Université de Montréal

Methods for multi-class segmentation of molecular sequences

par

Ming-Te Cheng

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

mars, 2006

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé :

Methods for multi-class segmentation of molecular sequences

présenté par

Ming-Te Cheng

a été évalué par un jury composé des personnes suivantes:

François Major
président-rapporteur

Miklós Csűrös
directeur de recherche

Sylvie Hamel
membre du jury

Mémoire accepté le 26 avril 2006

Résumé

Le mémoire présente les modèles statistiques pour la segmentation de séquences moléculaires. Il décrit des algorithmes de segmentation et leur implémentation, en utilisant diverses méthodes de pénalisation pour la complexité du modèle, avec des restrictions possibles des longueurs de segments. Les méthodes sont illustrées sur les séquences d'ADN du bactériophage *lambda*, *Methanocaldococcus jannaschii*, et du complexe majeur d'histocompatibilité humain.

Mots clés : Segmentation, Modèles Statistiques, Isochores, Taux de GC

Abstract

The thesis discusses statistical models for the segmentation of molecular sequences. It describes segmentation algorithms and their implementation, using various penalization methods for model complexity, along with possible restrictions on segment lengths. The methods are illustrated on the DNA sequences of bacteriophage *lambda*, *Methanocaldococcus jannaschii* and the human Major Histocompatibility Complex.

Keywords: Segmentation, Statistical Models, Isochores, GC-Content

Table of Contents

1 Introduction 1

1.1 Plan . . . 2

1.2 Contributions . . . 3

2 Segmentation in Sequence Analysis 4

2.1 DNA: A Brief Introduction . . . 4

2.2 Isochores . . . 7

2.2.1 Description . . . 7

2.2.2 Proposed Causes . . . 8

2.2.3 Existence . . . 12

2.3 Other Applications of Segmentation Models . . . 15

3 Statistical Models 18

3.1 Bayesian Approach . . . 19

3.2 Hidden Markov Model . . . 21

3.3 Complexity Penalties . . . 23

3.3.1 Akaike's Information Criterion . . . 23

3.3.2 Bayesian Information Criterion . . . 26

3.3.3 Minimum Description Length . . . 27

4 Algorithmic Problems In Statistical Models **28**

4.1 Forward-Backward Algorithm 28

4.2 Viterbi Algorithm 31

4.3 Penalty-Based Best Segmentation 34

4.3.1 Description 35

4.3.2 Algorithms 36

4.4 Penalty-Based Best Segmentation with Minimum Segment Lengths . 41

4.4.1 Description 41

4.4.2 Algorithms 43

5 Experimental Results **49**

5.1 Bacteriophage Lambda 54

5.1.1 Description 54

5.1.2 Tests 58

5.1.3 Results 59

5.2 RNA Genes in Thermophiles 71

5.2.1 Description 71

5.2.2 Tests 74

5.2.3 Results 74

5.3 Major Histocompatibility Complex 93

5.3.1 Description 93

5.3.2 Tests 93

5.3.3 Results 95

6 Conclusions **98**

6.1 Discussion 98

6.2 Future Work 99

List of Figures

2.1 Molecular structure of the 2'-deoxyribose sugar. 5
2.2 Schematic molecular view of a DNA strand. 5
2.3 Molecular structure of nucleotide bases with hydrogen bonding. . . . 6
2.4 Schematic molecular view of a double strand of DNA. 7
2.5 Reciprocal and non-reciprocal recombination following crossing-over. . 11

3.1 An example of HMM modelling isochores in human genome. 24

4.1 Illustration of operations required for computing forward variable. . . 29
4.2 Illustration of computing forward variable in terms of observations and states. 30
4.3 Illustration of operations required for computing backward variable. . 32
4.4 Penalty-based best segmentation algorithm. 37
4.5 Penalty-based best segmentation with traceback array algorithm. . . 38
4.6 Traceback algorithm. 40
4.7 Maximum-likelihood estimation of segments algorithm. 40
4.8 Penalty-based minimum length segmentation algorithm. 44
4.9 Penalty-based minimum length segmentation with traceback array algorithm. 46
4.10 Traceback with minimum segment lengths algorithm. 47

4.11 Maximum-likelihood estimation of segments with minimum lengths algorithm. 48

5.1 Probability calculation algorithm. 51
5.2 Add data to traceback array algorithm. 52
5.3 Byte compression into traceback array algorithm. 53
5.4 Read data to traceback array algorithm. 54
5.5 Byte decompression from traceback array algorithm. 55
5.6 Distribution in bacteriophage λ via gradient centrifugation. 56
5.7 Distribution in bacteriophage λ via EMBOSS. 57
5.8 Bacteriophage λ comparison (2 classes, BIC and MDL). 60
5.9 Bacteriophage λ with minimum lengths comparison (2 classes, none). 62
5.10 Bacteriophage λ with minimum lengths comparison (2 classes, AIC). 63
5.11 Bacteriophage λ with minimum lengths comparison (3 classes, none). 65
5.12 Bacteriophage λ with minimum lengths comparison (3 classes, AIC). 66
5.13 Bacteriophage λ comparison (4 classes, BIC). 69
5.14 Bacteriophage λ versus EMBOSS distribution (4 classes, BIC). 70
5.15 GC-content of RNase P RNA genes in Prokaryotes versus optimal growth temperature. 72
5.16 Helical GC-content of RNase P RNA genes in Prokaryotes versus optimal growth temperature. 73
5.17 *M. jannaschii* comparison (2 classes, BIC and MDL). 78
5.18 *M. jannaschii* comparison (3 classes, BIC and MDL). 82
5.19 *M. jannaschii* with minimum lengths comparison (3 classes, BIC and MDL). 83
5.20 *M. jannaschii* comparison (6 classes, BIC and MDL). 85

5.21 *M. jannaschii* with minimum lengths comparison (6 classes, BIC and MDL). 86
5.22 *M. jannaschii* comparison (10 classes, BIC and MDL). 90
5.23 *M. jannaschii* with minimum lengths comparison (10 classes, BIC and MDl). 91
5.24 Segmented MHC sequence. 94
5.25 MHC comparison (4 classes, BIC and MDL). 95
5.26 MHC with minimum lengths comparison (4 classes, BIC and MDL). . 96
5.27 MHC with minimum lengths versus EMBOSS distribution (4 classes, BIC). 97

List of Tables

5.1 Bacteriophage λ via density centrifugation quantitative data. 56

5.2 Complexity penalty values α tested for bacteriophage λ. 58

5.3 Probability values $p_j(x)$ tested for bacteriophage λ. 58

5.4 Minimum length values tested for bacteriophage λ. 59

5.5 Distribution in bacteriophage λ for BIC and MDL tests (2 classes). . 61

5.6 Bacteriophage λ comparison (2 classes, BIC and MDL). 61

5.7 Bacteriophage λ experimental data (2 classes). 61

5.8 Bacteriophage λ with minimum lengths comparison (2 classes, none). 62

5.9 Bacteriophage λ with minimum lengths comparison (2 classes, AIC). 63

5.10 Bacteriophage λ with minimum lengths experimental data (2 classes). 64

5.11 Bacteriophage λ with minimum lengths comparison (3 classes, BIC and MDL). 65

5.12 Bacteriophage λ with minimum lengths comparison (3 classes, none). 66

5.13 Bacteriophage λ with minimum lengths comparison (3 classes, AIC). 67

5.14 Bacteriophage λ experimental data (3 classes). 67

5.15 Bacteriophage λ with minimum lengths experimental data (3 classes). 67

5.16 Bacteriophage λ with minimum lengths comparison (4 classes, BIC). . 68

5.17 Complexity penalty values α tested for *M. jannaschii*. 74

5.18 Probability values $p_j(x)$ tested for *M. jannaschii*. 75

5.19 Minimum length values tested for *M. jannaschii*. 76
5.20 RNA found for *M. jannaschii* (2 classes). 79
5.21 Experimental data for *M. jannaschii* (2 classes). 80
5.22 RNA found for *M. jannaschii* (3 classes). 81
5.23 Experimental data for *M. jannaschii* (3 classes). 84
5.24 RNA found for *M. jannaschii* (6 classes). 87
5.25 Experimental data for *M. jannaschii* (6 classes). 88
5.26 RNA found for *M. jannaschii* (10 classes). 89
5.27 Experimental data for *M. jannaschii* (10 classes). 92
5.28 Complexity penalty values α tested for MHC. 93
5.29 Probability values $p_j(x)$ tested for MHC. 94
5.30 Minimum length values tested for MHC. 94

For Mom, Dad, and Li-Te.

Acknowledgments

I would like to thank Dr. Miklós Csűrös for his invaluable friendship, inspiration, support, patience, and guidance, which were crucial in bringing this thesis in fruition.

I would like to thank my friends for their encouragement in completing this thesis.

Finally, I would like to thank my parents and my brother for their prayers, encouragement, and love.

Chapter 1

Introduction

Large-scale sequencing projects like the Human Genome Project produce a great wealth and variety of sequence data. Molecular sequences in sequence data banks such as GenBank of the National Center of Biotechnology Information (NCBI) add up to more than 100 billion base pairs now. There is an increasing need of developing efficient tools to analyze these sequences. A class of analysis methods involves sequence segmentation, consisting of dividing a sequence into fairly homogeneous parts by some measure of homogeneity.

Csűrös (2004) investigated the problem of determining maximum-scoring segment sets that can be applied to a number of molecular biology problems, such as DNA and protein segmentation. To calculate potential segment sets for a given sequence, Csűrös presented a number of fast algorithms in which different statistical models were used for two classes. In our research work, we demonstrate how sequence segmentation can be carried out efficiently using various statistical models with multiple classes.

1.1 Plan

In Chapter 2, we provide a brief introduction to DNA and isochores. We also give a description of the proposals that eventually led to the existence of the isochore theory. Furthermore, we present arguments pertaining to the actual existence of isochores. In addition to problems associated with the sequence segmentation based on isochore content, we discuss different applications of segmentation algorithms to other problems.

In Chapter 3, we present statistical models that can be used in segmentation, including Bayesian and hidden Markov model, as well as statistical notions of complexity.

In Chapter 4, we present the algorithms that can be used to solve the problems encountered in segmentation statistical models. We also present the penalty-based best segmentation model. First, we provide a description of how this model can be implemented through dynamic programming. Secondly, we present the implemented algorithms without and with traceback. Thirdly, we show how this model can be incorporated in maximum-likelihood estimation. Finally, we demonstrate how minimum segment length values can be incorporated into this model.

In Chapter 5, we present the experiments used to evaluate our implemented algorithms on three sequences: bacteriophage-λ genome, the genome of *Methanocaldococcus jannaschii* (*M. jannaschii*), and the sequence of the major histocompatibility complex (MHC) on human chromosome 6. We present the tests used for each sequence and the observed results.

Chapter 6 provides a summary of the discussed segmentation concepts and, most importantly, our analysis of the experimental results.

1.2 Contributions

To carry out our sequence segmentations, we extend the two-class algorithms presented by Csűrös (2004) to propose two algorithmic models that uses an arbitrary number of classes: the penalty-based best segmentation model and the penalty-based best segmentation with minimum segment lengths model. Although both models partition a given sequence using maximum-likelihood estimation, only the latter takes minimum segment lengths into consideration. We also incorporated the best segmentation score for the implemented traceback algorithms to refer to when determining the best segmentation of a sequence. For model parameter estimation, we proposed the use of Laplace pseudo-counters. As well, we incorporated the data compression algorithm in order to reduce the amount of allocated memory. We implemented our segmentation models using Java and evaluated the different penalization methods on bacteriophage-λ, *M. jannaschii*, and MHC on human chromosome 6. Finally, we conducted a RNase P analysis of helical and non-helical GC-content versus optimal growth temperature on Prokaryotes.

Chapter 2

Segmentation in Sequence Analysis

2.1 DNA: A Brief Introduction

Deoxyribonucleic Acid (DNA) is a nucleic acid that contains the genetic information required for any organism to function biologically (Watson and Crick 1953). As initially proposed by James Watson and Francis Crick in 1953, it is characterized as a double helix where each nucleotide base in one strand is bonded to a base in the other strand.

Each strand is a chain of repetitive units known as nucleotides. A nucleotide consists of a 2'-deoxyribose sugar and a phosphate group with a so-called *base*. The molecular structure of the sugar is illustrated in Figure 2.1, where the numbered values represent carbon atom positions. DNA is measured in base pairs, that is, kbp (thousand base pairs), Mbp (million base pairs), and Gbp (billion base pairs) are "units" used in the Biotechnology community.

As shown in Figure 2.2, nucleotides can form a polynucleotide chain by connecting to each other through a covalent bond between the 3'-carbon of one nucleotide, the phosphate residue, and the 5'-carbon of the next unit. The sugar and phosphate molecules are represented as "r" and "p" symbols, respectively.

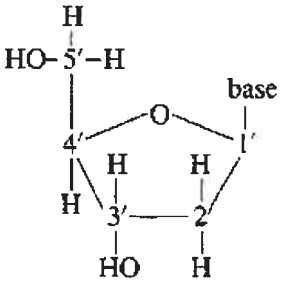


Figure 2.1. Molecular structure of the 2'-deoxyribose sugar (Setubal and Meidanis 1997).

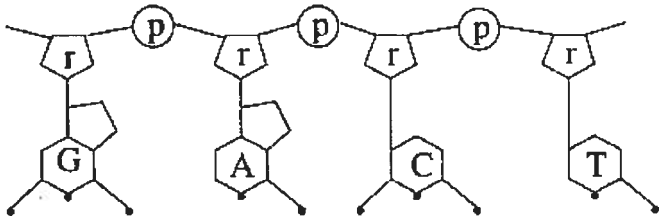


Figure 2.2. Schematic molecular view of a DNA strand (Setubal and Meidanis 1997).

DNA molecules normally comprise two polynucleotides, called *strands*. Each 1'-carbon in the strand contains a nucleotide base attached to it, which can be one of the four different types: adenine (A), guanine (G), cytosine (C), and thymine (T). Nucleotide bases can be categorized into two main groups, namely, purines (A and G) and pyrimidines (C and T). The strands are connected together by forming hydrogen bonds at the bases as illustrated in Figure 2.3, where A-T and C-G are defined as complementary or Watson-Crick base pairs. Figure 2.4 provides a schematic molecular structure view of a double strand of DNA. A DNA molecule is determined thus by the sequence of bases on one of its strands, represented as a sequence of characters over the alphabet $\Sigma = \{A, C, G, T\}$.

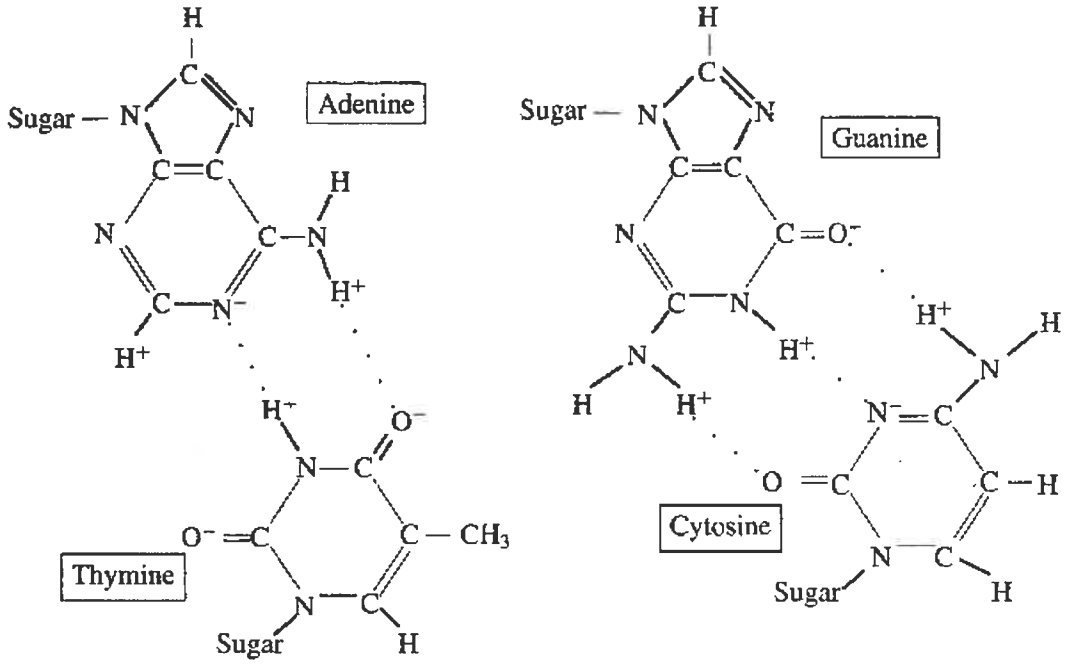


Figure 2.3. Molecular structure of nucleotide bases with hydrogen bonding (dotted lines) (Setubal and Meidanis 1997).

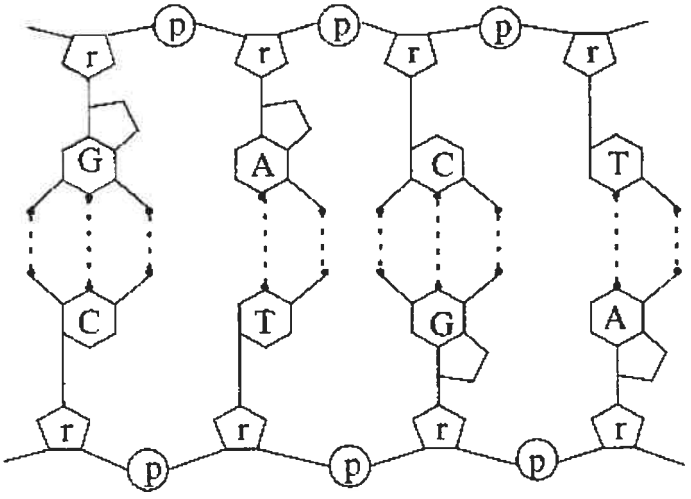


Figure 2.4. Schematic molecular view of a double strand of DNA (Setubal and Meidanis 1997).

2.2 Isochores

Sequence segmentation involves dividing a given sequence into fairly homogeneous parts by some measure of homogeneity. As an example, we discuss here the segmentation of a DNA sequence into so-called isochores.

2.2.1 Description

A profile of guanine and cytosine (GC) levels can be used to characterize variation along chromosomes, where the natural partition of a chromosome sequence is defined as abrupt changes in GC level (Paces et al. 2004). GC levels are correlated with key biological properties in many eukaryotes, such as gene density changes, replication timing switches, and differences between the locations of adjacent regions in the interphase nucleus.

Bernardi (2000) proposed the isochore theory which describes the structural com-

position of the genomes of warm-blooded vertebrates. Unlike the classical work of Meselson et al. (1957) where CsCl density gradients were used in equilbrium centrifugation to reveal broad, assymetrical bands in DNA, Filipski et al. (1973) proposed that high-resolution fractionation is possible by using equilibrium centrifugation in Cs_2SO_4-Ag^+ density gradients instead to partition DNA-silver complexes according to the frequency of silver-binding sites on DNA molecules. When applied to bovine DNA, this technique revealed three distinct families of fragments (comprising 85% of the genome) having different GC content. This subsequently led to the discovery that DNA fractionation reveals the compositional heterogeneity of high molecular weight, "main band" (i.e. non-satellite, non-ribosomal) bovine DNA. Furthermore, the laboratory concluded that vertebrate genomes are comprised of a mosaic of isochores, which are defined as long DNA segments of more than 300-kbp that are compositionally homogeneous and belong to a small number of families characterized by different GC levels.

These isochores reflect a level of genome organization (Eyre-Walker and Hurst 2001) since it is observed that GC-rich components of the genome yielded a higher number in terms of gene density, short interspersed repetitive DNA elements, and recombination frequency. For GC-poor components of the genome, however, it is found that they almost exclusively possess long interspersed repetitive DNA elements.

2.2.2 Proposed Causes

Scientists are interested in finding an explanation of why there is a large-scale variation in base composition along chromosomes. It was suggested that variation could be due to three non-mutually exclusive processes: mutation bias, natural selection, and biased gene conversion.

Mutation Bias

Different base composition can be expected if there are different mutation processes acting on different parts of the genome. With a simple probabilistic model, assume that each nucleotide may mutate independently by the same mutation probabilities along the sequence. The base composition will converge towards the stationary probabilities determined by the substitution probabilities. If the mutation probabilities are not identical along the genome, then the base composition will vary too.

Wolfe et al. (1989) noted that the concentrations of free nucleotides affect the pattern of base misincorporation during DNA replication. For example, G and C nucleotides tend to be preferentially misincorporated into DNA that is replicated in a pool of free nucleotides rich in G and C. They also observed that free nucleotide concentrations vary during the cell cycle, and that different parts of the genome are replicated at different times. Wolfe et al. concluded that the regions of the genome with different replication times should have different mutation patterns, and, ultimately, different base compositions.

Filipski (1987) hypothesized that variation in the efficiency of DNA repair might be responsible for the formation and maintenance of isochores. He reasoned that this is due to the variable efficiency of certain types of DNA repair and that some types of pair are known to be biased, thereby causing variation in the pattern of mutation.

Fryxell and Zuckerkandl (2000) suggested that isochores are a consequence of cytosine deamination, which is defined as the reaction of a water molecule with the amino group on position 4 of the pyrimidine ring of cytosine, thereby resulting in the conversion of cytosine to uracil (Eyre-Walker and Hurst 2001). They found that the deamination of methyl-cytosine and cytosine (i.e. C to T and C to U, respectively) is expected to occur more easily in AT-rich DNA than in GC-rich DNA since the former tends to be more unstable. They proposed that an isochore structure can be assembled if a DNA sequence somehow becomes GC-rich, consequently causing

a reduction in cytosine deamination and an increase in GC content in the surround areas.

Natural Selection

Bernardi and Bernardi (1986) suggested that isochores are the consequence of natural selection, which is defined as the differential multiplication of mutant types. This occurs through either negative selection, that is, the elimination of organisms with deleterious mutations, or positive selection via the preferential propogation of organisms with advantageous mutations with respect to environmental pressures. One hypothesis implies that natural selection acts upon the increased thermostability of DNA caused by GC-enrichment in order to adapt to any temperature increase in warm-blooded vertebrates, where there is a tendency for GC-rich DNA to be more thermally stable than AT-rich DNA. To demonstrate this hypothesis, Bernardi referred to the isochores found in the human genome, where the GC-richest and gene-richest isochores found in a set of R(everse)-chromosomal bands coincide with the T(elomeric) chromosomal bands previously identified as particularly resistant to thermal denaturation (Saccone et al. 1993). As well, a difference in amino-acid compositions and hydropathies between GC-rich and GC-poor isochores was observed.

