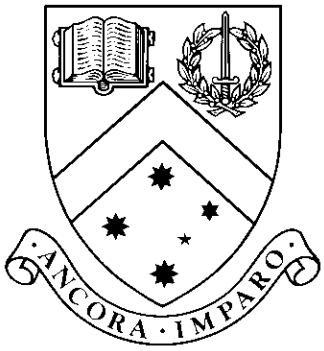
Information Theoretic Approaches to Biological Sequence Analyses

by

Minh Duc Cao, BCompSc



Thesis

Submitted by Minh Duc Cao

for fulfillment of the Requirements for the Degree of

Doctor of Philosophy (0190)

Supervisors:

Dr. Trevor Dix and Dr. Lloyd Allison

Clayton School of Information Technology
Monash University

September, 2010

Copyright notices

This thesis was typeset with LaTeX 2_ε by the author.

To grandpa – Kính tặng ông nội

Acknowledgments

I wish to express my deepest gratitude to my parents who have always been behind me on everything in my life. They have sacrificed much to ensure I would receive the best possible education and the best for my future. Long before I could teach them how to use a computer and to surf the Internet, they taught me the basis of computer science through mathematics and logic. Above all, they nurtured me with the love for science and the passion for doing research, which have become the foundation for this thesis. Thank you, Mom and Dad!

I am very grateful to my supervisors, Trevor Dix and Lloyd Allison, for their guidance, support and encouragement both in academic and life matters. They gave me the freedom to explore on my own interests, and at the same time, kept me on track of the research. They helped me overcome many crisis situations to finally get the thesis done. The discussions with them were always stimulating and inspiring. I am very thankful to the tremendous time they spent proof-reading my thesis. Thank you so much for supervising me in making a thesis that I can be proud of.

I was extremely lucky to have received advice, help and technical support from many people for the research leading to this thesis. I thank Robert Huestis for the useful biology lessons and numerous helpful discussions on the *Plasmodium* genomes. I thank Linda Stern for the insightful discussions and her advice on how to write a bioinformatics paper. I thank Torsten Seemann and Paul Harrison for giving many helpful comments and feedback on my research, and for providing a platform for my experiments. I am grateful to Ba Duc Nguyen, Dhananjay Thiruvady, Jessie Nghiem and Marsha Minchenko for proofreading parts of the thesis.

I also take this opportunity to acknowledge the Monash International Postgraduate Research Scholarship and the Monash Graduate Scholarship financial support. I am grateful to the Monash Research Graduate School, Australian Mathematical Sciences Institute (AMSI), the International Centre of Excellence for Education in Mathematics (ICE-EM) and the Clayton School of Information Technology for providing travel grants to attend various conferences and workshops. I also owe Justin Zobel and NICTA thanks for the generous support to cover parts of my publication fees.

Special thanks go to Trang Nguyen for her devoted support and encouragement throughout these difficult years. She has helped create the "intrinsic motivations" for me to complete the thesis. I truly appreciate her patience of listening to my talking about

my research though her background is not related to biology and computer science. Her thorough proof-reading and many valuable comments have immensely improved the presentation of this thesis. Of course, any remaining *speling* mistakes and *grammatically* errors are solely my responsibility.

It was a great pleasure to share an office with Amiza Amir, Arun Mani, Kerri Morgan and Marsha Minchenko. Thank you so much for the family-like atmosphere and all the fun we had. I am grateful to the O'Sullivans, Quynh, Neil, Dylan and Connor, for making me feel not too far from my family. Their warm reception and their beautiful flower farm offered me the distractions from my thesis whenever I needed. Thanks also go to Huy Hoang Ngo for sharing the frustrating and painful time during the writing of our theses. My circle of friends and relatives gave me constant support and care for the last few years. Thank you all!

My special gratitude goes to grandpa, who has always been very supportive. He has longed for seeing his beloved grandson becoming a doctor. Sadly, he could not wait to see his wish fulfilled. This thesis is dedicated to him.

Minh Duc Cao

Monash University
September 2010

Lời cảm ơn

Trước tiên, tôi xin bày tỏ lòng biết ơn sâu sắc đối với bố mẹ, người đã luôn hỗ trợ tôi trong tất cả mọi điều trong cuộc sống. Bố mẹ đã chịu hy sinh rất nhiều để đảm bảo cho tôi một sự giáo dục và những tiền đề tốt nhất cho tương lai. Rất lâu trước khi tôi có thể hướng dẫn bố mẹ cách sử dụng máy vi tính và lướt web, bố mẹ đã dạy cho tôi nền tảng cơ bản của khoa học máy tính qua toán học và logic học. Trên tất cả, bố mẹ đã truyền cho tôi tình yêu khoa học và niềm say mê nghiên cứu. Những điều đó đã trở thành nền tảng không thể thiếu cho luận văn này của tôi ngày hôm nay. Con cảm ơn bố mẹ!

