

- [Baase and Van Gelder 2000] S. Baase and A. Van Gelder. *Computer Algorithms*, chapter 6. Addison Wesley, third edition, 2000.
- [Batzoglou *et al.* 1998] S. Batzoglou, B. Berger, D. J. Kleitman, E. S. Lander, and L. Pachter. Recent developments in computational gene recognition. In *ICM: Proceedings of the International Congress of Mathematicians*, 1998.
- [Benedetto *et al.* 2002] D. Benedetto, E. Caglioti, and V. Loreto. Language trees and zipping. *Physical Review Letters*, 88(4):048702—1 – 048702—4, January 2002.
- [Bergheim 2002a] A. Bergheim. Personal communication, October 2002.
- [Bergheim 2002b] A. Bergheim. Personal communication, May 2002.
- [Burset and Guigó 1996] M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–357, 1996.
- [Deutsch 1996] P. Deutsch. *DEFLATE Compressed Data Format Specification*. Network Working Group, May 1996. Internet RFC 1951.
- [Dong and Searls 1994] S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23:540–551, 1994.
- [Fickett 1982] J. W. Fickett. Recognition of protein coding regions in DNA sequences. *Nucleic Acids Research*, 10(17):5033–5318, 1982.
- [Gailly and Adler 1993] J.-L. Gailly and M. Adler. *GZIP documentation and sources*, 1993. Accessed June 2002. <ftp://prep.ai.mit.edu/pub/gnu/gzip/>.
- [Goodman 2002] J. Goodman. Extended comment on language trees and zipping. Accessed June 2002. [www.research.microsoft.com/~joshuago/physicscomment.ps](http://www.research.microsoft.com/~joshuago/physicscomment.ps), February 2002.
- [Guigó 1998] R. Guigó. DNA composition, codon usage and exon prediction. Accessed June 2002. [www.rockefeller.edu/wli/gene/guigo99.pdf](http://www.rockefeller.edu/wli/gene/guigo99.pdf), May 1998.
- [Haussler 1998] D. Haussler. Computational genefinding. *Trends in Biochemical Sciences*, 1998.
- [Hunter 1993] L. Hunter. *Artificial Intelligence and Molecular Biology*, chapter 1, pages 1–46. MIT Press, 1993.
- [Kulp *et al.* 1996] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In *ISMB-96*, pages 134–142, St. Louis, USA, 1996. AAAI Press.
- [Lelewer and Hirschberg 1987] D. A. Lelewer and D. S. Hirschberg. Data compression. *ACM Computing Surveys*, 19(3):261–296, September 1987.

- [Lempel and Ziv 1977] A. Lempel and J. Ziv. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.
- [NCBI ] National Center for Biotechnology Information (NCBI) Website. [www.ncbi.nlm.nih.gov](http://www.ncbi.nlm.nih.gov).
- [Pevzner 2000] P. A. Pevzner. *Computational Molecular Biology: an algorithmic approach*. MIT Press, 2000.
- [Stormo and Haussler 1994] G. D. Stormo and D. Haussler. Optimally parsing a sequence into different classes based on multiple types of evidence. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 4755–4761. AAAI Press, 1994.
- [Wood 1993] D. Wood. *Data Structures, Algorithms, and Performance*. Addison-Wesley, 1993.