Unlike the observations made for warm-blooded vertebrates, previous work concluded that there was no correlation between GC-content and habitat temperature in the case of prokaryotes (Galtier and Lobry 1997; Hurst and Merchant 2001) and cold-blooded vertebrates (Belle et al. 2002; Ream et al. 2003). Furthermore, Vinogradov (2001) observed that the bendability of genomic sequences of warm-blooded vertebrates increased faster than their thermostability as the GC-content increased. Hence, Vinogradov (2003) proposed an alternative hypothesis in which the formation of isochores was primarily due to the bendability, and not thermostability, of the DNA molecule for active transcription in the GC-rich regions and for gene suppression in the GC-poor regions.

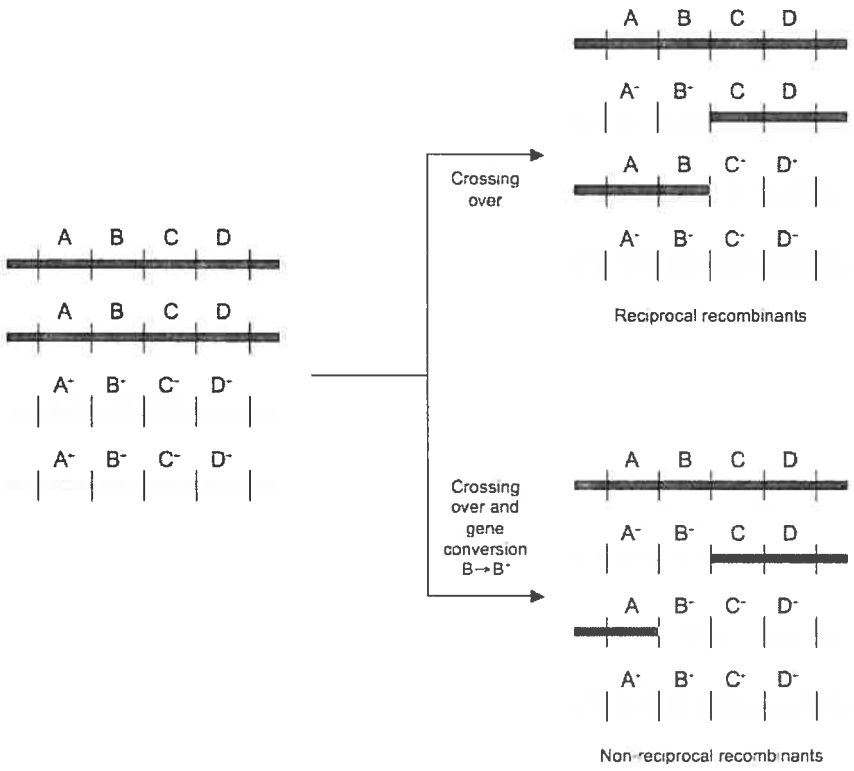


Figure 2.5. Reciprocal and non-reciprocal recombination following crossing-over (Watson 2004).

Biased Gene Conversion

Gene conversion is defined as a non-reciprocal recombination process that causes one sequence to be converted into the other. An illustration of a crossing-over between two chromatids (double-stranded DNA molecules) is shown in Figure 2.5. We define an *allele* as an alternative form of a gene that occupies a specific position on a chromosome. Suppose that the strands contain a number of alleles as indicated by alphabetical letters. When crossing-over occurs, there are two possible recombinants: reciprocal, where an equal 2:2 segregation of the entering alleles is observed; non-reciprocal, where a 3:1 segregation and gene conversion of allele B to B^+ is exhibited.

Biased gene conversion is when the two possible directions occur with unequal probabilities (Eyre-Walker and Hurst 2001). Biased gene conversion would lead to a base mismatch if the heteroduplex DNA extends across a heterozygous site. These base mismatches are sometimes repaired by the DNA-repair machinery, although they tend to be biased and thereby leading to an excess of one allele (i.e. one of the different forms of a gene or DNA sequence that can exist at a single locus) in reproductive cells. Their hypothesis is based on two observations that demonstrates the correlation between the recombination rate and GC content, that is,

1. There is a correlation between the frequency of recombination and GC content both between and within human chromosomes.

2. Sequences that have stopped recombining are either declining in GC content, or have a lower GC content than their recombining paralogues (i.e. a locus that is homologous to another in the same genome).

2.2.3 Existence

Demonstrated by the numerous amount of studies since its first proposition, the isochore theory is considered to be the most reliable method of studying the long-range compositional structures of metazoan genomes within an evolutionary framework (Cohen et al. 2005). Furthermore, it is generally assumed that isochores exist and that they contain genes with corresponding GC contents. When the draft of the human genome was first proposed, however, some scientists questioned about the existence of isochores or at least their usefulness.

The authors of the initial draft of the human genome studied the genome sequence to examine whether strict isochores could be identified (Lander et al. 2001). The authors defined the term "strict isochores" as sequences that cannot be distinguished from random sequences, in which every nucleotide is free to change. To illustrate

their argument that isochores do not merit the prefix "iso", Lander et al. divided the human genome sequence into 300-kbp windows and subdivided each window into 20-kbp subwindows. From calculating the average GC content of each window and subwindow, and from examining the relationship between the variance of the GC content in the subwindows and the average GC content in each window. Lander et al. concluded that the idea of isochores being strictly homogeneous can be ruled out since the residual variance was too large to be consistent with a homogenous distribution.

Despite the arguments of Lander et al., Li et al. (2003) maintained that isochores do merit the prefix "iso" . Li et al. presented two points in which the strict isochores correspond to the isochore concept originally developed and defined by Bernardi. First, Bernardi (2001) defined isochores as *fairly* homogeneous regions. Unlike Lander et al., Li et al. considered the GC% mean values and variance to be two independent parameters of a statistical distribution, and found support for the isochore theory in the human genome sequence.

Cohen et al. (2005) examined whether it is possible to provide a concrete definition of isochores so that the human genome can be described as isochoric. They also studied the extent to which each isochore can be classified into a particular isochore family. Their work was based on a number of criteria that they have proposed to describe the properties of isochores, namely:

1. *Characteristic GC content* : An isochore is a DNA segment possessing a characteristic GC content that significantly differs to those in adjacent regions.

2. *Homogeneity* : An isochore is more homogeneous in its composition than the chromosome on which it resides.

3. *Minimum segment length* : The length of an isochore typically exceeds 300-kbp.

4. *Genome coverage* : The overwhelming majority of the human genome consists of segments satisfying the first three criteria.

5. *Isochore families* : The human genome is composed of five isochore families, each with different Gaussian-distributed GC content.

6. *Isochore assignment into families* : Each isochore can be associated with a isochore family based only on its compositional properties.

To identify and quantify isochoric regions, Cohen et al. used a binary recursive segmentation procedure proposed by Bernaola-Galván et al. (1996) to partition the human genome. The method repeatedly splits the sequence based on an entropy measure to maximize the difference between neighbouring regions. A statistical test, similar to the one of Li et al. (2003) was used. The segmentation procedure revealed that the distribution of segment lengths does not have a characteristic length scale. Cohen et al. (2005) observed isochores span less than half of the sequenced portion of the human genome if they satisfy the first three attributes, but found that alternative isochores with lower cutoff lengths also satisfy the same criteria. Although the human genome is traditionally described using five isochore families, Cohen et al. found that four families already capture the GC content distribution. Finally, Cohen et al. questioned the use of the Gaussian model to define isochores since they did not find any evidence of a robust multi-Gaussian description of alternative sets of isochores. Due to overlaps found between candidate families, they also had difficulty in reliably classifying the segments into families by compositional properties.

Although the existence of isochores remains debatable up to this point, all studies agreed that the definition of isochores is relative. Furthermore, they agreed that the genome does contain large homogeneous regions of distinctive GC content, and is worthwhile to redefine the isochore concept to describe the dynamics of GC content within the human genome.

2.3 Other Applications of Segmentation Models

In addition to isochores in higher vertebrates, there is a wide variety of segmentation models which are used to answer a number of biological problems. For instance, genome alignments exhibit a mosaic structure where segments correspond to regions with different evolutionary pressure.

Horizontal gene transfer between bacteria, as demonstrated in a wide variety of ecosystems, plays an important role in the acquisition of adaptive traits, such as pathogenicity, resistance to antibiotics or heavy metals like mercury and arsenic. Furthermore, it is considered to be heavily influential in bacterial evolution. Bacteria are known to integrate prophages and have other ways of integrating foreign DNA sequences through DNA segments. Nicolas et al. (2002) used a statistical segmentation-based approach to study heterogeneities in the *Bacillus subtilis*. Specifically, they applied hidden Markov models in which each type of segment is characterized by its own statistical oligonucleotide composition. Their objectives were to reconstruct segments from DNA sequences and characterize the identified segment types, with the aim of investigating correlations between segment types and biological DNA features such as horizontal gene transfers. From their analysis, they revealed a number of heterogeneities including those related to horizontal gene transfer, the GT richness of hydrophobic proteins, and the codon usage frequency of highly expressed genes.

The application of segmentation models to biological problems is not exclusive to DNA sequence analysis. Romero et al. (1997) proposed methods that predict locally disordered (so-called low complexity) regions that are based on physiochemical features of a set of relatively short domains found in proteins of an otherwise known structure. Wootton and Federhen (1993) introduced the SEG algorithm which automatically partitions protein sequences into low- and high- complexity segments.

Functional regions in genomic sequences were traditionally predicted by identifying features associated with genes or regulatory regions. Functional regions tend to

be conserved in sequences that have evolved from a common ancestor. In contrast, non-functional regions are more likely to mutate. Consequently, functional regions can be identified in genomes using sequence comparison. Non-functional regions need to be statistically diverged so that statistical procedures can distinguish them from functional regions. Because of this, features present only at close evolutionary proximity are lost, thereby limiting the usefulness of such comparisons (McAuliffe et al. 2004). Boffelli et al. (2003) used *phylogenetic shadowing* which involves segmenting alignments between closely related species into regions with high- and low- mutation rates. This technique would enable the localization of regions of collective variation and complementary regions of conservation, thus facilitating the identification of coding and non-coding functional genes. Using the phylogenetic shadowing concept, McAuliffe et al. proposed the generalized hidden Markov phylogeny (GHMP) in order to determine the genomic sequences systematically, where the GHMP is presented as a directed graphical model (Jordan and Sejnowski 2001).

Segmentation models can also be implemented to partition proteins into a number of segments. Krogh et al. (1994) applied hidden Markov models to the problems associated with statistical modelling, database searching, and multiple alignment of protein families and protein domains. To construct their hidden Markov model, they defined the 20 amino acids from which protein molecules are composed of as *states* and the strings of amino acids that form the primary protein sequence as *observations*. For each set of proteins, their model represents one in which high probability to the sequences in that particular set are assigned.

A related application of segmentation models to protein partitioning is the topology prediction of helical transmembrane proteins. Tusnády and Simon (1998) proposed the HMMTOP method, which is based on the hypothesis that the difference in the amino acid distributions in various structural parts of these proteins determine the localizations of the transmembrane segments and the topology. They constructed a hidden Markov model that consisted of five states that describe transmembrane

protein structure: inside loop, inside helix tail, membrane helix, outside helix tail, and outside loop. Rather than accounting only the absolute amino acid compositions of various structural parts, their approach involves finding the combination of states that yield maximal divergences in the amino acid distribution. Alternatively, Krogh et al. (2001) presented the TMHMM method in which they take the alternation between cytoplasmic and non-cytoplasmic loops in helical transmembrane proteins into consideration. Their hidden Markov model consisted of seven types of states: helix core, helix caps on either side of the membrane, short loop on cytoplasmic side, short and long loop on non-cytoplasmic side, and a globular domain state. Each state contains a probability distribution over the 20 amino acids that characterizes the variability of amino acids in the region it models. The amino acid and transition probabilities were calculated from techniques that compute the maximum posterior probabilities given a prior and the observed frequencies. The TMHMM method predicts the transmembrane helices by determining the most probable topology given the hidden Markov model.

Chapter 3

Statistical Models

The advantage of probabilistic models is that they can describe the relationships between various quantities while considering the underlying uncertainty associated with them. This leads to the efficient use of available information when making predictions about biological sequences (Liu and Lawrence 1999). Statistics is mainly focused on making inferences, which can be defined as the process of deriving conclusions from facts and premises. In our case, the facts are the observed data, the premises are represented by a probabilistic model of biological sequences, and the conclusions are related to the unobserved quantities. This chapter provides a description of a number of statistical models that can be used to segment a given molecular sequence.

Let Σ be a finite alphabet; for DNA sequences, $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. We consider a DNA sequence $\mathbf{x} = x_1 x_2 \ldots x_n$ as the observed value of a sequence of random variables $\mathbf{X} = X_1 X_2 \ldots X_n$. We define segments as a continuous interval $[i, j]$ such that $z_i = z_{i+1} = \ldots = z_j$. The distribution of $\mathbf{X}$ is determined by a sequence of hidden variables $\mathbf{z} = z_1 z_2 \ldots z_n$ which define these segments, where $z_i \in \{0, \ldots, k-1\}$. The segmentation vector, $\mathbf{z}$ is the value taken by a random variable $\mathbf{Z} = Z_1 Z_2 \ldots Z_n$. The distribution of each X_i is completely determined by Z_i through the probabilities

$p_j(x) = P\{X_i = x|Z_i = j\}$. Notice that we assume that these distributions do not depend on i.

For a given segmentation, the likelihood of the observed sequence is written as

$$L(\mathbf{z}) = P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\}. \tag{3.1}$$

The main objective of using statistical models is to determine $\mathbf{z}$.

Equation 3.1 can be expanded as

$$L(\mathbf{z}) = \prod_{i=1}^{n} p_{z_i}(x_i). \tag{3.2}$$

The log-likelihood function can be derived as

$$\begin{aligned} l(\mathbf{z}) &= \log L(\mathbf{z}) \\ &= \sum_{i=1}^{n} \log p_{z_i}(x_i) \\ &= \sum_{i=1}^{n} \log p_0(x_i) + \sum_{i=1}^{n} \log \frac{p_{z_i}(x_i)}{p_0(x_i)}. \end{aligned} \tag{3.3}$$

(In this thesis, log denotes natural logarithm.) The formula can be interpreted in the following manner: the first term is the null hypothesis that all the x_i are in class 0, and the second term denotes the log-likelihood ratio for the alternative hypothesis defined by $\mathbf{z}$.

3.1 Bayesian Approach

Classical (or frequentist) statistics such as maximum-likelihood estimation interpret their probabilities as purely frequencies or ratios. In contrast, the Bayesian approach models consider their probability distributions as a measure of belief in a proposition.

A Bayesian approach allows for prior knowledge and reasonable prior concepts to be built into statistical analysis (Ewens and Grant 2001). Bayesian analysis involves determining a joint probability and defining the appropriate posterior distribution by using the calculated joint probability and the observed data.

In our case, the goal of using the Bayesian approach is to determine $\mathbf{z}$. A number of different hypotheses about $\mathbf{Z}$ can be compared by using the posterior probability, that is

$$P\{\mathbf{Z} = \mathbf{z}|\mathbf{X} = \mathbf{x}\}. \tag{3.4}$$

The likelihood function as presented in Equation 3.4 can be derived by calculating the joint probability, which is $Joint = Likelihood \times Prior$ or

$$P\{\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}\} = P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\} P\{\mathbf{Z} = \mathbf{z}\}. \tag{3.5}$$

By the definition of conditional probabilities,

$$P\{\mathbf{X} = \mathbf{x}\} = \sum_{\forall \mathbf{z}} P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\} P\{\mathbf{Z} = \mathbf{z}\}. \tag{3.6}$$

The posterior distribution can found through Bayes' theorem,

$$\begin{aligned} P\{\mathbf{Z} = \mathbf{z}|\mathbf{X} = \mathbf{x}\} &= \frac{P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\} P\{\mathbf{Z} = \mathbf{z}\}}{P\{\mathbf{X} = \mathbf{x}\}} \\ &= \frac{P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\} P\{\mathbf{Z} = \mathbf{z}\}}{\sum_{\forall \mathbf{z}} P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\} P\{\mathbf{Z} = \mathbf{z}\}}. \end{aligned} \tag{3.7}$$

Since the denominator of the equation will be the same for all choices of $\mathbf{z}$ when $\mathbf{x}$ is fixed, the best $\mathbf{z}$ can be found by maximizing the numerator, that is,

$$M(\mathbf{z}) = P\{\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}\} P\{\mathbf{Z} = \mathbf{z}\}. \tag{3.8}$$

The $\mathbf{z}$ that maximizes $M(\mathbf{z})$ is called the *maximum a-posterior* (MAP) segmentation.

A number of different approaches can be used to deal with the prior distribution $P\{\mathbf{Z} = \mathbf{z}\}$. One idea is to assume that $\mathbf{z}$ is a uniform distribution, that is, every segmentation is equally possible. Using this assumption, the probability of obtaining $\mathbf{z}$ is

$$\begin{aligned} P\{\mathbf{Z} = \mathbf{z}\} &= \prod_{i=1}^{n} \frac{1}{k} \\ &= \frac{1}{k^n}. \end{aligned} \tag{3.9}$$

Applying Equation 3.9 to the joint probability equation, we will have

$$M(\mathbf{z}) = \frac{1}{k^n} \prod_{i=1}^{n} p_{z_i}(x_i). \tag{3.10}$$

Clearly, $M(\mathbf{z})$ is maximized when $p_{z_i}(x_i)$ is maximal in each position i. Such a segmentation maximizes the likelihood but then every i can be a segment of length 1, which hardly captures any meaningful pattern in the data. In order to avoid such cases of potential "overfitting", the space of acceptable segmentations must be restricted either by imposing different prior distributions, or by using other statistical model estimation methods than MAP. Prior distributions may impose a fixed (or bounded) number of segments, or bounds on segment lengths. Another popular prior distribution is defined by a Markov model, as discussed in the forthcoming Section 3.2. An alternative to MAP estimation is to use complexity penalties as presented in Section 3.3.

3.2 Hidden Markov Model

The hidden Markov model (HMM) can be described as a series of observations by a "hidden" stochastic process (Krogh et al. 1994). In a tutorial paper, Lawrence Rabiner demonstrated how the hidden Markov model can be applied to problems in

speech recognition (Rabiner 1989). In this case, sounds forming a word represent the observations while the model is one that generates these sounds through its hidden random process in which a probability distribution is defined over possible sound sequences (Krogh et al. 1994). Ideally, a good word model would assign high probability to likely modelled sound sequences and low probability to other sequences.

In computational molecular biology research, the hidden Markov model was implemented in a number of applications including protein multiple alignment and functional classification (Krogh et al. 1994), protein folding prediction (Di Francesco et al. 1997), bacterial and eukaryotic gene recognition (Burge and Karlin 1997; Kulp et al. 1996; Henderson et al. 1997), DNA functional site analysis and prediction (Crowley et al. 1997), and nucleosomal DNA periodical pattern identification (Baldi et al. 1996). The first application of HMMs to genetic data was proposed by Churchill (1989) who used it to segment mitochondrial and phage genomes by nucleotide composition. The model makes the assumption that the different segments can be classified into a finite set of states, where the nucleotide data is assumed to follow some probability distribution. The states are assumed to randomly switch from one to the other with low probability.

In our statistical segmentation model, HMMs represent the case when $Z_1, \ldots, Z_n$ form a Markov chain with states $\{0, \ldots, k-1\}$. The prior distribution is specified as follows. Let the initial state distribution be $\pi = \{\pi_j\}$ where

$$\pi_j = P\{z_1 = j\} \qquad 0 \leq j \leq k-1.$$

denote the probability of transition between states j and j' by $t_{j-j'}$. The probability of obtaining the sequence of hidden variables (i.e., a particular segmentation) is computed by taking the transition probabilities between states into consideration, that

is

$$P\{\mathbf{Z}=\mathbf{z}\} = \pi_{z_1} t_{z_1 \to z_2} t_{z_2 \to z_3} \ldots t_{z_{n-1} \to z_n} \tag{3.11}$$

Bernardi et al. described the human genome as a mosaic of segments representing isochores (Bernardi 2000; Bernardi 2001). Furthermore, their research revealed that there are five different classes of isochores in the human genome that are defined by their GC-levels: H1, H2, H3, L1, and L2. Figure 3.1 illustrates how the isochore structure of the genome can modelled by a hidden Markov model. The different isochore classes are the states.

Now, using $M(\mathbf{z}) = P\{\mathbf{X}=\mathbf{x}, \mathbf{Z}=\mathbf{z}\}$ from Equation 3.8, the joint probability can be written as

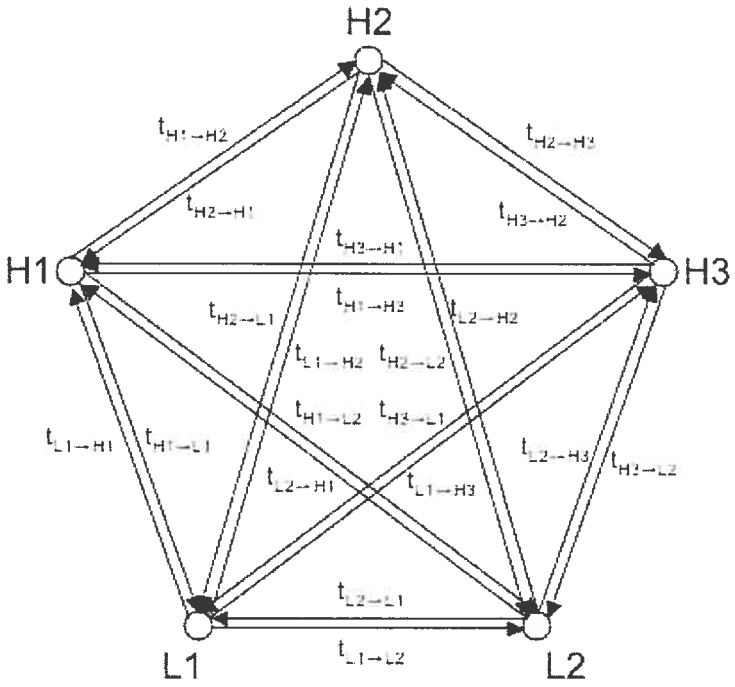
$$\begin{aligned} P\{\mathbf{X}=\mathbf{x}, \mathbf{Z}=\mathbf{z}\} &= P\{\mathbf{X}=\mathbf{x}|\mathbf{Z}=\mathbf{z}\} P\{\mathbf{Z}=\mathbf{z}\} \\ &= \pi_{z_1} p_{z_1}(x_1) t_{z_1 \to z_2} p_{z_2}(x_2) t_{z_2 \to z_3} \ldots t_{z_{n-1} \to z_n} p_{z_n}(x_n). \end{aligned} \tag{3.12}$$

3.3 Complexity Penalties

This section reviews some alternatives to imposing prior probabilities in order to handle the overfitting when maximizing the likelihood. The basic idea is to maximize the sum of the log-likelihood and a so-called complexity penalty. We discuss three penalization methods: Akaike's information criterion, the Bayesian information criterion, and the principle of minimum description length.