Tôi cũng rất biết ơn các thầy hướng dẫn của tôi, tiến sỹ Trevor Dix và tiến sỹ Lloyd Allison, vì những lời khuyên quý báu, sự ủng hộ và khích lệ mà họ dành cho tôi trong công việc nghiên cứu cũng như trong cuộc sống. Hai thầy đã tạo điều kiện cho tôi được tự do khám phá theo sở thích của mình, đồng thời vẫn giữ tôi theo sát hướng nghiên cứu. Họ đã giúp tôi vượt qua được những giai đoạn khó khăn nhất để hoàn thành chương trình nghiên cứu. Những buổi thảo luận với thầy luôn mang lại nhiều cảm hứng thú vị. Tôi rất cảm ơn các thầy về khoảng thời gian dành cho việc đọc và nhận xét về luận văn của tôi. Xin cảm ơn các thầy đã hướng dẫn và giúp tôi hoàn thành một luận văn mà tôi có thể tự hào.

Tôi đã rất may mắn nhận được nhiều lời khuyên và sự giúp đỡ từ nhiều người trong quá trình nghiên cứu của mình. Tôi xin cảm ơn anh Robert Huestis vì đã dạy cho tôi nhiều kiến thức sinh học hữu ích và những buổi thảo luận về bộ gen của trùng sốt rét. Tôi xin cảm ơn cô Linda Stern về những lời khuyên của cô về phương pháp viết luận trong lĩnh vực tin sinh học. Tôi cũng cảm ơn các anh Torsten Seemann và Paul Harrison vì cho tôi những phản hồi quý giá về công trình nghiên cứu của tôi, và vì đã cung cấp việc truy cập vào một máy chủ để tôi thực hiện thí nghiệm. Tôi cũng xin cảm ơn các bạn Nguyễn Bá Đức, Dhananjay Thiruvady, Nghiêm Phương Thảo và Marsha Minchenko đã đọc và nhận xét một phần bản thảo của quyển luận văn này.

Nhân đây, tôi cũng xin cảm ơn sự hỗ trợ tài chính của quỹ học bổng nghiên cứu sau đại học cho sinh viên quốc tế (MIPRS) và quỹ học bổng sau đại học của trường đại học Monash (MGS). Tôi cũng rất biết ơn phòng nghiên cứu sau đại học trường đại học Monash, viện toán học Australia (AMSI), trung tâm quốc tế về giáo dục toán học (ICE-EM) và khoa công nghệ thông tin tại Clayton của trường đại học Monash đã tài trợ kinh phí cho tôi tham sự các hội thảo. Tôi cũng xin cảm ơn giáo sư Justin Zobel và NICTA đã hào phóng hỗ trợ một phần phí xuất bản cho tôi.

Lời cảm ơn đặc biệt xin dành cho Nguyễn Thị Huyền Trang vì sự ủng hộ tận tâm và sự chia sẻ những khó khăn với tôi trong mấy năm qua. Em đã tạo ra một nguồn "động lực nội tại" để tôi hoàn thành quyển luận văn này. Tôi rất cảm kích sự chịu khó lắng nghe và những nhận xét của em về công trình nghiên cứu của tôi mặc dù chuyên ngành của em không phải là tin học và sinh học. Việc tận tình đọc và chỉnh sửa bản thảo của em đã góp phần tạo nên chất lượng của quyển luận văn này. Tất nhiên, tất cả những sai sót còn sót lại hoàn toàn thuộc về trách nhiệm của tôi.

Tôi đã rất thoải mái khi được làm việc cùng phòng với các bạn Amiza Amir, Arun Mani, Kerri Morgan và Marsha Minchenko. Cảm ơn các bạn rất nhiều vì bầu không khí thân mật như trong một gia đình và những niềm vui mà chúng ta đã chia sẻ cùng nhau. Tôi rất biết ơn gia đình nhà O'Sullivans, chị Quỳnh, anh Neil và 2 cháu Bầu Bí, về sự đón tiếp nồng hậu mỗi lần tôi đến thăm trang trại hoa tuyệt vời của họ sau những ngày tháng nghiên cứu vất vả. Cả gia đình đã làm cho tôi vợi đi nỗi nhớ nhà rất nhiều. Tôi cũng muốn cảm ơn anh Ngô Huy Hoàng vì đã chia sẻ quãng thời gian khó khăn vất vả khi chúng tôi cùng viết luận văn. Gia đình và bè bạn đã luôn mang đến cho tôi những sự khích lệ và quan tâm thật quý giá. Xin cảm ơn tất cả!