3.3.1 Akaike's Information Criterion

The maximum likelihood principle is encountered in two different branches of statistical theories: estimation theory through the maximum likelihood method, and test theory through the log-likelihood ratio. Akaike (1974) argued that the quantities



x	C	C	G	A	C	T	C	C	T	A	G	...
z	H1	H1	H2	L1	L2	L2	H3	H1	H1	H3	H2	...

	States				
	H1	H2	H3	L1	L2
P(C)	P_1	P_2	P_3	P_4	P_5
P(G)	P_1	P_2	P_3	P_4	P_5
P(A)	$1-P_1$	$1-P_2$	$1-P_3$	$1-P_4$	$1-P_5$
P(T)	$1-P_1$	$1-P_2$	$1-P_3$	$1-P_4$	$1-P_5$

Figure 3.1. An example of HMM modelling isochores in human genome. Note that the actual isochores found in the human genome are much longer than those depicted here.

obtained from maximum likelihood estimates are most sensitive to small variations of the parameters around the true statistical model. As an estimate of a measure of fit of a statistical model, he proposed an information criterion (AIC), which can be described as an extension of the maximum likelihood principle for problems in which the final estimate of a finite parameter model can be calculated when presented alternative maximum likelihood estimates from various restrictions of the model. For this paper, we consider "Akaike's information criterion" as an equivalent term to "an information criterion".

Akaike used the Kullback-Leibler divergence function to find the minimum difference between two probability distributions, namely, the true and approximate segmentation models. Suppose we use a DNA sequence $\mathbf{x} = x_1 x_2 \dots x_n$ as a sequence of observed values of a sequence of random variables $\mathbf{X} = X_1 X_2 \dots X_n$ and a sequence of hidden variables $\mathbf{z} = z_1 z_2 \dots z_n$ that defines the segments, where $z_i \in \{0, \dots, k-1\}$ and $\mathbf{z}$ is derived from a random variable $\mathbf{Z} = Z_1 Z_2 \dots Z_n$. As well, we define $f(\mathbf{x}) = P\{\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}^*\}$ and $g(\mathbf{x}) = P\{\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}\}$ where $\mathbf{z}^*$ is the true segmentation and $\mathbf{z}$ the approximate. Then, the Kullback-Leibler divergence function can be written as

$$\begin{aligned} KL &= \sum_{\mathbf{x}} f(x) \log \frac{f(x)}{g(x)} \\ &= \sum_{\mathbf{x}} f(x) \log(f(x)) - \sum_{\mathbf{x}} f(x) \log(g(x)) \end{aligned} \tag{3.13}$$

Note that the first term of Equation 3.13 is fixed since there can be only one true segmentation probability model. Also, notice that if the approximate segmentation probability model is the same as the true model, that is, if $g(\mathbf{x})$ is the same as $f(\mathbf{x})$, then the Kullback-Leibler divergence will be zero. This demonstrates that the model with minimal Kullback-Leibler divergence will be considered as the best estimated sequence $\mathbf{z}$.

To obtain the optimum estimated sequence $\mathbf{z}$, Akaike suggested that we can determine the linear approximation of Equation 3.13 by taking the second-term as an approximation to Akaike's Information Criterion, that is,

$$AIC = \log L(\mathbf{z}) - \Delta \tag{3.14}$$

where the model whose AIC value is largest will be chosen to represent the sequence $\mathbf{z}$. We define m as the number of segments in the segmentation defined by $\mathbf{z}$ and Δ as the model dimension. Note that Equation 3.14 was multipled by -2 in the original paper presented by Akaike and considered as Akaike's information criterion due to "historical reasons" (Burnham and Anderson 2004). In our case, $\mathbf{z}$ can be described by a list of pairs of parameters (i.e., segment length, segment class) for m segments, so we write the model dimension as $\Delta = 2m$. Hence, AIC can be rewritten from Equation 3.14 as

$$AIC = \log L(\mathbf{z}) - 2m. \tag{3.15}$$

3.3.2 Bayesian Information Criterion

Akaike presented the AIC as an extension of the maximum likelihood principle. Schwarz took a similar approach and considered the problem in terms of Bayesian statistics (Schwarz 1978). In contrast to Akaike's information criterion, the Bayesian information criterion (BIC) suggests that the model dimension should be multiplied by $\frac{1}{2}\log n$. Hence, the BIC can be written as

$$BIC = \log L(\mathbf{z}) - \frac{1}{2}\Delta \log n. \tag{3.16}$$

The segmentation $\mathbf{z}$ which maximizes the BIC of Equation 3.16 is chosen as the optimal one. As before, we consider the length and the segment class as our parameters for m segments, so we write the model dimension as $\Delta = 2m$. Thus, we use the

model dimension to rewrite the BIC as

$$BIC = \log L(\mathbf{z}) - m \log n. \tag{3.17}$$

3.3.3 Minimum Description Length

Another possible approach to measuring complexity bias is through the minimum description length (MDL) method. Rissanen (1983) presented this concept by attempting to find the estimate that minimizes the total number of binary digits required to rewrite the observed data, where each observation consists of a precision value. For a fixed segmentation $\mathbf{z}$, an optimal encoding uses $-\log_2 p_{z_i}(x_i)$ bits on average to encode the character x_i in every position i. Furthermore, $\log_2 n$ bits can be used to encode the length of one segment, and $\log_2 k$ bits to specify its class. Hence, the total code length can be written as

$$\Omega = \sum_{i=1}^{n}(-\log_2 p_{z_i}(x_i)) + m(\log_2 n + \log_2 k) \tag{3.18}$$

where m is the number of segments in the segmentation defined by z. Equation 3.18 can rewritten as

$$\Omega = -\frac{\sum_{i=1}^{n} \log p_{z_i}(x_i) - m(\log n + \log k)}{\log 2}. \tag{3.19}$$

Referring to Equation 3.19, the minimum description length concept would maximize the numerator. Notice that by doing this, the first term corresponds to the log-likelihood function and the second term can be interpreted as a penalty on model complexity:

$$MDL = m(\log n + \log k) \tag{3.20}$$

Chapter 4

Algorithmic Problems In Statistical Models

4.1 Forward-Backward Algorithm

The probability of the observed sequence $\mathbf{x}$ can be calculated by finding the sum of the joint probability over all possible hidden state sequences $\mathbf{z}$. However, it is not computationally feasible since its time complexity is $O(nk^n)$ for a hidden Markov model of k states and sequence length n. To solve this, an alternate and more efficient method is through the forward-backward procedure.

Let the forward variable $\alpha_i(j)$, i.e., the probability of the partial observation sequence and state $j \in \{0, k-1\}$ up to index i, be

$$\alpha_i(j) \quad = \quad P\{x_1 x_2 \ldots x_i, z_i = j\}.$$

Then, $\alpha_i(j)$ can be solved into three steps, that is

1. Initialization:

$$\alpha_1(j) = \pi_j p_j(x_1), \qquad 0 \leq j \leq k-1.$$

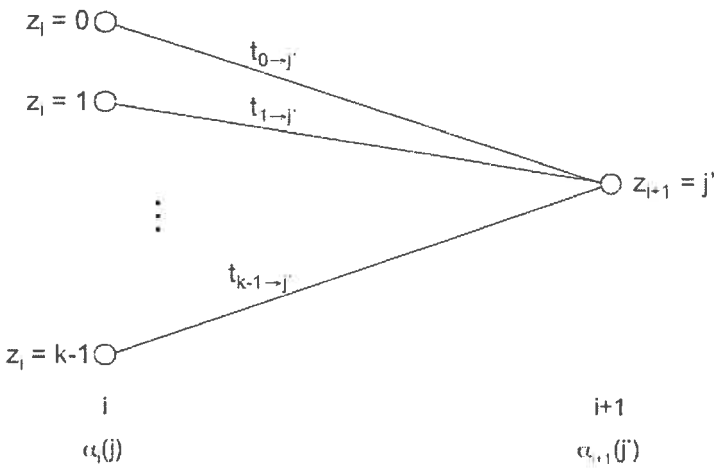


Figure 4.1. Illustration of operations required for computing forward variable $\alpha_{i+1}(j')$.

2. Induction:

$$\alpha_{i+1}(j') = \left[\sum_{j=0}^{k-1} \alpha_i(j) t_{j\to j'}\right] p_{j'}(x_{i+1}), \qquad 0 \le i \le n-1$$
$$0 \le j' \le k-1.$$

3. Termination:

$$P\{\mathbf{X} = \mathbf{x}\} = \sum_{j=0}^{k-1} \alpha_n(j).$$

The induction step can be illustrated as shown in Figure 4.1, where it demonstrates how state j' can be reached at index $i+1$ from all possible states $z_i \in \{0, k-1\}$, where the transition probability $t_{j\to j'}$, observation x_{i+1} in state j', and partial observation sequence $\alpha_i(j)$ from all k states are taken into consideration. After the termination step, it can be seen that the desired calculation of $P\{\mathbf{X} = \mathbf{x}\}$ is obtained as the sum

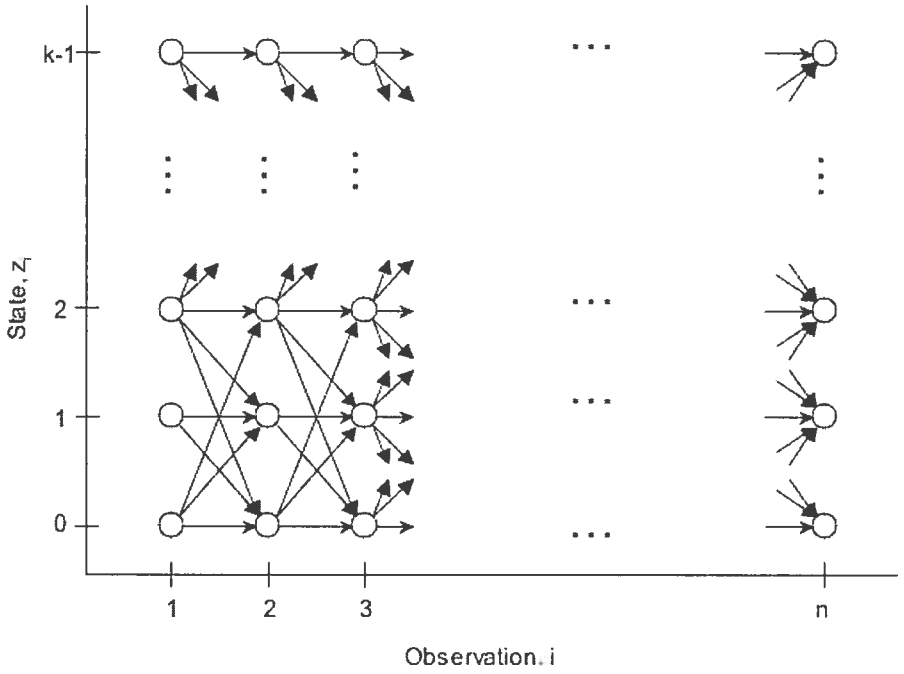


Figure 4.2. Illustration of computing $\alpha_i(j)$ in terms of observations x_i and states z_i.

of the terminal forward variables $\alpha_n(j)$.

Figure 4.2 illustrates the calculations involved in the forward procedure, where each state at index $i+1$ considers all possible states z_i.

Similarly, let the backward variable $\beta_i(j)$, i.e., the probability of the partial observation sequence from $i+1$ to the end, given state j at index i, be

$$\beta_i(j) = P\{x_{i+1}x_{i+2}\ldots x_n | z_i = j\}.$$

Then, $\beta_i(j)$ can be solved into three steps, that is

1. Initialization:

$$\beta_n(j) = 1, \qquad 0 \leq j \leq k-1.$$

2. Induction:

$$\beta_i(j) = \sum_{j'=0}^{k-1} t_{j \to j'} p_{j'}(x_{i+1}) \beta_{i+1}(j'), \qquad i = n-1, n-2, \ldots, 1,$$
$$0 \leq j \leq k-1.$$

3. Termination:

$$P\{\mathbf{X} = \mathbf{x}\} = \sum_{j=0}^{k-1} \pi_j \beta_1(j).$$

Like the induction step in the forward procedure, Figure 4.3 demonstrates how state j takes all possible states $z_{i+1} \in \{0, k-1\}$ into consideration, factoring in the transition probability $t_{j \to j'}$, observation x_{i+1} in state j', and remaining partial observation sequence $\beta_{i+1}(j')$. After the termination step, $P\{\mathbf{X} = \mathbf{x}\}$ is obtained as the sum of the terminal backward variables $\beta_1(j)$.

4.2 Viterbi Algorithm

For hidden Markov models, the optimal state sequence associated with the given observation sequence can found by implementing the Viterbi algorithm through dynamic programming, where one would find a single best state sequence while taking several possible optimality criteria into consideration.

Let $\mathbf{z}^* = z_1^* z_2^* \ldots z_n^*$ represent the best state sequence that is to be determined for a given observation $\mathbf{x} = x_1 x_2 \ldots x_n$. Suppose that $\delta_1(j)$ represent the initial best

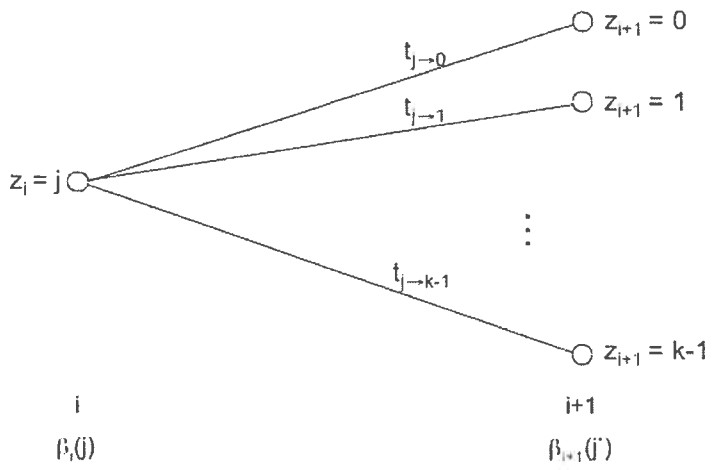


Figure 4.3. Illustration of operations required for computing backward variable $\beta_i(j)$.

score, i.e.

$$\delta_1(j) = \pi_j.$$

Furthermore, let $\delta_i(j)$ be the best score along a single path at index i up to state $j \in \{0, k-1\}$, i.e.,

$$\begin{aligned} \delta_i(j) &= \max_{z_1 z_2 \ldots z_{i-1}} P\{z_1 z_2 \ldots z_i = j, x_1 x_2 \ldots x_n\} \\ &= \left[\max_{0 \le j' \le k-1} \delta_{i-1}(j') t_{j'-j}\right] p_j(x_i). \end{aligned} \tag{4.1}$$

Using Equation 4.1, we can inductively find $\delta_{i+1}(j')$ as

$$\delta_{i+1}(j') = \left[\max_{0 \le j \le k-1} \delta_i(j) t_{j-j'}\right] p_{j'}(x_{i+1}). \tag{4.2}$$

where $j' \in \{0, k-1\}$. Then, the best sequence of hidden variables $\mathbf{z}$ when given the observed sequence $\mathbf{x}$ can be represented as

$$\begin{aligned} \mathbf{z}^* &= \arg\max_{\mathbf{z}} P\{\mathbf{z}|\mathbf{x}\} \\ &= \arg\max_{0\le j\le k-1} [\delta_n(j)]. \end{aligned} \tag{4.3}$$

Let $\psi_i(j')$ be an array that keeps track of the argument used to maximize Equation 4.2 in order to calculate the state sequence. Then, the Viterbi procedure can be presented as:

1. Initialization:

$$\begin{aligned} \delta_1(j) &= \pi_j p_j(x_1), \qquad 0 \le j \le k-1 \\ \psi_1(j) &= 0. \end{aligned}$$

2. Recursion:

$$\begin{aligned} \delta_i(j') &= \max_{0\le j\le k-1} [\delta_{i-1}(j) t_{j\to j'}]\, p_{j'}(x_i), \qquad 2 \le i \le n \\ &\qquad\qquad 0 \le j' \le k-1 \\ \psi_i(j) &= \arg\max_{0\le j\le k-1} [\delta_{i-1}(j) t_{j\to j'}] \qquad 2 \le i \le n \\ &\qquad\qquad 0 \le j' \le k-1. \end{aligned}$$

3. Termination:

$$\begin{aligned} P^* &= \max_{0\le j\le k-1} [\delta_n(j)] \\ z_n^* &= \arg\max_{0\le j\le k-1} [\delta_n(j)]. \end{aligned}$$

4. Backtracking:

$$z_i^* = \psi_{i+1}(z_{i+1}^*), \qquad i = n-1, n-2, \ldots, 1.$$

4.3 Penalty-Based Best Segmentation

In this section we consider the case of partitioning the DNA sequence using the penalty-based best segmentation model. Section 4.3.1 provides a description of how the penalty-based best segmentation model can be implemented using dynamic programming. Section 4.3.2 presents the implemented algorithms, without and with traceback, as well as how the DNA sequence can be iteratively segmented by using maximum-likelihood estimation.

Let $\mathbf{x} = x_1 \ldots x_n$ be a sequence of characters over the alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_r\}$ (e.g., a DNA sequence with $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$). Let $\omega_j : \Sigma \mapsto \mathbb{R}$ represent a scoring function for the letters in class $j \in \{0, \ldots, k-1\}$. A segmentation of $\mathbf{x}$ is defined by the segmentation indicators $\mathbf{z} = z_1 \ldots z_n$ where $z_i \in \{0, \ldots, k-1\}$. The score of such a segmentation is $\omega(\mathbf{z}) = \sum_{i=1}^{n} \omega_{z_i}(x_i)$. Given ω_j and $\mathbf{x}$, our main objective is to find $\mathbf{z}$ that maximizes $\omega(\mathbf{z})$ or its penalized form

$$\omega(\mathbf{z}) - \alpha \sum_{i=2}^{n} \mathbb{I}_{z_i \neq z_{i-1}}.$$

where α is a segmentation complexity penalty. This general scoring framework includes likelihood maximization as a special case: set

$$\omega_j(x_i) = \log \frac{p_j(x_i)}{p_0(x_i)} \tag{4.4}$$

from Equation 3.3. The various complexity penalties of Equation 3.3 are reflected in the choice of α. In a more compact form, we describe a segmentation by the set of its

segments. A *segment* $S = [a, b] \mapsto j$ is a maximal contiguous subsequence of indices $\{a, a+1, \ldots, b\}$ such that $z_a = z_{a+1} = \ldots = z_b = j$. A *segmentation set* Φ is defined as the partition of the indices $[1, n]$ into a set of segments, that is, $\Phi = \{\phi_1, \ldots, \phi_m\}$ where each ϕ_j is a segment (and consecutive segments belong to different classes).

4.3.1 Description

The penalized segmentation score takes the segmentation penalty α into consideration for each segment transition found in Φ and can be written as $V = V(\Phi) - \alpha \cdot (|\Phi| - 1)$, where $|\Phi| > 0$.

Lemma 1 *Let $V(i, j)$ be defined as the best segmentation score of $[1 \ldots i]$ that ends with class j. If $i > 1$, then*

$$V(i, j) = \omega_j(x_i) + \max_{c \in [0.k-1]} \{V(i-1, c) - \alpha \cdot \mathbb{I}_{c \neq j}\} \tag{4.5}$$

where $V(1, j) = \omega_j(x_1)$ and

$$\mathbb{I}_{c \neq j} = \begin{cases} 1 & \text{if } c \neq j \\ 0 & \text{otherwise} \end{cases}$$

Proof Consider the segmentation set of $[1, i]$ by extending the last interval of a segmentation set associated with $V(i-1, j)$, and by adding $[i, i]$ belonging to class j to a segmentation set associated with $V(i-1, c)$ where $c \neq j$. Then,

$$V(i, j) \geq \omega_j(x_i) + \max_{c \in [0.k-1]} \{V(i-1, c) - \alpha, V(i-1, j)\}$$

where $(\omega_j(x_i) + V(i-1, c) - \alpha)$ and $(\omega_j(x_i) + V(i-1, j))$ are the respective penalized scores. However, there is no inequality since i can be removed from the segmentation set of $[1, i]$ to obtain the segmentation set of $[1, i-1]$. ∎

Lemma 2 *$V(i,j)$ takes two different cases into consideration: $c = j$ and $c \neq j$*

$$V(i,j) = \omega_j(x_i) + \max\left\{V(i-1,j), \max_{c\in[0.k-1]}\{V(i-1,c) - \alpha\}\right\}$$

Proof Let β represent the second term of the equation, that is

$$\beta = \max\left\{V(i-1,j), \max_{c\in[0.k-1]}\{V(i-1,c) - \alpha\}\right\}$$

Since it is trivial that $V(i-1,j) > V(i-1,j) - \alpha$, β can be rewritten such that

$$\beta = \max\left\{V(i-1,j), \max_{c\in[0,k-1]\setminus j}\{V(i-1,c) - \alpha\}\right\}$$

Hence, this lemma's equation is the same as Equation 4.5, that is,

$$\begin{aligned} V(i,j) &= \omega_j(x_i) + \max\left\{V(i-1,j), \max_{c\in[0.k-1]\setminus j}\{V(i-1,c) - \alpha\}\right\} \\ &= \omega_j(x_i) + \max_{c\in[0.k-1]}\{V(i-1,c) - \alpha\cdot\mathbb{I}_{c\neq j}\} \end{aligned}$$

∎

Lemma 1 implies a dynamic programming algorithm that executes in $O(nk^2)$ time. The implementation of Lemma 2, however, implies that the maximum value of $V(i-1,c)$ can be pre-calculated and would reduce the time complexity to $O(nk)$ time.

4.3.2 Algorithms

Without Traceback Implementation

Equation 4.5 demonstrates the accumulated score calculation at nucleotide x_i by using the best possible prefix score at the previous nucleotide x_{i-1} via dynamic programming. This idea is employed in the algorithm presented in Figure 4.4, where the

scores at the first nucleotide x_1 are initialized as written in the the loop body 1-3. The loop body 4-13 processes the scores of the remaining nucleotides x_i for each class, where Line 5 helps reduce the number of iterations of determining the maximum value of the scores from the previous nucleotide. Lines 8 and 10 are implementations of Lemma 2 for each class.

Lemma 3 *The algorithm presented in Figure 4.4 finds an optimal segmentation into k classes for a DNA sequence of length n in $O(nk)$ time.*

Proof The initialization of the scores for $j \in [0, k-1]$ at x_1 is performed in the first loop body 1-3 in $O(k)$ time. The second loop body 4-13 is executed $n-1$ times, where the maximum of $V(i-1, c)$ for $c \in [0, k-1]$ in Line 5 and the inner loop body 6-12 are determined in $O(k)$ time for each iteration. Therefore, the time complexity can be simplified to yield $O(nk)$. ∎

```
   Input  : Sequence X, segmentation penalty α
   Output: Best penalized segmentation Φ
 1 for j ← 0 to k − 1 do
 2     V(1,j) ← ω_j(x_1):
 3 end
 4 for i ← 2 to n do
 5     V_max ← max_{c∈[0,k−1]} V(i-1,c);
 6     for j ← 0 to k − 1 do
 7         if V(i-1,j) ≥ V_max − α then
 8             V(i,j) ← ω_j(x_i) + V(i-1,j);
 9         else
10             V(i,j) ← ω_j(x_i) + V_max − α;
11         end
12     end
13 end
```

Figure 4.4. Penalty-based best segmentation algorithm.

With Traceback Implementation

Let $P(i,j)$ denote an element of the traceback array containing the class number of the previous nucleotide x_{i-1} referred to when calculating the accumulated penalized segmentation score $V(i,j)$ via Equation 4.5. For example, if $V(i,j) = \omega_j(x_i) + V(i-1,c) - \alpha$ and $c \neq j$, then $P(i,j) = c$ (otherwise, $P(i,j) = j$). The algorithm presented in Figure 4.5 is an adapted version of algorithm denoted in Figure 4.4 which incorporates the traceback array feature, where $P(i,j)$ is assigned in Lines 10 and 13 for each $1 < i \leq n$ and class c. Note that the traceback array elements at the beginning of the sequence for each class c, that is, $P(1,c)$ are not assigned. Line 17 assembles the segmentation set Φ through the traceback algorithm.

Input : Sequence $\mathbf{X}$, segmentation penalty α
Output: Best penalized segmentation Φ

```
1  for j ← 0 to k − 1 do
2      V(1,j) ← ω_j(x_1);
3  end
4  for i ← 2 to n do
5      V_max ← max_{c∈[0,k−1]} V(i-1,c);
6      C_max ← arg max_{c∈[0,k−1]} V(i-1,c);
7      for j ← 0 to k − 1 do
8          if V(i-1,j) ≥ V_max − α then
9              V(i,j) ← ω_j(x_i) + V(i-1,j);
10             P(i,j) ← j;
11         else
12             V(i,j) ← ω_j(x_i) + V_max − α;
13             P(i,j) ← C_max;
14         end
15     end
16 end
17 Φ ← Traceback(V, P);
```

Figure 4.5. Penalty-based best segmentation with traceback array algorithm.

The traceback algorithm is implemented as shown in Figure 4.6, where it computes the segments obtained from the penalty-based best segmentation algorithm. We use $\mathbf{z} = z_1 \dots z_n$ to store the class numbers of the sequence. As well, let δ be the current segment class number, δ' the segment class number read from the traceback array, i_{start} the start of a segment, i_{end} the end of a segment, and ρ the iterator for the list of segments. This algorithm is initialized by determining which class contains the maximum accumulated penalized score for the sequence as illustrated in Line 4. In the loop body 6-10, δ' is read from the traceback array (Line 7) and is inserted into z_{i-1} (Line 9). The segmentation set Φ is assembled in the loop body 11-18, where the segment $\phi_\rho = \{[i_{start}, i_{end}] \mapsto z_{i-1}\}$ is inserted into Φ if z_{i-1} is not equal to z_i (Line 12). As well, i_{end} and i_{start} are updated as denoted in Lines 13 and 15, respectively. The last segment inserted to the list is represented in Line 20.

Maximum-Likelihood Estimation of Segments

Maximum-likelihood estimation can be used to approximately determine the segments of a given sequence when provided probability values. This can be implemented in conjunction with the penalty-based best segmentation algorithm as presented in Figure 4.7.

The algorithm will be repeated executed in the loop body 1-5 until the log-likelihood ratio converges. For each iteration, a new score is calculated based on the log-likelihood ratio and a new segmentation is computed via the penalty-based best segmentation algorithm as presented in Figure 4.5. Furthermore, new probability values are calculated using Laplace pseudo-counters as implemented in the algorithm denoted in Figure 5.1.

Input : Segmentation score array V, traceback array P, penalty α
Output: Assembled segmentation Φ

$i_{start} \leftarrow 1$;
$i_{end} \leftarrow 1$;
$\rho \leftarrow 1$;
$\delta \leftarrow \arg\max_{c \in [0..k-1]} V(n,c)$;
$z_n \leftarrow \delta$;
for $i \leftarrow n$ **downto** 2 **do**
    $\delta' \leftarrow P(i,\delta)$;
    $\delta \leftarrow \delta'$;
    $z_{i-1} \leftarrow \delta$;
end
for $i \leftarrow 2$ **to** n **do**
    if $z_{i-1} \neq z_i$ **then**
        $i_{end} \leftarrow i - 1$;
        $\phi_\rho \leftarrow \{[i_{start}..i_{end}] \mapsto z_{i-1}\}$;
        $i_{start} \leftarrow i$;
        $\rho \leftarrow \rho + 1$;
    end
end
$i_{end} \leftarrow n$;
$\phi_\rho \leftarrow \{[i_{start}..i_{end}] \mapsto z_n\}$;

Figure 4.6. Traceback algorithm.

Input : Sequence $\mathbf{X}$, segmentation penalty α
Output: Maximum-likelihood segmentation via penalty-based best segmentation algorithm Φ

repeat
    $\omega_j(x) \leftarrow \log \frac{p_j(x)}{p_0(x)}$;
    $\Phi \leftarrow$ `BestSegmentation`$(\mathbf{X}, \alpha)$;
    $p \leftarrow$ `ProbabilityCalculation`(Φ);
until *convergence* ;

Figure 4.7. Maximum-likelihood estimation of segments via penalty-based best segmentation algorithm.

4.4 Penalty-Based Best Segmentation with Minimum Segment Lengths

In this section we discuss the case of partitioning the DNA sequence using the penalty-based best segmentation with minimum segment lengths model. Section 4.4.1 provides a description of how the penalty-based best segmentation with minimum segment lengths model can be implemented using dynamic programming. Section 4.4.2 presents the implemented algorithms without and with traceback, as well as an algorithm in which the DNA sequence can be iteratively segmented by using maximum-likelihood estimation.

4.4.1 Description

While calculating via the penalty-based best segmentation equation as presented in Equation 4.5 yields a pattern of segments in the sequence, it is possible that overfitting data (i.e. $\phi_l = [i,i] \mapsto j : \omega_{i,j} > 0$ segments) can occur and consequently render the data biologically meaningless. Hence, Equation 4.5 can be modified such that it takes minimum segment lengths into consideration.

Let m_j be defined as the minimum segment length for class j. Thus, for $m \in [1, m_j]$, and $i \in [m, n]$, let $\Phi^j_{i,m}$ represent the segmentation sets of $[1, i]$ that maximize $V(\Phi)$ while satisfying the requirements for all segment lengths except for the last one of class j whose length is at least m.

Lemma 4 *Let $V_{short}(i,j) = V(\Phi^j_{i,1})$ and $V_{long}(i,j) = V(\Phi^j_{i,m_j})$. Therefore, the following recursions represent the calculation of the weights of these segmentation sets, that is*

$$\begin{aligned} V_{short}(i,j) &= \omega_j(x_i) + \\ &\quad \max\left\{ V_{short}(i-1,j), \max_{c\in[0,k-1]} \{V_{long}(i-1,c) - \alpha \cdot \mathbb{I}_{c\neq j}\} \right\} \end{aligned} \quad (4.6)$$

$$V_{long}(i,j) = V_{short}(i-m_j+1,j) + \sum_{l=i-m_j+2}^{i} \omega_j(x_l) \tag{4.7}$$

where $V_{short}(1,j) = \omega_j(x_1)$ *and* $c \in [0,k-1] \setminus j$.

Proof For Equation 4.6, let ϕ_{short} be some segment of class j at nucleotide x_i, i.e. $\phi_{short} = [i,i] \mapsto j$. If the last segment of $\Phi^j_{i,1}$ includes nucleotide x_{i-1} (i.e. nucleotide x_i and x_{i-1} are of the same nucleotide), then $\Phi^j_{i,1}$ is obtained by extending the last segment of $\Phi^j_{i-1,j}$. Otherwise, $\Phi^j_{i,1} = \Phi^c_{i-1,m_c} \cup \{\phi_{short}\}$.

For Equation 4.7, let ϕ_{long} be some segment of class j from nucleotides x_{i-m_j+2} to x_i, i.e. $\phi_{long} = [i-m_j+2,i] \mapsto j$. Φ^j_{i,m_j} is obtained by extending the last segment of $\Phi^j_{i-m_j+1,1}$ with segment ϕ_{long}. ∎

Lemma 5 V_{short} *takes two different cases into consideration:* $c = j$ *and* $c \neq j$

$$V_{short}(i,j) = \omega_j(x_i) + \max\left\{V_{short}(i-1,j), \max_{c\in[0,k-1]}\{V_{long}(i-1,c) - \alpha\}\right\}$$

Proof Let β represent the second term of the equation, that is

$$\beta = \max\left\{V_{short}(i-1,j), \max_{c\in[0,k-1]}\{V_{long}(i-1,c) - \alpha\}\right\}$$

Let m_{long} be the required minimum segment length associated with $V_{long}(i-1,j)-\alpha$. Likewise, let $m_{short} = 1$ be the required minimum segment length associated with $V_{short}(i-1,j)$. Since $m_{long} \geq m_{short}$, then

$$V_{short}(i-1,j) > V_{long}(i-1,j) - \alpha.$$

Hence, β can be rewritten such that

$$\beta = \max\left\{V_{short}(i-1,j), \max_{c\in[0,k-1]\setminus j}\{V_{long}(i-1,c) - \alpha\}\right\}$$

Therefore, this lemma's equation is the same as Equation 4.6, that is,

$$\begin{aligned} V_{short}(i,j) &= \omega_j(x_i) + \max\left\{V_{short}(i-1,j), \max_{c\in[0..k-1]\setminus j}\{V_{long}(i-1,c)-\alpha\}\right\} \\ &= \omega_j(x_i) + \\ &\quad \max\left\{V_{short}(i-1,j), \max_{c\in[0..k-1]}\{V_{long}(i-1,c)-\alpha\cdot\mathbb{I}_{c\neq j}\}\right\} \end{aligned}$$

∎

Lemma 4 implies a dynamic programming of that computes an optimal segmentation set that is subject to the given minimum segment lengths in $O(nk^2)$ time. However, like Lemma 2, applying Lemma 5 would allow the pre-calculation of the maximum value of $V_{long}(i-1,c)$, thereby reducing the time complexity to $O(nk)$.

4.4.2 Algorithms

Without Traceback Implementation

Referring to Equations 4.6 and 4.7, the occurrence of overfitting data can be eliminated by specifying the minimum segment lengths of segments. Let m be an array containing minimum segment lengths for each class j. The algorithm presented in Figure 4.8 illustrates this computation where both V_{short} and V_{long} scores at the first nucleotide x_i are initialized for each class in the loop body 1-10. Loop body 11-29 processes the scores of the remaining nucleotides x_i, where the largest value of $V_{long}(i-1,c)$ is determined in Line 12 in order to reduce the number of required iterations. The inner loop body 13-28 is an implementation of Equations 4.6 and 4.7.

Lemma 6 *The algorithm presented in Figure 4.8 finds an optimal segmentation into k classes for a DNA sequence of length n in $O(nk)$ time.*

Proof The initialization of the scores for $j \in [0..k-1]$ at x_1 is performed in the first loop body 1-10 in $O(k)$ time. The second loop body 11-29 is executed $n-1$

times, where the maximum calculations in Line 12 and the inner loop body 13-28 are determined in $O(k)$ time for each iteration. Therefore, the time complexity can be simplified to yield $O(nk)$. ∎

Input : Sequence $\mathbf{X}$, segmentation penalty α, minimum segment lengths m
Output: Minimum length segmentation Φ

```
1  for j ← 0 to k − 1 do
2      V_short(1,j) ← ω_j(x_i);
3      if m_j > 1 then
4          V_long(1,j) ← −∞;
5          s_j ← ω_j(x_1);
6      else
7          V_long(1,j) ← V_short(1,j);
8          s_j ← 0;
9      end
10 end
11 for i ← 2 to n do
12     V_max ← max_{c∈[0,k−1]} V_long(i-1,c);
13     for j ← 0 to k − 1 do
14         s_j ← s_j + ω_j(x_i);
15         if m_j > 1 then
16             s_j ← s_j − ω_j(x_{i−m_j+1});
17         end
18         if V_short(i-1,j) ≥ V_max − α then
19             V_short(i,j) ← ω_j(x_i) + V_short(i-1,j);
20         else
21             V_short(i,j) ← ω_j(x_i) + V_max − α;
22         end
23         if i ≥ m_j then
24             V_long(i,j) ← V_short(i − m_j + 1,j) + s_j;
25         else
26             V_long(i,j) ← −∞;
27         end
28     end
29 end
```

Figure 4.8. Penalty-based minimum length segmentation algorithm.

With Traceback Implementation

The algorithm presented in Figure 4.9 is an adapted version of the algorithm denoted in Figure 4.8 which incorporates the traceback array feature. Similar to its penalty-based best segmentation traceback counterpart, the traceback array elements $P(i, j)$ is assigned for each nucleotide x_i and class c in Lines 21 and 24. Again, the traceback array elements at the first nucleotide x_1 for each class c, that is $P(1, c)$ are not assigned. Line 33 assembles the segmentation set Φ through the traceback with minimum segment lengths algorithm.

The traceback with minimum segment lengths algorithm is presented in Figure 4.10, where it computes the segments while taking the specified minimum segment lengths into consideration. We use $\mathbf{z} = z_1 \dots z_n$ to store the class numbers of the sequence. Furthermore, let δ be the current segment class number, δ' the segment class number read from the traceback array, $m_{counter}$ the minimum length counter, i_{start} the start of a segment, i_{end} the end of a segment, and ρ the iterator for the list of segments. This algorithm is initialized by determining which class contains the maximum accumulated penalized score for the sequence (Line 3) and setting the $m_{counter}$ based on this class (Line 4). In the loop body 7-18, $m_{counter}$ decrements whenever it is not 1 (Line 9), otherwise δ and $m_{counter}$ changes whenever δ is not equal to δ' (Line 12). In either case, z_{i-1} stores the current class number δ. The segmentation set Φ is assembled in the loop body 19-26, where the segment $\phi_\rho = \{[i_{start}, i_{end}] \mapsto z_{i-1}\}$ is inserted into Φ if z_{i-1} is not equal to z_i (Line 20). As well, i_{end} and i_{start} are updated as denoted in Lines 21 and 23, respectively. The last segment inserted to the list is represented in Line 28.

Maximum-Likelihood Estimation of Segments

Like its penalty-based best segmentation algorithm counterpart, maximum-likelihood estimation can be used with the penalty-based best segmentation with minimum

Input : Sequence $\mathbf{X}$, segmentation penalty α, minimum segment lengths m
Output: Minimum length segmentation Φ

```
1  for j ← 0 to k − 1 do
2      V_short(1,j) ← ω_j(x_i);
3      if m_j > 1 then
4          V_long(1,j) ← −∞;
5          s_j ← ω_j(x_1);
6      else
7          V_long(1,j) ← V_short(1,j);
8          s_j ← 0;
9      end
10 end
11 for i ← 2 to n do
12     V_max ← max_{c∈[0,k−1]} V_long(i-1,c);
13     C_max ← arg max_{c∈[0,k−1]} V_long(i-1,c);
14     for j ← 0 to k − 1 do
15         s_j ← s_j + ω_j(x_i);
16         if m_j > 1 then
17             s_j ← s_j − ω_j(x_{i−m_j+1});
18         end
19         if V_short(i-1,j) ≥ V_max − α then
20             V_short(i,j) ← ω_j(x_i) + V_short(i-1,j);
21             P(i,j) ← j;
22         else
23             V_short(i,j) ← ω_j(x_i) + V_max − α;
24             P(i,j) ← C_max;
25         end
26         if i ≥ m_j then
27             V_long(i,j) ← V_short(i − m_j + 1,j) + s_j;
28         else
29             V_long(i,j) ← −∞;
30         end
31     end
32 end
33 Φ ← MinimumLengthTraceback(V, P, m);
```

Figure 4.9. Penalty-based minimum length segmentation with traceback array algorithm.

Input : Segmentation score array V, traceback array P, minimum segment lengths m
Output: Assembled segmentation Φ

$i_{start} \leftarrow 1$;
$i_{end} \leftarrow 1$;
$\delta \leftarrow \arg\max_{c \in [0..k-1]} V(n, c)$;
$m_{counter} \leftarrow m_{\delta}$;
$z_n \leftarrow \delta$;
$\rho \leftarrow 1$;
for $i \leftarrow n$ **downto** 2 **do**
    if $m_{counter} > 1$ **then**
        $m_{counter} \leftarrow m_{counter} - 1$;
    else
        $\delta' \leftarrow P(i, \delta)$;
        if $\delta \neq \delta'$ **then**
            $\delta \leftarrow \delta'$;
            $m_{counter} \leftarrow m_{\delta}$;
        end
    end
    $z_{i-1} \leftarrow \delta$;
end
for $i \leftarrow 2$ **to** n **do**
    if $z_{i-1} \neq z_i$ **then**
        $i_{end} \leftarrow i - 1$;
        $\phi_{\rho} \leftarrow \{[i_{start}, i_{end}] \mapsto z_{i-1}\}$;
        $i_{start} \leftarrow i$;
        $\rho \leftarrow \rho + 1$;
    end
end
$i_{end} \leftarrow n$;
$\phi_{\rho} \leftarrow \{[i_{start}, i_{end}] \mapsto z_n\}$;

Figure 4.10. Traceback with minimum segment lengths algorithm.

segment lengths algorithm to approximately find the segments of a given sequence when provided probability values, as implemented in the algorithm as shown in Figure 4.11.

Input : Sequence $\mathbf{X}$, segmentation penalty α, minimum segment lengths m
Output: Maximum-likelihood segmentation via penalty-based best segmentation algorithm Φ

1 **repeat**
2 $\quad \omega_j(x) \leftarrow \log \frac{p_j(x)}{p_0(x)}$;
3 $\quad \Phi \leftarrow$ `MinimumLengthSegmentation`$(\mathbf{X}, \alpha, m)$;
4 $\quad p \leftarrow$ `ProbabilityCalculation`(Φ);
5 **until** *convergence* ;

Figure 4.11. Maximum-likelihood estimation of segments via penalty-based minimum length segmentation algorithm.

The algorithm will be repeatedly executed in the loop body 1-5 until the log-likelihood ratio converges. For each iteration, a new score is calculated based on the log-likelihood ratio and a new segmentation is computed via the penalty-based best segmentation with minimum segment lengths algorithm as described in the algorithm presented in Figure 4.9. Furthermore, new probability values are calculated using Laplace pseudo-counters as implemented in the algorithm presented in Figure 5.1.

Chapter 5

Experimental Results

GC-content is defined as the measurement of the relative frequency of G (guanine) and C (cytosine) found in a region. When applied to DNA segmentation, this has been used to determine segments whose contents are rich in guanine and cytosine.

The actual genome sequences of the organisms used in this research can be downloaded from the National Center for Biotechnology Information (NCBI) website (i.e. *http://www.ncbi.nlm.nih.gov*).

The European Molecular Biology Open Software Suite (EMBOSS) is a collection of software analysis programs designed to meet the needs of the molecular biology user community (Rice et al. 2000). Among the numerous functionalities that EMBOSS can perform include sequence alignment, nucleotide sequence pattern analysis, and GC-content sequence analysis. We compared our segmentation results with those obtained from the GC-content analysis tool found in the latest version of EMBOSS, which can be downloaded from its homepage (i.e. *http://emboss.sourceforge.net/*).

Before we discuss the experimental results, the following presents other implementations that were included to complement our sequence segmentation algorithms.

Model Parameter Estimation

In Chapter 3, we presented the log-likelihood function for our statistical models as denoted in Equation 3.3, where the first term denotes the null hypothesis that the segments is of class 0 and the second term is the log-likelihood ratio defined by the observation sequence **z**. In order to avoid encountering probability values of 0 in the denominator of the log-likelihood ratio, Laplace pseudo-counters can be used as denoted in Equation 5.1

$$p_i = \frac{n_i + 1}{n + r} \tag{5.1}$$

where $p = \{p_1, p_2, \ldots, p_r\}$ is a collection of probability values for alphabet Σ and $n_1, n_2, \ldots, n_r$ represent the counters of characters $1, \ldots, r$ in positions for a given class in a given segmentation.

Laplace pseudo-counters were implemented as presented in Figure 5.1. For each class j, let the total number of nucleotides be represented as n_{total} and the counters of nucleotide alphabet $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ be denoted as $n_{\sigma_1}, n_{\sigma_2}, \ldots, n_{\sigma_r}$. The algorithm iteratively counts the number of each nucleotide alphabet and total number of nucleotides found in the current class as expressed in the loop body 3-8. Ultimately, the new probability values are calculated in Line 10.

Data Compression

In Chapter 4, we presented the penalty-based best segmentation algorithms in which dynamic programming was implemented by applying the traceback array. As previously mentioned, the traceback array contains class numbers at various nucleotide positions. To increase memory efficiency, we stored our traceback values by compressing them in byte (8-bit) arrays, where the value is stored in the appropriate position in the data byte.

To construct our traceback array, we needed to determine the fixed amount of

Input : Segmentation Φ
Output: New probability values p

for $j \leftarrow 0$ **to** $k-1$ **do**
    $n_{total} \leftarrow 0$;
    for $i \leftarrow 1$ **to** n **do**
        if $z_i = j$ **then**
            $n_{x_i} \leftarrow n_{x_i} + 1$;
            $n_{total} \leftarrow n_{total} + 1$;
        end
    end
    for $\sigma \leftarrow \sigma_1$ **to** σ_r **do**
        $p_\sigma \leftarrow \frac{n_\sigma + 1}{n_{total} + r}$;
    end
end

Figure 5.1. Probability calculation algorithm.

memory required to store our traceback values efficiently. For a sequence $\mathbf{X}$ and k classes, let C represent the number of classes that can be stored per byte, b the number of bits required per class, r the number of required bits to construct the traceback array, and R the number of required bytes equivalent to r. Determining C depends on k, that is,

$$C = \begin{cases} 1 & \text{if } 16 < k \leq 256 \\ 2 & \text{if } 4 < k \leq 16 \\ 4 & \text{if } 2 < k \leq 4 \\ 8 & \text{otherwise} \end{cases} \tag{5.2}$$

Because a byte consists of 8 bits, b can be calculated by dividing 8 bits by the number of classes per byte, that is,

$$b = \frac{8}{C}. \tag{5.3}$$

Furthermore, r can be determined by taking the sequence length, number of classes,

and number of bits per class into consideration, or

$$r = (|\mathbf{X}| - 1) \times k \times b. \tag{5.4}$$

We convert the number of required bits r to the number of required bytes R by using

$$R = \left\lceil \frac{r}{8} \right\rceil. \tag{5.5}$$

Suppose T denote the traceback array, t_0 the initial traceback index, t_{offset} the traceback offset value, t_{idx} the traceback index, and c the number of counted classes. Using these variables, we can determine where to add our new value in our traceback array, as presented in Figure 5.2. Line 5 uses the compress static function to store the data at the appropriate place in the traceback array. Suppose n denote the new byte value. Then, the implementation of this function is presented as shown in Figure 5.3, where we used logic operations and hexadecimal values to assign the value accordingly.

Input: Value v, nucleotide index i, class index j

1 $t_0 \leftarrow \frac{(i-2)\times k\times b}{8}$;
2 $t_{offset} \leftarrow \frac{j}{C}$;
3 $t_{idx} \leftarrow t_0 + t_{offset}$;
4 $c \leftarrow c + 1$;
5 $T(t_{idx}) \leftarrow$ `Compress`$(T(t_{idx}), v, C, c)$;

Figure 5.2. Add data to traceback array algorithm.

Similar to Figure 5.2, the appropriate value can be read from the traceback array as presented in Figure 5.4. Line 4 translates the nucleotide index input value to the appropriate index value in the traceback array. Also, line 5 calls the decompress static function which returns the byte value. This decompression function is presented in Figure 5.5.

Input : Current byte value B, value V, number of classes per byte C, number of counted classes c
Output: New compressed byte value n

```
1  switch C do
2      case 1
3          n ← V;
4      case 2
5          if c mod C = 1 then  n ← (V << 4) ∨ (B ∧ 0x0F)
6          else  n ← V ∨ (B ∧ 0xF0);
7      case 4
8          switch c mod C do
9              case 1
10                 n ← (V << 6) ∨ (B ∧ 0x3F);
11             case 2
12                 n ← (V << 4) ∨ (B ∧ 0xCF);
13             case 3
14                 n ← (V << 2) ∨ (B ∧ 0xF3);
15             case 0
16                 n ← V ∨ (B ∧ 0xFC);
17         end
18     case 8
19         switch c mod C do
20             case 1
21                 n ← (V << 7) ∨ (B ∧ 0x7F);
22             case 2
23                 n ← (V << 6) ∨ (B ∧ 0xBF);
24             case 3
25                 n ← (V << 5) ∨ (B ∧ 0xDF);
26             case 4
27                 n ← (V << 4) ∨ (B ∧ 0xEF);
28             case 5
29                 n ← (V << 3) ∨ (B ∧ 0xF7);
30             case 6
31                 n ← (V << 2) ∨ (B ∧ 0xFB);
32             case 7
33                 n ← (V << 1) ∨ (B ∧ 0xFD);
34             case 0
35                 n ← V ∨ (B ∧ 0xFE);
36         end
37 end
```

Figure 5.3. Byte compression into traceback array algorithm.

Input : Nucleotide index i, class index j
Output: Byte value from traceback array n

1 $t_0 \leftarrow \frac{(i-2)\times k\times b}{8}$;
2 $t_{offset} \leftarrow \frac{j}{C}$;
3 $t_{idx} \leftarrow t_0 + t_{offset}$;
4 $d \leftarrow (i-2) \times k \times j + 1$;
5 $n \leftarrow$ `Decompress(`$T(t_{idx})$, C, d `)`;

Figure 5.4. Read data to traceback array algorithm.

5.1 Bacteriophage Lambda

5.1.1 Description

The bacteriophage λ is a parasite of the intestinal bacterium *Escherichia coli* that is commonly used as a benchmark sequence for the comparison of segmentation algorithms (Boys and Henderson 2004). The reasoning for considering this organism is due to its experimental segmentation being based on the gradient centrifugation of its GC-content as conducted by Skalka et al. They identified six sections of differing GC-content and deduced that the lengths given for the three shorter sections are not exact, while the lengths of the three longer sections are. Furthermore, they concluded that any errors found anywhere other than the three longer sections are compensated equally in the 43%-GC and 48%-GC sections (Skalka et al. 1968). Their experiment work is quantitatively presented in Figure 5.6 and Table 5.1.

To analyze the nucleotide distribution for bacteriophage λ graphically, we used the isochore analysis application included in EMBOSS. This application operates by calculating the GC-content within a fixed-length window and incrementally shifting this window along the entire sequence. Figure 5.7 provides an illustration of the nucleotide distribution for bacteriophage λ based on GC-content using EMBOSS, where we used the default values of 1-kb and 0.1-kb as our window length and shift increment values, respectively.

Input : Current byte value B, number of classes per byte C, converted class index c
Output: Appropriate byte value n

```
1  switch C do
2      case 1
3          n ← B;
4      case 2
5          if c mod C = 1 then  n ← (B ∧ 0xF0) >> 4
6          else  n ← (B ∧ 0x0F);
7      case 4
8          switch c mod C do
9              case 1
10                 n ← (B ∧ 0xC0) >> 6;
11             case 2
12                 n ← (B ∧ 0x30) >> 4;
13             case 3
14                 n ← (B ∧ 0x0C) >> 2;
15             case 0
16                 n ← (B ∧ 0x03);
17         end
18     case 8
19         switch c mod C do
20             case 1
21                 n ← (B ∧ 0x80) >> 7;
22             case 2
23                 n ← (B ∧ 0x40) >> 6;
24             case 3
25                 n ← (B ∧ 0x20) >> 5;
26             case 4
27                 n ← (B ∧ 0x10) >> 4;
28             case 5
29                 n ← (B ∧ 0x08) >> 3;
30             case 6
31                 n ← (B ∧ 0x04) >> 2;
32             case 7
33                 n ← (B ∧ 0x02) >> 1;
34             case 0
35                 n ← (B ∧ 0x01);
36         end
37 end
```

Figure 5.5. Byte decompression from traceback array algorithm.

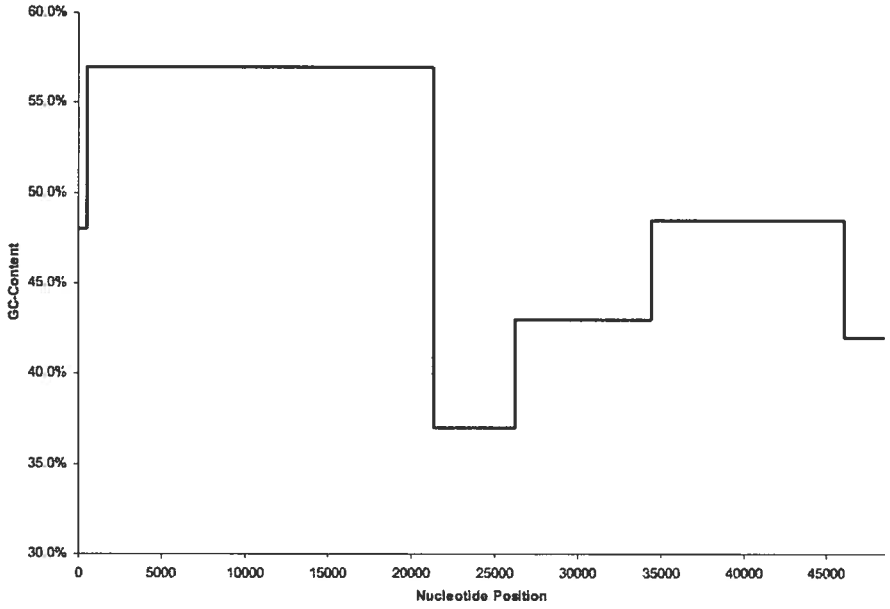


Figure 5.6. Nucleotide distribution in bacteriophage λ based on GC-content via gradient centrifugation.

Segment	Nucleotide Start	Nucleotide End	%GC
1	1	485	48
2	486	21340	57
3	21341	26191	37
4	26192	34436	43
5	34437	46077	48
6	46078	48502	42

Table 5.1. Quantitative data describing the nucleotide distribution in bacteriophage λ via density centrifugation.

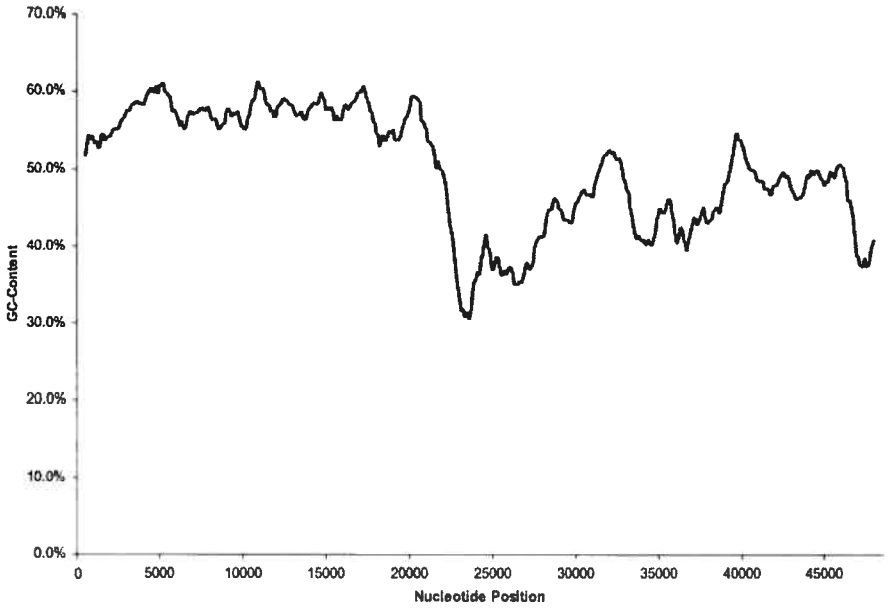


Figure 5.7. Nucleotide distribution in bacteriophage λ based on GC-content via EMBOSS.

5.1.2 Tests

We tested our algorithms on the bacteriophage λ (48,502 base pairs, GenBank accession NC_001416) using different complexity penalty values α as depicted in Table 5.2 (where k is the number of classes to be considered). We also tested the DNA segmentation on a number of classes with the probability values presented in Table 5.3 and the minimum length values listed in Table 5.4. We compared each DNA segmentation test to the obtained bacteriophage λ nucleotide distribution conducted by Skalka et al.

Test Name	α
None	0
AIC	2
BIC	$\log 48.502$
MDL	$\log 48.502 + \log k$

Table 5.2. Complexity penalty values α tested for bacteriophage λ.

Case	$p_j(x)$	
2 Classes	$p_0(S) = 0.5$ $p_1(S) = 0.52$	$p_0(W) = 0.5$ $p_1(W) = 0.48$
3 Classes	$p_0(S) = 0.5$ $p_1(S) = 0.52$ $p_2(S) = 0.7$	$p_0(W) = 0.5$ $p_1(W) = 0.48$ $p_2(W) = 0.3$
4 Classes	$p_0(S) = 0.37$ $p_1(S) = 0.42$ $p_2(S) = 0.48$ $p_3(S) = 0.57$	$p_0(W) = 0.63$ $p_1(W) = 0.58$ $p_2(W) = 0.52$ $p_2(W) = 0.43$

Table 5.3. Probability values $p_j(x)$ tested for bacteriophage λ.

Case	Class	Minimum Length
2 Classes	0	200
	1	200
3 Classes	0	200
	1	200
	2	200
4 Classes	0	200
	1	200
	2	200
	3	200

Table 5.4. Minimum length values tested for bacteriophage λ.

5.1.3 Results

Case 1: 2 Classes

For the case in which there is no complexity penalty value, we found the bacteriophage λ genome to be heavily segmented such that the segmentation consists of two types of segments, i.e., segments consisting of only C or G nucleotides and segments that do not. In addition, we observed the occurrence of overfitting data, where segments of length 1 are found. These observations are similar to those found in the AIC test, although the GC-content for the GC-poor and GC-rich segments will not necessarily be 0% and 100%, respectively.

The DNA segmentation for the BIC and MDL tests yielded four segments as shown graphically in Figure 5.8 and quantitatively in Table 5.5. As illustrated by the solid and dashed lines in both graphs, we found both experimental nucleotide distributions to be graphically similar, and not necessarily exact, to those found through gradient centrifugation. Because it was previously found that the lengths of the three longer sections are more exact for gradient centrifugation, we calculated the average GC-content within these sections on our experimental data, say, A, B, and C. For every section in both graphs, we found a resemblance between the calculated

average experimental GC-content and the corresponding density centrifugation value as presented in Table 5.6. A summary of GC-content for each test is presented in Table 5.7.

When taking minimum lengths into consideration, Figures 5.9 and 5.10 represent the nucleotide distribution after DNA segmentation for no penalty and AIC tests, respectively. As well, we found that the BIC and MDL yielded identical results to those obtained without minimum lengths. Calculating the average experimental GC-content in the same manner as previously mentioned, we found the values for each section to be comparable to those obtained through density centrifugation as demonstrated in Tables 5.8 and 5.9. We summarized each test in Table 5.10.

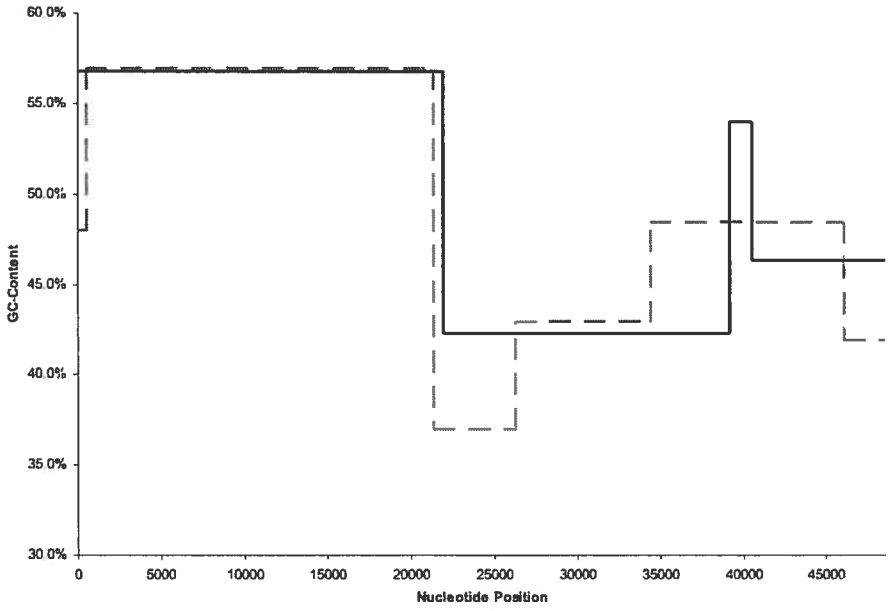


Figure 5.8. Nucleotide distribution in bacteriophage λ based on GC-content for BIC and MDL tests when using 2 classes (solid line) and density centrifugation (dashed line).

Segment	Nucleotide Start	Nucleotide End	%GC
1	1	21923	57
2	21924	39174	42
3	39175	40550	54
4	40551	48502	46

Table 5.5. Quantitative data describing the nucleotide distribution in bacteriophage λ for BIC and MDL tests (2 classes).

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	43
C	34437	46077	48	47

Table 5.6. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via BIC and MDL tests when using 2 classes.

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC
None	0	948.65906	0	100
AIC	2.0	3481.035	33	56
BIC	10.78936	768.2105	44	57
MDL	11.482508	766.13116	44	57

Table 5.7. Summary of experimental data for bacteriophage λ when using 2 classes.

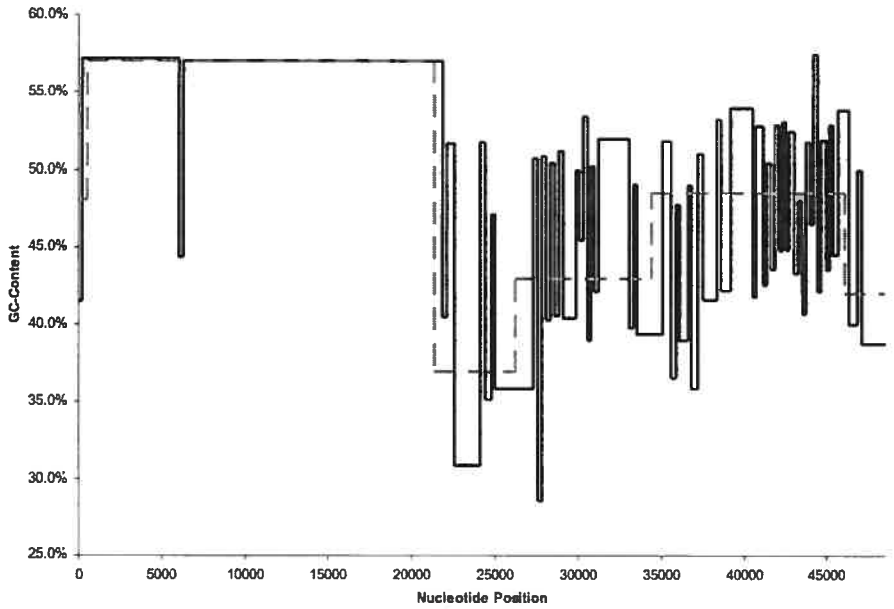


Figure 5.9. Nucleotide distribution in bacteriophage λ based on GC-content for no penalty test with minimum lengths when using 2 classes (solid line) and density centrifugation (dashed line).

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	40
C	34437	46077	48	47

Table 5.8. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via no penalty test with minimum lengths when using 2 classes.

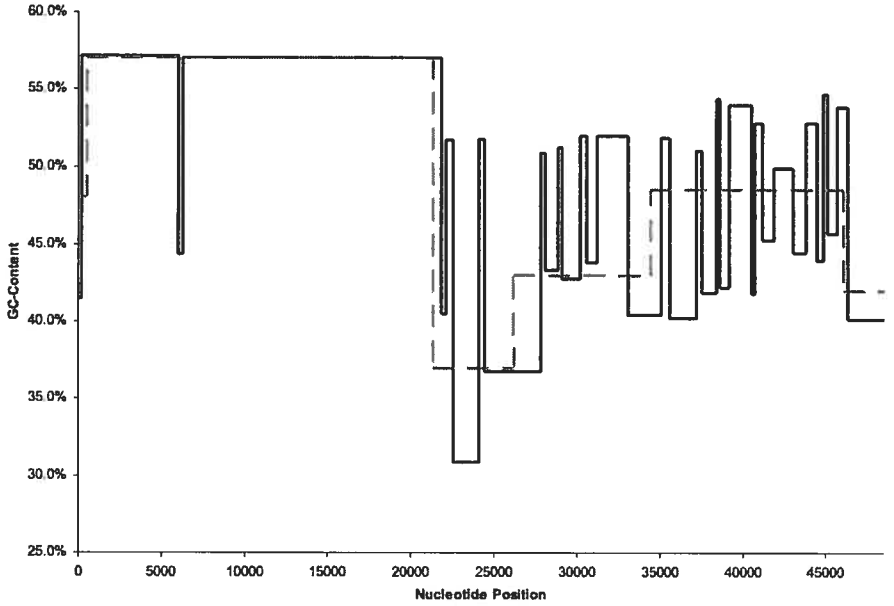


Figure 5.10. Nucleotide distribution in bacteriophage λ based on GC-content for AIC test with minimum length when using 2 classes (solid line) and density centrifugation (dashed line).

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	44
C	34437	46077	48	47

Table 5.9. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via AIC test with minimum lengths when using 2 classes.

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC
None	0	1830.736	39	55
AIC	2.0	1443.6943	40	56
BIC	10.78936	768.2105	44	57
MDL	11.482508	766.13116	44	57

Table 5.10. Summary of experimental data for bacteriophage λ when using 2 classes with minimum lengths.

Case 2: 3 Classes

We observed that the DNA segmentation obtained for the no penalty, AIC, BIC, and MDL tests were identical to those obtained in the case in which we considered 2 classes. Like the previous case, although we did not find an exact graphical match between the experimental and theoretical nucleotide distributions, we did find the average experimental GC-content within the three sections to be comparable to those obtained from density centrifugation as shown in Table 5.11. The summary of GC-content for each test is presented in Table 5.14, where the "-" markers represent classes that are not found in the DNA segmentation.

Taking the minimum lengths into consideration, we obtained the segmentation for no penalty and AIC tests as illustrated in Figure 5.11 and 5.12, respectively. Furthermore, we found the average experimental GC-content within the three longest sections to be similar to those obtained from density centrifugation as shown in Tables 5.12 and 5.13. The DNA segmentation results for the BIC and MDL tests were found to be identical to their counterparts in the 2 classes case. The GC-content for each test is summarized in Table 5.15.

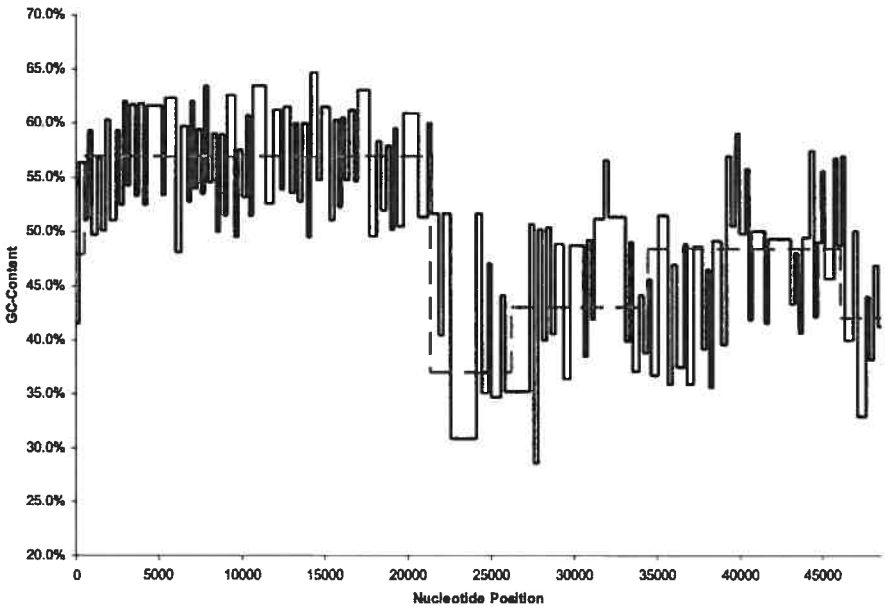


Figure 5.11. Nucleotide distribution in bacteriophage λ based on GC-content for no penalty test with minimum lengths when using 3 classes (solid line) and density centrifugation (dashed line).

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	44
C	34437	46077	48	47

Table 5.11. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via BIC and MDL tests with minimum lengths when using 3 classes.

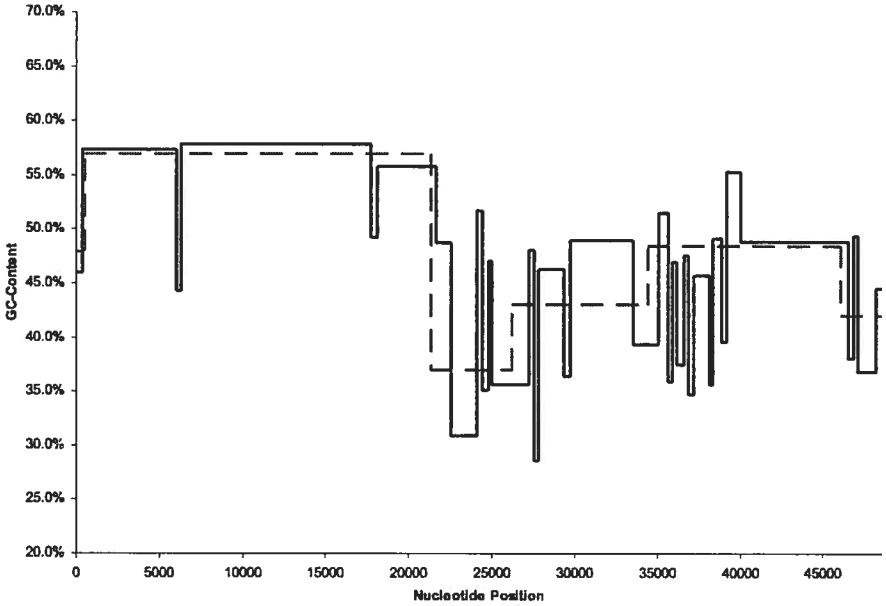


Figure 5.12. Nucleotide distribution in bacteriophage λ based on GC-content for AIC test with minimum lengths when using 3 classes (solid line) and density centrifugation (dashed line).

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	44
C	34437	46077	48	47

Table 5.12. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via no penalty test with minimum lengths when using 3 classes.

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	44
C	34437	46077	48	47

Table 5.13. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via AIC test with minimum lengths when using 3 classes.

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC	Class 2 %GC
None	0	8135.6226	0	-	100
AIC	2.0	14303.221	18	53	100
BIC	10.78936	768.2105	44	57	-
MDL	11.887973	764.9146	44	57	-

Table 5.14. Summary of experimental data for bacteriophage λ when using 3 classes.

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC	Class 2 %GC
None	0	2526.8127	37	50	60
AIC	2.0	2627.7852	36	48	57
BIC	10.78936	768.2105	44	57	-
MDL	11.887973	764.9146	44	57	-

Table 5.15. Summary of experimental data for bacteriophage λ when using 3 classes with minimum lengths.

Case 3: 4 Classes

Based on the results obtained from the previous cases, we found the risk of overfitting data to be high for the no penalty and AIC tests. Furthermore, we found the segmentation obtained from the BIC and MDL tests to be comparable to each other. Using the three longest sections defined by density centrifugation in Figure 5.13, the average experimental GC-content values within these boundaries were found to be comparable to those determined by density centrifugation as shown in Table 5.16.

We graphically compared the nucleotide distribution between the experimental and EMBOSS-calculated data using BIC penalty. As illustrated in Figure 5.14, it can be seen that the trend describing the experimental data (dashed line) gave an interestingly good estimation of the nucleotide distribution found by its EMBOSS counterpart (solid line). This was also attained when the defined minimum length values were applied.

Segment	Nucleotide Start	Nucleotide End	Theoretical %GC	Average Experimental %GC
A	486	21340	57	57
B	26192	34436	43	44
C	34437	46077	48	47

Table 5.16. Comparison of the nucleotide distribution in bacteriophage λ between those found via density centrifugation and those found via BIC test when using 4 classes.

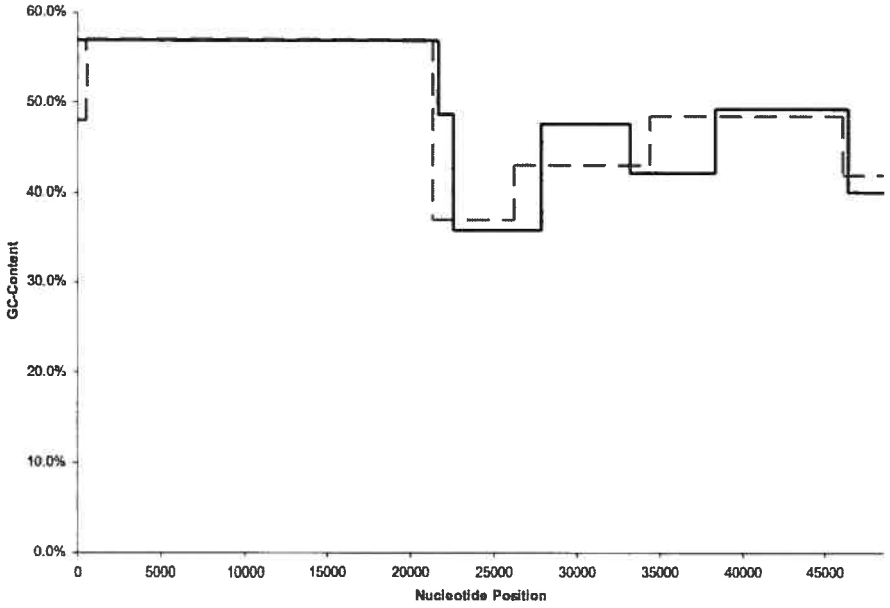


Figure 5.13. Nucleotide distribution in bacteriophage λ based on GC-content for BIC test when using 4 classes (solid line) and density centrifugation (dashed line).

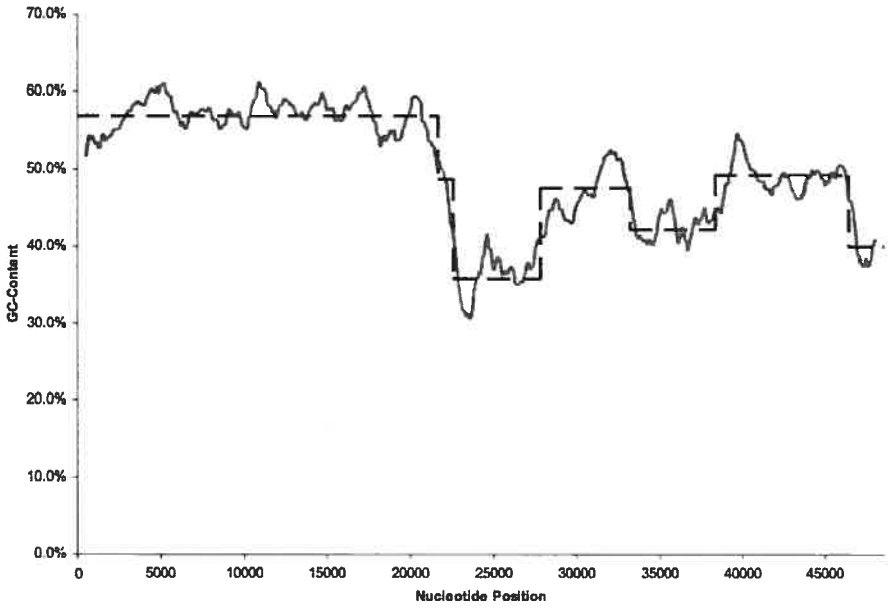


Figure 5.14. Nucleotide distribution in bacteriophage λ based on GC-content for BIC when using 4 classes (dashed line) and EMBOSS (solid line).

5.2 RNA Genes in Thermophiles

5.2.1 Description

Klein et al. (2002) used a hidden Markov model consisting of two states, labelled as "RNA" and "background genome", to segment the thermophile Archaebacteria *Methanocaldococcus jannaschii* (*M. jannaschii*) genome based on GC-content in order to find non-coding RNA genes. They defined the "RNA" state to be GC-rich in its transfer RNA (tRNA) and ribosomal RNA (rRNA) content and therefore was assigned a higher GC-content emission probability, whereas the "background genome" state is said to be GC-poor and therefore was assigned a low GC-content emission probability. According to the NCBI database, there are 6 known ribosomal RNA and 37 known transfer RNA segments found in the *M. jannaschii* genome.

Eddy (2001) determined that structural RNA genes in Prokaryotes tend to have a GC-content that is proportional to the optimal temperature growth, including those of tRNA and rRNA genes (Galtier and Lobry 1997). However, we found that this is not necessarily the case for Ribonuclease (RNase) P RNA genes in Prokaryotes as illustrated in Figure 5.15, where we analyzed the GC-content of the genes stored in the Ribonuclease P Database (*http://jwbrown.mbio.ncsu.edu/RNaseP*) (Brown 1999) and used the optimal growth temperature values found in the Prokaryotic Growth Temperature Database (*http://pgtdb.csie.ncu.edu.tw*) (Huang et al. 2004). This is also the case if we analyzed the GC-content in the helices of these genes as shown in Figure 5.16.

For *Methanocaldococcus jannaschii* in the NCBI database, the RNase P RNA gene stored in the ribonuclease P database can be found between base 643,507 and 643,758.

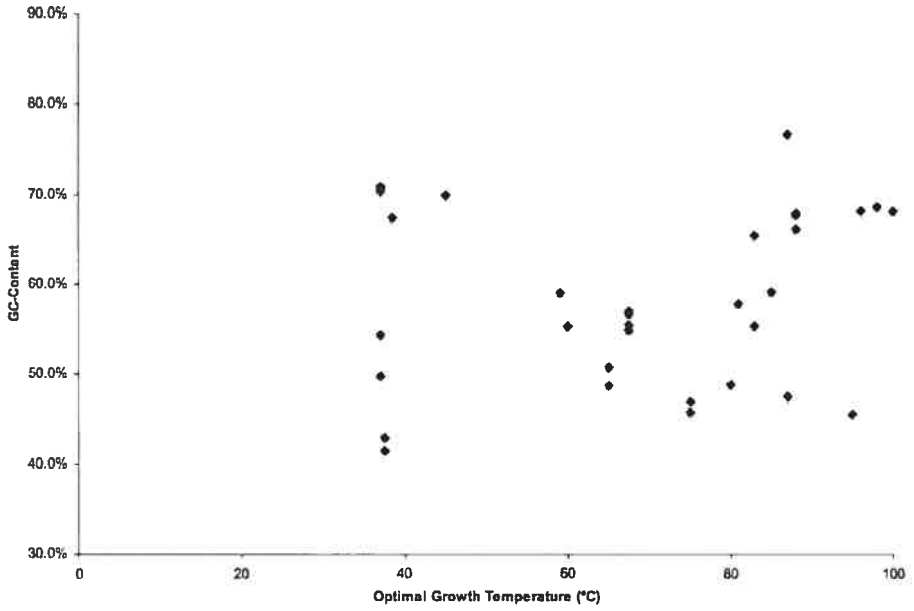


Figure 5.15. Comparison between GC-content of RNase P RNA genes in Prokaryotes versus optimal growth temperature.

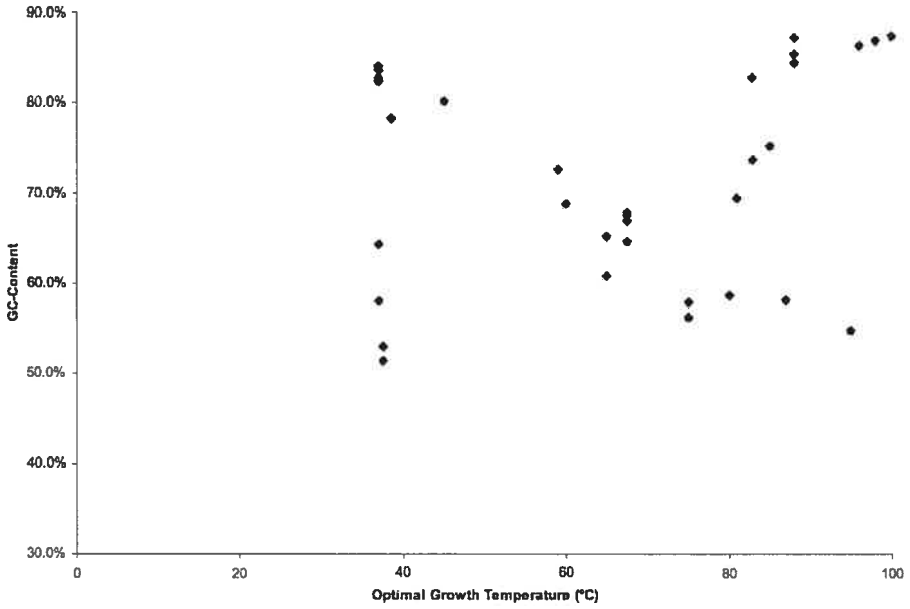


Figure 5.16. Comparison between helical GC-content of RNase P RNA genes in Prokaryotes versus optimal growth temperature.

5.2.2 Tests

We tested our algorithms on *M. jannaschii* (1,664,970 base pairs, GenBank accession NC_000909) using different complexity penalty values α as depicted in Table 5.17. We also tested the DNA segmentation on various number of segments with the probability values presented in Table 5.18 and the minimum length values listed in Table 5.19. The probability values are based on the genome's overall GC-content. Since the *M. jannaschii* genome consisted of RNA segments whose contents are GC-rich, we measured the nucleotide distribution and the number of RNA genes detected in our DNA segmentation tests.

Test Name	α
None	0
AIC	2
BIC	$\log 1{,}664{,}970$
MDL	$\log 1{,}664{,}970 + \log k$

Table 5.17. Complexity penalty values α tested for *M. jannaschii*.

5.2.3 Results

Case 1: 2 Classes

Using the AIC and no penalty values to segment the *M. jannaschii* genome, we found the nucleotide distribution of the sequence to be heavily segmented with GC-rich and GC-poor fragments. We observed data overftting in which the segmented sequence revealed numerous segments of length 1, although the degree of overfitting for AIC was found to be smaller than if we used no penalty at all. Because the sequence was heavily segmented, all ribosomal and transfer RNA segments given by the NCBI database were found.

Case	$p_j(x)$	
2 Classes	$p_0(S) = 0.34$	$p_0(W) = 0.66$
	$p_1(S) = 0.72$	$p_1(W) = 0.28$
3 Classes	$p_0(S) = 0.31$	$p_0(W) = 0.69$
	$p_1(S) = 0.64$	$p_1(W) = 0.36$
	$p_2(S) = 0.74$	$p_2(W) = 0.26$
6 Classes	$p_0(S) = 0.31$	$p_0(W) = 0.69$
	$p_1(S) = 0.60$	$p_1(W) = 0.40$
	$p_2(S) = 0.61$	$p_2(W) = 0.39$
	$p_3(S) = 0.62$	$p_3(W) = 0.38$
	$p_4(S) = 0.71$	$p_4(W) = 0.29$
	$p_5(S) = 0.72$	$p_5(W) = 0.28$
10 Classes	$p_0(S) = 0.31$	$p_0(W) = 0.69$
	$p_1(S) = 0.64$	$p_1(W) = 0.36$
	$p_2(S) = 0.74$	$p_2(W) = 0.26$
	$p_3(S) = 0.68$	$p_3(W) = 0.32$
	$p_4(S) = 0.72$	$p_4(W) = 0.28$
	$p_5(S) = 0.63$	$p_5(W) = 0.37$
	$p_6(S) = 0.65$	$p_6(W) = 0.35$
	$p_7(S) = 0.66$	$p_7(W) = 0.33$
	$p_8(S) = 0.69$	$p_8(W) = 0.31$
	$p_9(S) = 0.70$	$p_9(W) = 0.30$

Table 5.18. Probability values $p_j(x)$ tested for $M.\ jannaschii$.

Case	Class	Minimum Length
2 Classes	0	50
	1	50
3 Classes	0	100
	1	100
	2	100
6 Classes	0	100
	1	100
	2	100
	3	100
	4	100
	5	100
10 Classes	0	100
	1	100
	2	100
	3	100
	4	100
	5	100
	6	100
	7	100
	8	100
	9	100

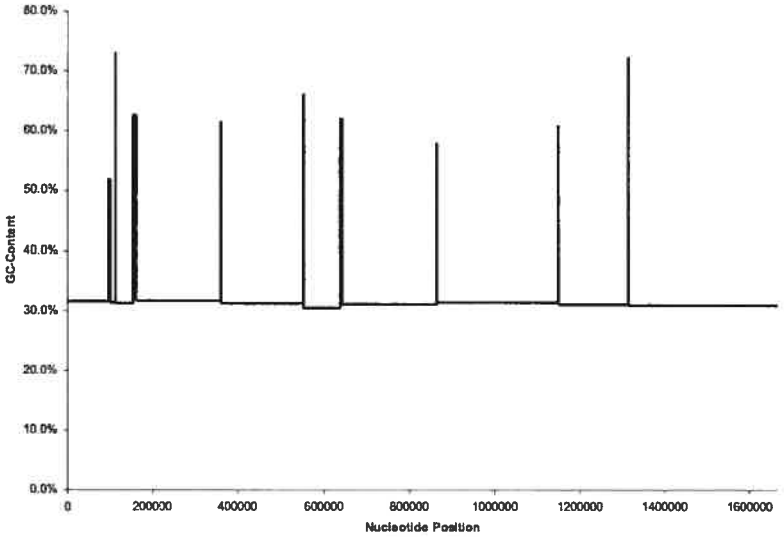
Table 5.19. Minimum length values tested for *M. jannaschii*.

From our DNA segmentation tests using BIC and MDL penalty values, we obtained the nucleotide distribution graphically shown in Figure 5.17 where there is no occurrence of overfitting (i.e., the length of each segment is greater than 1). As well, it can be seen that the segmentation revealed fragments that can be characterized as either GC-poor or GC-rich, where the GC-content of the former is approximately 30% and the majority of the latter is at least 60%. Table 5.20 shows the number of ribosomal and transfer RNA segments found in this genome as defined by the NCBI database, where all tests were able to find all ribosomal RNA segments. The BIC and MDL tests were able to find only 21 of the 37 known transfer RNA segments. However, many of the fragments that were determined experimentally contained more than one known RNA segment, whether it be rRNA or tRNA or both, and may have affected the GC-content of these fragments. For example, the segment [97326, 97823] contained 2 known tRNAs and was considered to be GC-rich, even though its GC-content is approximately 51.8%. Interestingly, we were also able to detect the RNase P RNA gene by comparing the associated GC-rich segment with the sequence associated with this gene found in the *M. jannaschii* genome at [643507.643758].

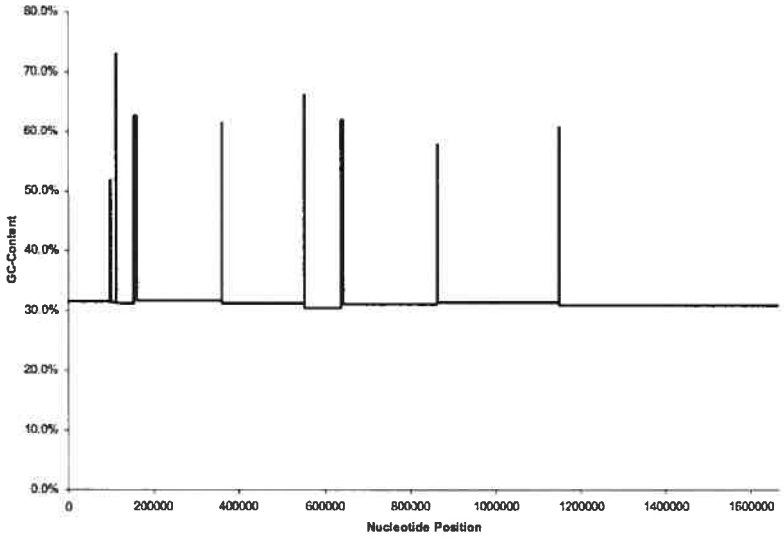
The GC-content for each class found in the segmentation is shown in Table 5.21. Although we used 2 classes to segment our sequence in order to determine GC-rich and GC-poor segments, we found the obtained GC-content values do not distinguish these two kinds of segments very well if we used either AIC or no penalties with minimum length values applied. Consequently, it may be difficult to distinguish between GC-rich and GC-poor segments when analyzing this segmentation. In contrast, the GC-content values calculated from the other tests were more characteristic of GC-rich and GC-poor segments and can be easily identified in the segmentation.

Case 2: 3 Classes

Like the case where two classes were considered, data overfitting was observed in the segmented *M. jannaschii* genome if we used AIC or no penalty values. Figures 5.18



(a) BIC



(b) MDL

Figure 5.17. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests when using 2 classes.

(a) Without minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	21	27
MDL	6	20	26

(b) With minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	21	27
MDL	6	20	26

Table 5.20. RNA found by implementation for *M. jannaschii* when using 2 classes without and with minimum lengths.

(a) Without minimum lengths

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC
None	0	392433.5	0.0	100.0
AIC	2.0	10919.063	29.4	78.5
BIC	14.325317	2292.9995	31.2	61.6
MDL	15.018465	2279.2048	31.2	61.6

(b) With minimum lengths

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC
None	0	74798.42	21.8	40.5
AIC	2.0	50029.78	21.7	36.5
BIC	14.325317	2292.9995	31.2	61.6
MDL	15.018465	2279.2048	31.2	61.6

Table 5.21. Summary of experimental data for *M. jannaschii* when using 2 classes without and with minimum lengths.

and 5.19 illustrates the nucleotide distribution from DNA segmentation using BIC and MDL for both without and with minimum length values.

For AIC and no penalty values, the DNA segmentation was able to reveal all the ribosomal and transfer RNA segments regardless of whether minimum length values were used or not. In contrast, as shown in Table 5.22, a fraction of transfer RNA were detected for BIC and MDL test cases. Like the previous case, the RNase P RNA gene for *M. jannaschii* was found.

Although segments in a sequence can be characterized as GC-rich and GC-poor, as demonstrated in the previous case, it may be possible to include additional GC-rich segments with different GC-content values. We segmented the *M. jannaschii* sequence using 3 classes, where we associated one class as GC-poor and two classes as GC-rich segments but with different GC-content values. As shown in Table 5.23, we could see how the *M. jannaschii* genome can be segmented based on this possibility. However,

our experimental data also demonstrated classes that were not found and therefore could be omitted from the segmentation.

(a) Without minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	22	28
MDL	6	20	26

(b) With minimum lengths

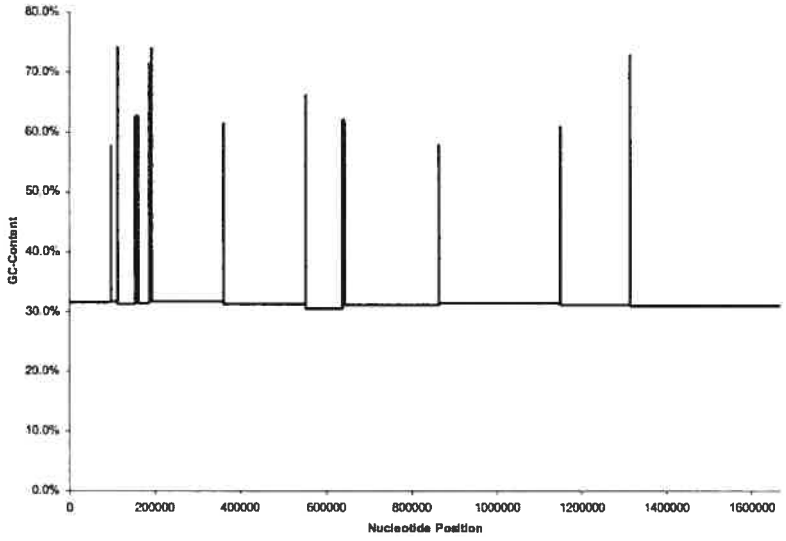
Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	20	26
MDL	6	19	25

Table 5.22. RNA found by implementation for *M. jannaschii* when using 3 classes without and with minimum lengths.

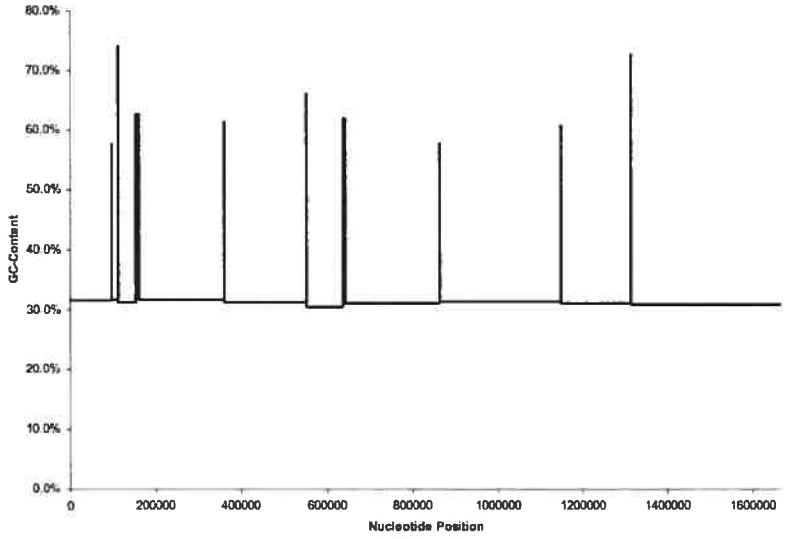
Case 3: 6 Classes

Overfitting was observed when we applied AIC or no penalty values to calculate the DNA segmentation of *M. jannaschii*. The nucleotide distribution for the BIC and MDL tests were illustrated Figures 5.20 and 5.21.

Unlike the AIC and no penalty tests in which the DNA segmentation revealed all the ribosomal and transfer RNA segments, a portion of transfer RNA were detected for BIC and MDL tests as shown in Table 5.24. The RNase P RNA gene for this genome was also observed.

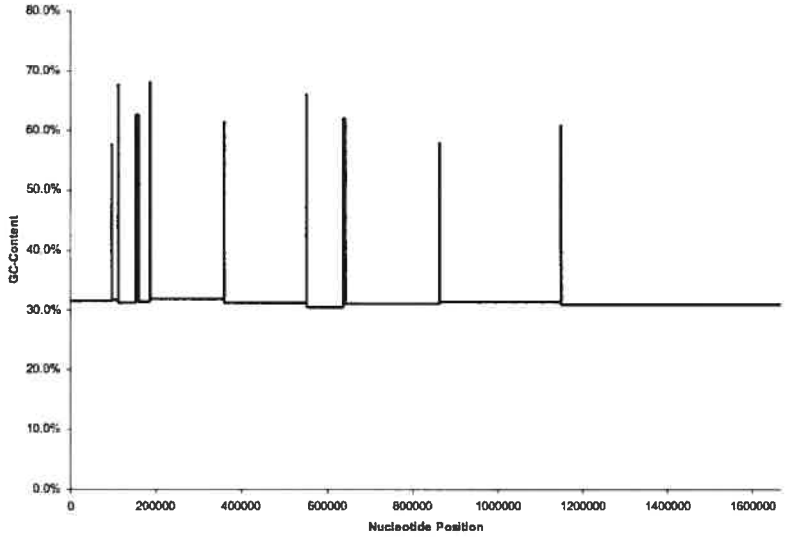


(a) BIC

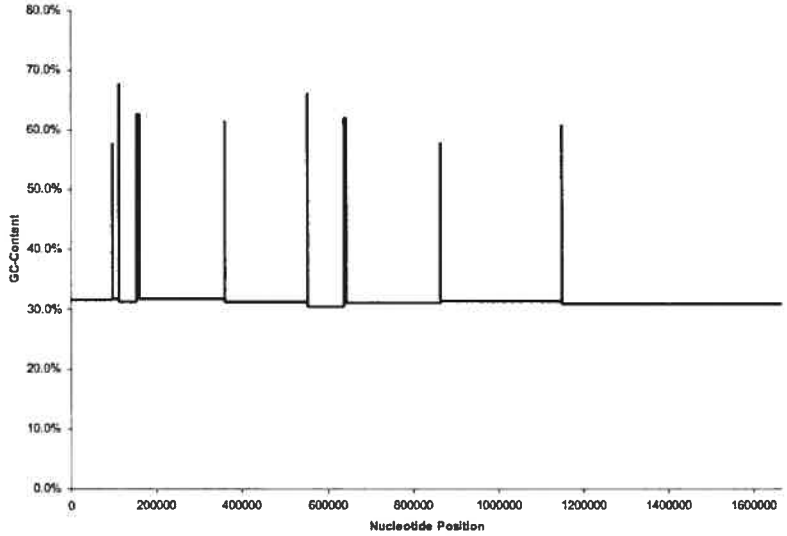


(b) MDL

Figure 5.18. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests when using 3 classes.



(a) BIC



(b) MDL

Figure 5.19. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests with minimum lengths when using 3 classes.

(a) Without minimum lengths

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC	Class 2 %GC
None	0	456894.3	0.0	-	100.0
AIC	2.0	174689.28	14.7	33.4	100.0
BIC	14.325317	2319.419	31.2	61.8	70.1
MDL	15.42393	2294.2566	31.2	61.8	67.9

(b) With minimum lengths

Test Name	Penalty	Score	Class 0 %GC	Class 1 %GC	Class 2
None	0	77393.234	20.8	31.1	42.0
AIC	2.0	9875.5	32.2	44.7	20.2
BIC	14.325317	2310.6177	31.2	61.9	-
MDL	15.42393	2290.3052	31.2	61.9	-

Table 5.23. Summary of experimental data for *M. jannaschii* when using 3 classes without and with minimum lengths.

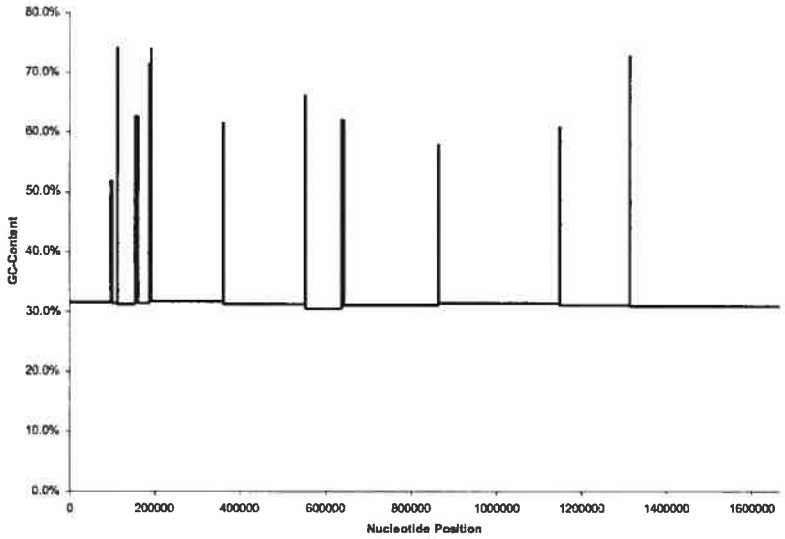
Tables 5.25 provides a summary of each DNA segmentation test for the *M. jannaschii* genome. Like before, it is possible that some segmentation classes can be omitted.

Case 4: 10 Classes

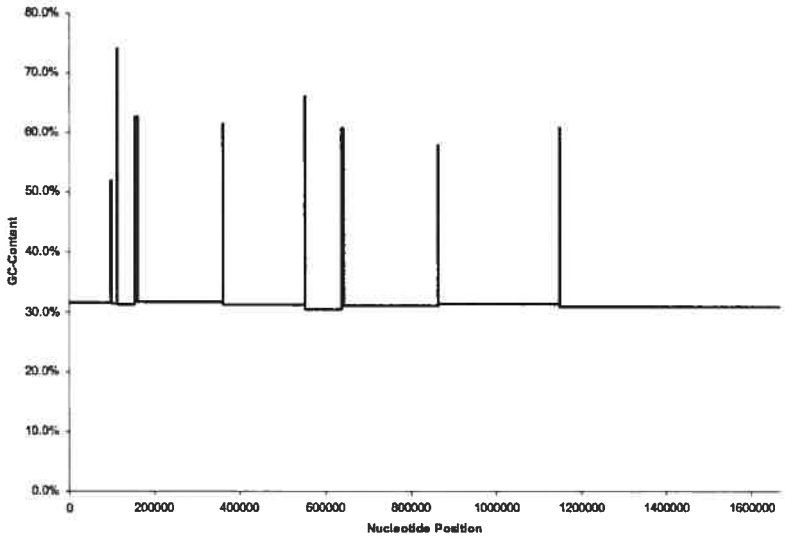
When we used the AIC or no penalty values to segment the *M. jannaschii* genome, the sequence was found to be heavily segmented and segmentation to be overfitted. Figures 5.22 and 5.23 gives a graphical representation of how the nucleotides are distributed based on GC-content.

Table 5.26 presents the number of ribosomal and transfer RNA segments found in comparison to those defined by the NCBI database. We also found the RNase P RNA gene for this genome as stored in the ribonuclease P database.

Even though we attempted to segment the *M. jannaschii* sequence using 10 classes

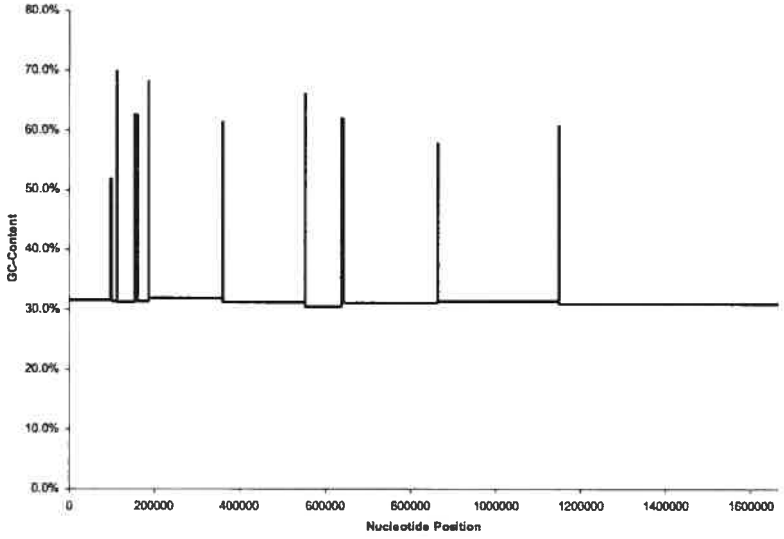


(a) BIC

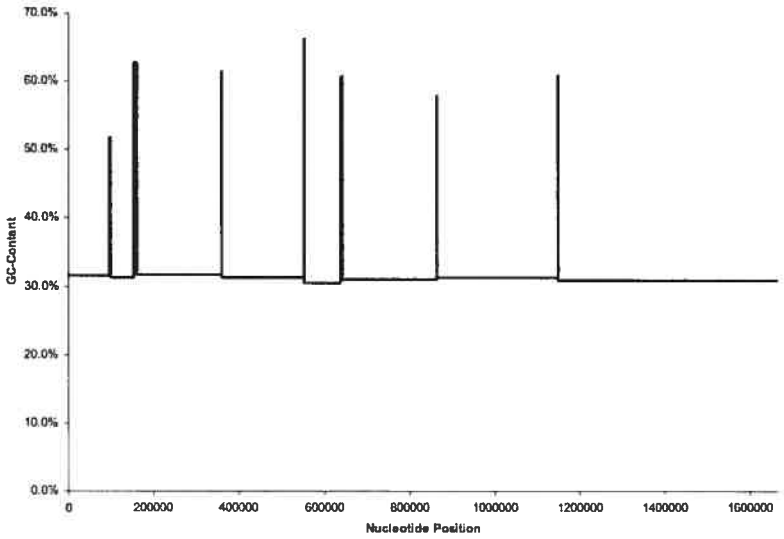


(b) MDL

Figure 5.20. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests when using 6 classes.



(a) BIC



(b) MDL

Figure 5.21. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests with minimum lengths when using 6 classes.

(a) Without minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	23	29
MDL	6	20	26

(b) With minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	21	27
MDL	6	19	25

Table 5.24. RNA found by implementation for *M. jannaschii* when using 6 classes without and with minimum lengths.

(a) Without minimum lengths

Test Name	Penalty	Score	Class	%GC	Class	%GC
None	0	441425.62	0	0.0	3	-
			1	-	4	100.0
			2	-	5	100.0
AIC	2.0	1328935.0	0	0.0	3	56.8
			1	23.0	4	78.2
			2	37.5	5	100.0
BIC	14.325317	2342.75	0	31.2	3	62.4
			1	55.4	4	-
			2	57.8	5	70.1
MDL	16.117077	2305.5107	0	31.2	3	62.7
			1	54.7	4	-
			2	60.6	5	63.8

(b) With minimum lengths

Test Name	Penalty	Score	Class	%GC	Class	%GC
None	0	116933.086	0	17.9	3	38.1
			1	25.1	4	44.8
			2	31.7	5	62.4
AIC	2.0	82396.46	0	19.2	3	40.2
			1	29.4	4	44.2
			2	34.9	5	62.6
BIC	14.325317	2334.582	0	31.2	3	62.4
			1	55.4	4	64.5
			2	57.8	5	-
MDL	16.117077	2305.8728	0	31.2	3	62.7
			1	54.7	4	-
			2	60.6	5	-

Table 5.25. Summary of experimental data for *M. jannaschii* when using 6 classes without and with minimum length.

that includes a variety of GC-content values, it is clear from the summary presented in Table 5.27 that some classes can be omitted since they were not found in the segmentation. Furthermore, the MDL penalty value may be too severe for segmentation since we observed fewer classes found than if we used BIC. Hence, it is possible to segment this sequence using 4 or 5 classes with the BIC segment transition penalty value applied.

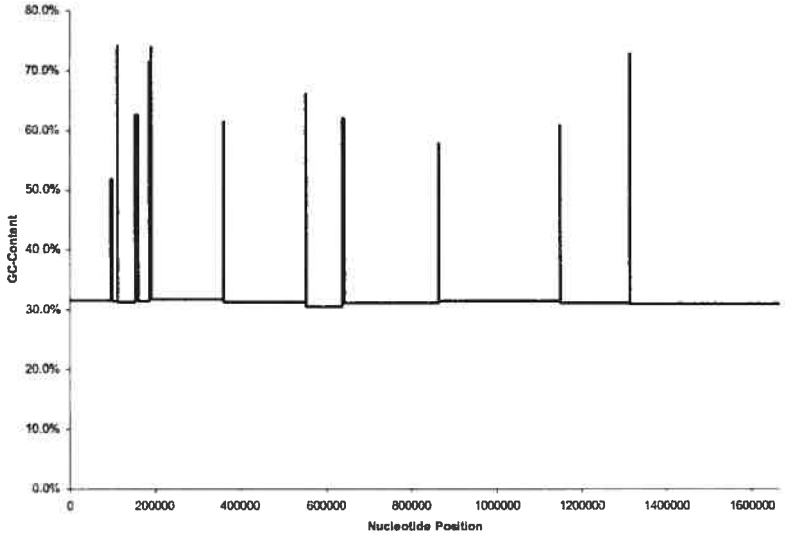
(a) Without minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	23	29
MDL	6	20	26

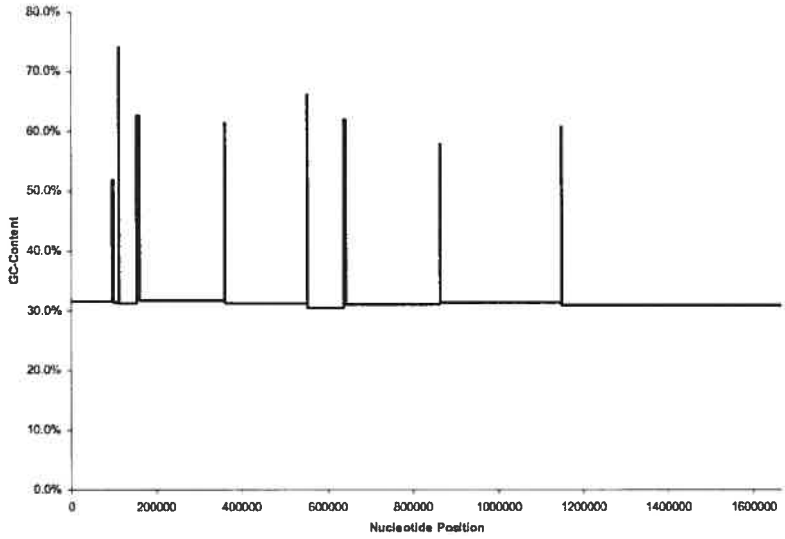
(b) With minimum lengths

Test Name	Number of rRNA found	Number of tRNA found	Number of rRNA and tRNA found
None	6	37	43
AIC	6	37	43
BIC	6	21	27
MDL	6	19	25

Table 5.26. RNA found by implementation for *M. jannaschii* when using 10 classes without and with minimum lengths.

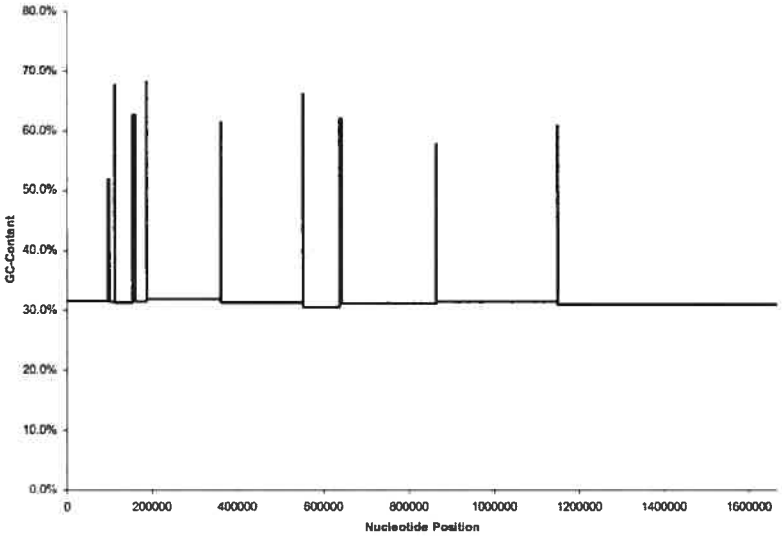


(a) BIC

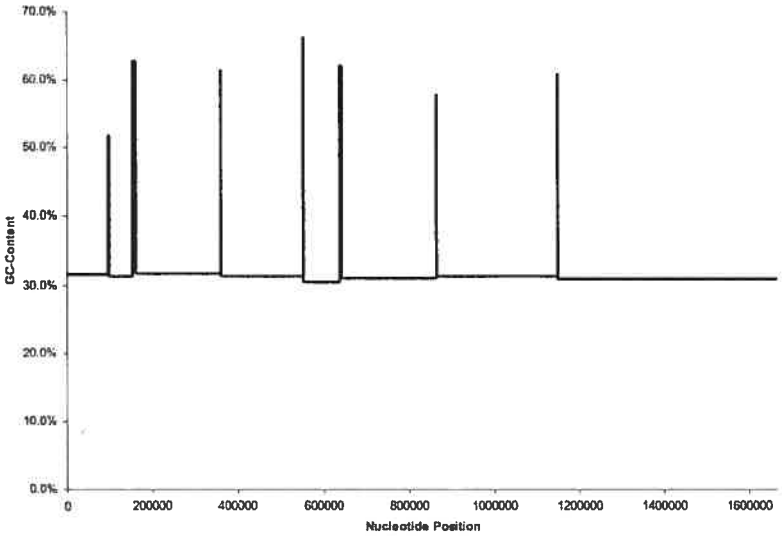


(b) MDL

Figure 5.22. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests when using 10 classes.



(a) BIC



(b) MDL

Figure 5.23. Nucleotide distribution in *M. jannaschii* based on GC-content for BIC and MDL tests with minimum lengths when using 10 classes.

(a) Without minimum lengths

Test Name	Penalty	Score	Class	%GC	Class	%GC	Class	%GC
None	0	456894.3	0	0.0	4	-	8	-
			1	-	5	-	9	-
			2	100.0	6	-		
			3	-	7	-		
AIC	2.0	308832.72	0	11.3	4	82.4	8	59.6
			1	31.2	5	22.4	9	71.6
			2	100.0	6	38.8		
			3	46.3	7	0.0		
BIC	14.325317	2330.747	0	31.2	4	66.7	8	-
			1	-	5	61.4	9	-
			2	68.1	6	63.4		
			3	-	7	-		
MDL	16.627903	2283.1248	0	31.2	4	-	8	-
			1	-	5	61.4	9	-
			2	63.8	6	63.4		
			3	-	7	-		

(b) With minimum lengths

Test Name	Penalty	Score	Class	%GC	Class	%GC	Class	%GC
None	0	456894.3	0	0.0	4	-	8	-
			1	-	5	-	9	-
			2	100.0	6	-		
			3	-	7	-		
AIC	2.0	65343.36	0	20.5	4	64.7	8	-
			1	36.0	5	30.6	9	-
			2	62.8	6	42.3		
			3	49.1	7	-		
BIC	14.325317	2322.5967	0	31.2	4	-	8	-
			1	-	5	61.4	9	-
			2	-	6	63.4		
			3	63.8	7	-		
MDL	16.627903	2281.4048	0	31.2	4	-	8	-
			1	-	5	61.4	9	-
			2	-	6	53.4		
			3	-	7	-		

Table 5.27. Summary of experimental data for *M. jannaschii* when using 10 classes without and with minimum length.

5.3 Major Histocompatibility Complex

5.3.1 Description

Using isochoric content as their parameter, Li et al. applied their recursive segmentation algorithm on the major histocompatibility complex (MHC) sequence on human chromosome 6p21. They obtained the result as illustrated in Figure 5.24 (Li et al. 2002), where they measured the GC-Content by using a moving window whose size is 150 kb and shift increment is 15 kb. The figure can be interpreted as follows: (A) The domain borders are represented by the vertical dotted lines, whereas the GC-content segmented domains are illustrated by the horizontal solid lines; (B) The segmentation strength s values are denoted by the vertical bars. According to the obtained graph, they observed that the segmentations at three known segment borders possessed the highest segmentation strength, where classes III and II are the most homogeneous segments. We confirmed their findings by using the EMBOSS software application using the same window size and shift increment values.

5.3.2 Tests

Like Li et al., we downloaded the MHC sequence found in the Sanger Center ("current concensus" version, 28 October 1999, 3,673,778 bases). We applied our algorithms using the complexity penalty values α as depicted in Table 5.28. As well, we tested the DNA segmentation using the probability values presented in Table 5.29 and the minimum length values listed in Table 5.30.

Test Name	α
BIC	$\log 3{,}673{,}778$
MDL	$\log 3.673{,}778 + \log k$

Table 5.28. Complexity penalty values α tested for MHC.

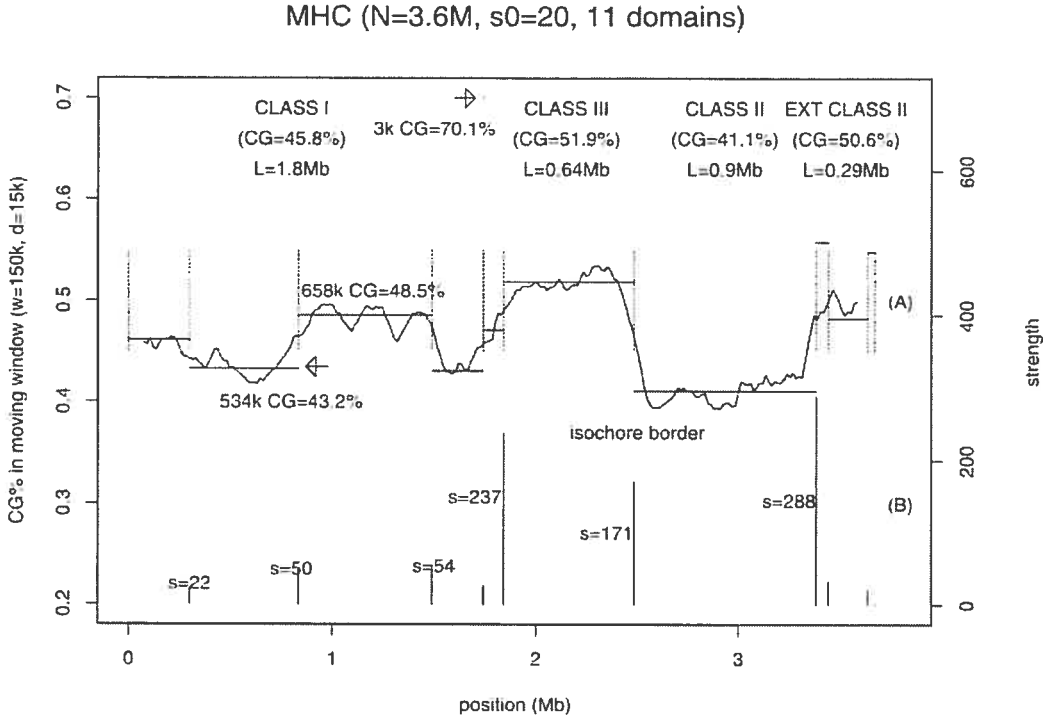


Figure 5.24. Segmented MHC sequence as found by Li et al. (Li et al. 2002)

Case	$p_j(x)$	
4 Classes	$p_0(S) = 0.458$	$p_0(W) = 0.542$
	$p_1(S) = 0.519$	$p_1(W) = 0.481$
	$p_2(S) = 0.411$	$p_2(W) = 0.589$
	$p_3(S) = 0.506$	$p_3(W) = 0.494$

Table 5.29. Probability values $p_j(x)$ tested for MHC.

Case	Class	Minimum Length
4 Classes	0	100,000
	1	100,000
	2	100,000
	3	100,000

Table 5.30. Minimum length values tested for MHC.

5.3.3 Results

According to the graph illustrated in Figure 5.25, we observed that our algorithm heavily segments the MHC sequence for the BIC test. If we take minimum lengths into consideration, however, we could characterize our obtained nucleotide distributions to be estimates of the distribution graph found by Li et al, as illustrated in Figure 5.26. Like Li et al., we found regions defined by classes II and III to be the most homogeneous, whereas the region defined by class I to be the least. Whether we apply minimum length values or not, we obtained identical nucleotide distribution graphs when we used the MDL penalty value.

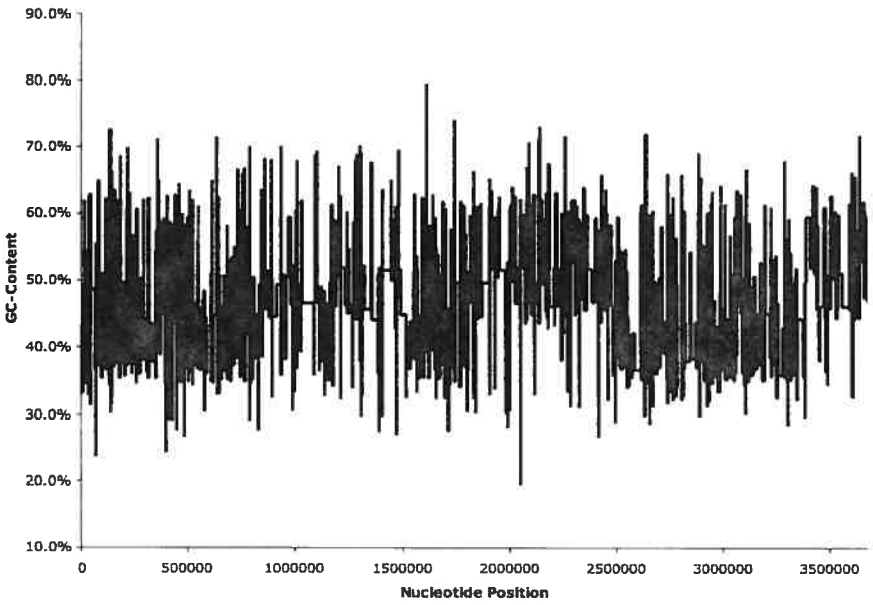


Figure 5.25. Nucleotide distribution in MHC based on GC-content for BIC and MDL tests when using 4 classes.

Figure 5.24 illustrated the segmentation of the MHC sequence, where it consisted of segments of at least 100-kb. As illustrated in Figure 5.27, we found the nucleotide

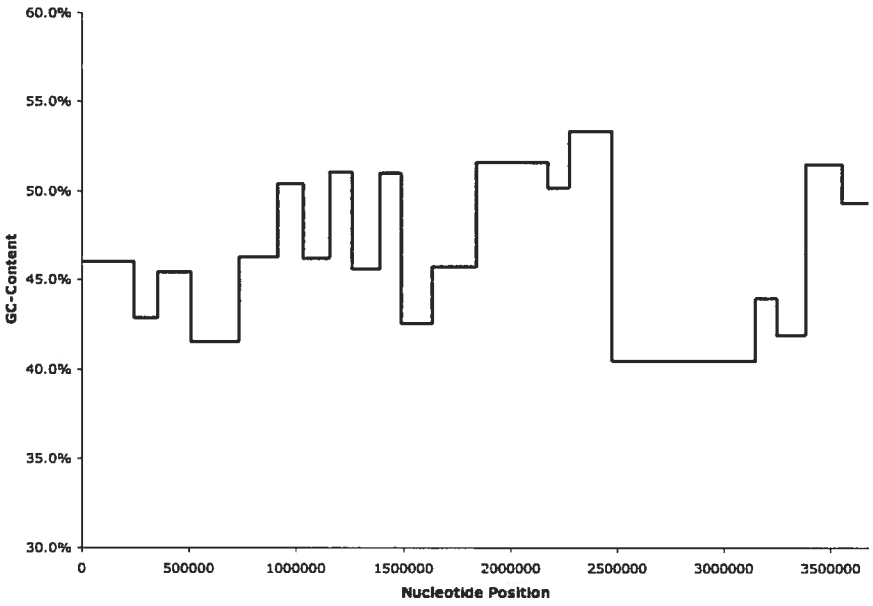


Figure 5.26. Nucleotide distribution in MHC based on GC-content for BIC and MDL tests with minimum lengths when using 4 classes.

distribution determined experimentally using the minimum lengths provided a reasonably good estimation of the distribution found by the EMBOSS software application, where we applied the same window size and shift increment values as those defined by Li et al.

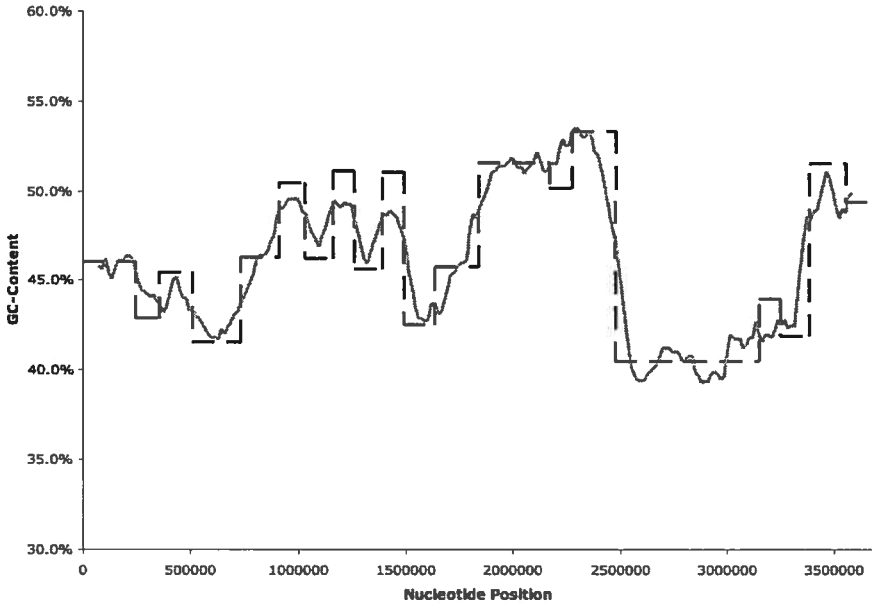


Figure 5.27. Nucleotide distribution in MHC based on GC-content for BIC test with minimum lengths when using 4 classes (dashed line) and EMBOSS (solid line).

Chapter 6

Conclusions

6.1 Discussion

In our research, we introduced different statistical models that can be used to segment a given genomic sequence: Bayesian approach, hidden Markov model, and different complexity penalties. As well, we discussed how segmentation algorithms can be also applied to other problems that do not involve segmenting sequences based on isochoric content.

Our research work presented two different implementations of the penalty-based best segmentation model: without and with minimum segment lengths. Using our implementations, we used the bacteriophage λ, *Methanocaldococcus jannaschii*, and MHC genome sequences to demonstrate how they can be segmented based on their GC-content. From our nucleotide distribution analysis of the bacteriophage λ and MHC genome sequences using 4 classes, we found the experimental segmentations gave an interestingly good graphical estimation of the graphs determined by the EMBOSS software application. In the case of the *Methanocaldococcus jannaschii* sequence, our implemented segmentation algorithms were able to identify all ribosomal and RNase P RNA fragments based on their GC-content. However, we found that the

number of transfer RNA fragments depends on the penalty and minimum segment lengths.

In general, we observed that the amount of segmentation in a sequence is inversely proportional to the complexity penalty values, that is, the number of segments tends to increase whenever the penalty is lower and vice versa. We found the risk of overfitting data to be higher when using AIC and no penalty values to segment our sequences. Furthermore, our tests demonstrated instances in which some probability classes were not found in the segmented sequence and therefore can be omitted. Finally, we found the MDL penalty value to be severe since, as demonstrated in our *Methanocaldococcus jannaschii* genome segmentation, there is a tendency for the number of found classes to be lower than if the BIC penalty value was applied.

6.2 Future Work

While the current implementation was able to successfully segment the sequences used in this project, some improvements could be made. For example, we would need to optimize the source code such that it can handle very long sequences since the current memory requirements is $O(n^2)$. Grice et al. (1997) proposed the checkpoint-based algorithm in which it would reduce memory requirements to $O(n \sqrt[L]{n})$, where L is some arbitrary integer, and would be applicable to the forward-backward training of linear hidden Markov models.

Although we were able to segment the bacteriophage λ, *M. jannaschii*, and major histocompatibility complex (MHC) on human chromosome 6 sequences, one additional test that would be of interest is the segmentation of the human genome where we could analyze the human isochores determined by our segmentation algorithms.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*, 716–723.

Baldi, P., S. Brunak, Y. Chauvin, and A. Krogh (1996). Naturally occurring nucleosome positioning signals in human exons and introns. *Journal of Molecular Biology 263*, 503–510.

Belle, E., N. Smith, and A. Eyre-Walker (2002). Analysis of the phylogenetic distribution of isochores in vertebrates and a test of the thermal stability hypothesis. *Journal of Molecular Evolution 55*, 356–363.

Bernaola-Galván, P., R. Róman-Roldán, and J. Oliver (1996). Compositional segmentation and long-range fractal correlation in DNA sequences. *Physical Review E 53*, 5181–5189.

Bernardi, G. (2000). Isochores and the evolutionary genomics of vertebrates. *Gene 241*, 3–17.

Bernardi, G. (2001). Misunderstandings about isochores, part 1. *Gene 276*, 3–13.

Bernardi, G. and G. Bernardi (1986). Compositional constraints and genome evolution. *Journal of Molecular Evolution 24*, 1–11.

Boffelli, D., J. McAuliffe, D. Ovcharenko, K. Lewis, I. Ovcharenko, L. Pachter, and E. Rubin (2003). Phylogenetic shadowing of primate sequences to find

functional regions of the human genome. *Science 299*, 1391–1394.

Boys, R. and D. Henderson (2004). A Bayesian approach to DNA sequence segmentation. *Biometrics 60*, 573–588.

Brown, J. (1999). The ribonuclease P database. *Nucleic Acids Research 27*, 314.

Burge, C. and S. Karlin (1997). Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology 268*, 78–94.

Burnham, K. and D. Anderson (2004). Multimodel inference: understanding AIC and BIC in model selection. *Sociological Methods & Research 33*, 261–304.

Churchill, G. (1989). Stochastic models for heterogenous DNA sequences. *Bulletin of Mathematical Biology 51*, 79–94.

Cohen, N., T. Dagan, L. Stone, and D. Graur (2005). GC composition of the human genome: in search of isochores. *Molecular Biology and Evolution 22*(5), 1260–1272.

Crowley, F., K. Roeder, and M. Bina (1997). A statistical model for locating regulatory regions in genomic DNA. *Journal of Molecular Biology 268*, 8–14.

Csűrös, M. (2004). Maximum-scoring segment sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 1*, 139–150.

Di Francesco, V., P. McQueen, J. Garnier, and P. Munson (1997). Incorporating global information into secondary structure prediction with hidden Markov models of protein folds. In *5th International Conference on Intelligent Systems for Molecular Biology*, pp. 100–103.

Eddy, S. (2001). Noncoding RNA genes and the modern RNA world. *Nature Reviews Genetics 2*, 919–929.

Ewens, W. and G. Grant (2001). *Statistical Methods in Bioinformatics: An Introduction*. Statistics for Biology and Health. New York, NY: Springer-Verlag New

York, Inc.

Eyre-Walker, A. and L. Hurst (2001). The evolution of isochores. *Nature Reviews Genetics 2*, 549–555.

Filipski, J. (1987). Correlation between molecular clock ticking, codon usage, fidelity of DNA repair, chromosome banding and chromatin compactness in germline cells. *FEBS Letters 217*, 184–186.

Filipski, J., J.-P. Thierry, and G. Bernardi (1973). An analysis of the bovine genome by Cs_2SO_4-Ag^+ density gradient centrifugation. *Journal of Molecular Biology 80*, 177–197.

Fryxell, K. and E. Zuckerkandl (2000). Cytosine deamination plays a primary role in the evolution of mammalian isochores. *Molecular Biology and Evolution 17*, 1371–1383.

Galtier, N. and J. Lobry (1997). Relationships between genomic G+C content, RNA secondary structures and optimal growth temperature in prokaryotes. *Journal of Molecular Evolution 44*, 632–636.

Grice, J., R. Hughey, and D. Speck (1997). Reduced space sequence alignment. *Computer Applications in the Biosciences 13*, 45–53.

Henderson, J., S. Salzberg, and K. Fasman (1997). Finding genes in DNA with an HMM. *Journal of Computational Biology 4*, 127–141.

Huang, S., L. Wu, H. Liang, K. Pan, and J. Horng (2004). PGTdb: a database providing growth temperatures of prokaryotes. *Bioinformatics 20*, 276–278.

Hurst, L. and A. Merchant (2001). High guanine-cytosine content is not an adaptation to high temperature, a comparative analysis amongst prokaryotes. *Proceedings of the Royal Society B: Biological Sciences 268*, 493–497.

Jordan, M. and T. Sejnowski (2001). *Graphical Models: Foundations of Neural Computation*. Cambridge, MA: MIT Press.

Klein, R., Z. Misulovin, and S. Eddy (2002). Noncoding RNA genes identified in AT-rich hyperthermophiles. *Proceedings of the National Academy of Sciences of the United States of America 99*, 7542–7547.

Krogh, A., M. Brown, L. Mian, K. Sjölander, and D. Haussler (1994). Hidden Markov models in computational biology: application to protein modeling. *Journal of Computational Biology 235*, 1501–1531.

Krogh, A., B. Larsson, G. Von Heijne, and E. Sonnhammer (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of Molecular Biology 305*, 567–580.

Kulp, D., D. Haussler, M. Reese, and F. Beckman (1996). A generalized hidden Markov model for the recognition of human genes in DNA. In *4th International Conference on Intelligent Systems for Molecular Biology*, pp. 134–141.

Lander, E., L. Linton, B. Birren, et al. (2001). Initial sequencing and analysis of the human genome. *Nature 409*, 860–921.

Li, W., P. Bernaola-Galvan, P. Carpena, and J. Oliver (2003). Isochores merit the prefix 'iso'. *Computational Biology and Chemistry 27*, 5–10.

Li, W., P. Bernaola-Galván, F. Haghighi, and I. Grosse (2002). Applications of recursive segmentation to the analysis of DNA sequences. *Computers and Chemistry 26*, 491–510.

Liu, J. and C. Lawrence (1999). Bayesian inference on biopolymer models. *Bioinformatics 15*(1), 38–52.

McAuliffe, J., L. Pachter, and M. Jordan (2004). Multiple-sequence functional annotation and the generalized hidden Markov phylogeny. *Bioinformatics 20*,

1850–1860.

Meselson, M., F. Stahl, and J. Vinograd (1957). Equilibrium sedimentation of macromolecules in density gradients. In *Proceedings of the National Academy of Sciences of the United States of America*, pp. 581–588.

Nicolas, P., L. Bize, F. Muri, M. Hoebeke, F. Rodolphe, S. Ehrlich, B. Prum, and P. Bessieres (2002). Mining *Bacillus subtilis* chromosome heterogeneities using hidden Markov models. *Nucleic Acids Research 30*, 1418–1426.

Paces, J., R. Zíka, V. Paces, A. Pavlícek, O. Clay, and G. Bernardi (2004). Representing GC variation along eukaryotic chromosomes. *Gene 333*, 135–141.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 257–286.

Ream, R., G. Johns, and G. Somero (2003). Base compositions of genes encoding alpha-actin and lactate dehydrogenase-a from differently adapted vertebrates show no temperature-adaptive variation in G+C content. *Molecular Biology and Evolution 20*, 105–110.

Rice, P., I. Longden, and A. Bleasby (2000). EMBOSS: the European molecular biology open software suite. *Trends in Genetics 16*, 276–277.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of Statistics 11*, 416–431.

Romero, P., Z. Obradović, C. Kissinger, J. Villafranca, and A. Dunker (1997). Identifying disordered regions in proteins from amino acid sequence. *The 1997 IEEE International Conference on Neural Networks Proceedings 1*, 90–95.

Saccone, C., A. De Sario, J. Wiegant, A. Rap, G. Della Valle, and G. Bernardi (1993). Corelations between isochores and chromosomal bands in the human

genome. In *Proceedings of the National Academy of Sciences of the United States of America*, pp. 11929–11933.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*, 461–464.

Setubal, J. and J. Meidanis (1997). *Introduction to Computational Molecular Biology*. Boston, MA: PWS Publishing.

Skalka, A., E. Burgi, and A. Hershey (1968). Segmental distribution of nucleotides in the DNA of bacteriophage lambda. *Journal of Molecular Biology 34*, 1–16.

Tusnády, G. and I. Simon (1998). Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *Journal of Molecular Biology 283*, 489–506.

Vinogradov, A. (2001). Bendable genes of warm-blooded vertebrates. *Molecular Biology and Evolution 18*, 2195–2200.

Vinogradov, A. (2003). DNA helix: the importance of being GC-rich. *Nucleic Acids Research 31*, 1838–1844.

Watson, J. (2004). *Molecular Biology of the Gene* (Fifth ed.). San Francisco, CA: Pearson/Benjamin Cummings.

Watson, J. and F. Crick (1953). A structure for deoxyribose nucleic acid. *Nature 171*, 737–738.

Wolfe, K., P. Sharp, and W.-H. Li (1989). Mutation rates differ among regions of the mammalian genome. *Nature 337*, 283–285.

Wootton, J. and S. Federhen (1993). Statistics of local complexity in amino acid sequences and sequence databases. *Computers and Chemistry 17*, 149–163.