Tôi muốn bày tỏ sự biết ơn đặc biệt đến ông nội, người đã luôn quan tâm theo dõi những bước tiến của tôi. Ông luôn mong mỏi được trông thấy người cháu của mình trở thành tiến sỹ. Thật đáng tiếc là ông đã không chờ được đến lúc ước mơ của mình trở thành hiện thực. Quyển luận văn này xin được thành kính dâng lên hương hồn ông.

Cao Minh Đức

Trường Đại Học Monash
Tháng 9, 2010

Contents

List of Tables . . . . . **xii**

List of Figures . . . . . **xiii**

Abstract . . . . . **xv**

1 Introduction **1**

1.1 Motivation 2

1.1.1 Biological Sequence Compression 2

1.1.2 Sequence Analysis 5

1.2 Research Objectives 6

1.3 Contributions 8

1.4 A Tour of the Thesis 11

1.5 Publications 12

1.6 Summary 13

2 Background **15**

2.1 Biological Background 15

2.1.1 Molecules of the Cells 15

2.1.2 The Genetic Code and the Central Dogma 19

2.1.3 Mechanisms of Evolution 24

2.1.4 Genome – Structure, Diversity and Size 26

2.2 Information Theory and Inference 29

2.2.1 Probability and Information 29

2.2.2 Statistical Inference and Minimum Message Length 33

2.2.3 Compression: Coding and Modelling 34

3 Biological Sequence Compression **39**

3.1 Introduction . . . 39
3.2 Applications of Biological Sequence Compression . . . 41
3.3 Biological Sequence Compression Review . . . 43
3.3.1 Compressible Features of Biological Sequences . . . 43
3.3.2 Review of Biological Sequence Compression Algorithms . . . 45
3.3.3 Analysis of Available Techniques for Biological Sequence Compression . . . 48
3.4 The Expert Model Compression Algorithm . . . 50
3.4.1 Types of Experts . . . 51
3.4.2 Proposing Repeat Experts . . . 52
3.4.3 Combining Experts' Predictions . . . 53
3.4.4 Variants of the Expert Model . . . 56
3.5 Experimental Results . . . 57
3.5.1 Comparison of DNA Compression Results . . . 58
3.5.2 Comparison of Protein Compression Results . . . 64
3.5.3 Compressibility of Genomes . . . 65
3.5.4 Information Content of Sequences . . . 68
3.5.5 Conditional Compression of Sequences . . . 70
3.6 A Side Application: Repeat Detection . . . 72
3.7 Discussion . . . 74
3.8 Summary . . . 77

4 Genomic Sequence Alignment by Compression . . . 79
4.1 Introduction . . . 79
4.2 Related Works . . . 81
4.3 Sequence Alignment Using Compression . . . 84
4.3.1 The Expert Model Compression Revisited . . . 86
4.3.2 Identifying Similar Regions . . . 87
4.4 Experimental Results . . . 90
4.4.1 Simulated Data . . . 90
4.4.2 Human-Mouse Data Set . . . 97
4.4.3 Malaria Data Set . . . 98
4.5 Visualisation of Alignment . . . 103
4.6 A Side Application: Computing Substitution Matrices . . . 105
4.6.1 Method . . . 106

4.6.2 Experimental Results . . . 107
4.7 Discussion . . . 109
4.8 Summary . . . 110

5 Phylogenetic Tree Construction . . . 113
5.1 Introduction . . . 113
5.2 Phylogenetic Analysis Background . . . 115
5.3 Review of Phylogenetic Analysis Methods . . . 120
5.4 An Information Theory Distance Measure . . . 127
5.5 Experimental Results . . . 133
5.5.1 Simulated Data . . . 133
5.5.2 Real Data . . . 138
5.6 A Side Application: Phylogenetics Analysis of Genomes . . . 141
5.6.1 Experimental Results . . . 143
5.7 Summary . . . 146

6 Conclusion . . . 149
6.1 The Expert Model . . . 149
6.2 Sequence Alignment . . . 150
6.3 Phylogenetic Analysis . . . 151
6.4 Closing Remarks . . . 152

List of Tables

2.1 Properties of 20 amino acids. 19
2.2 The dictionary of the genetic code. 22
2.3 Approximate genome sizes, gene numbers and gene densities of various organisms. 27

3.1 Description of the sequences in the DNA data set. 58
3.2 DNA compression by general purpose algorithms. 60
3.3 DNA compression by special purpose algorithms. 61
3.4 Compression results (in bps) and running time (in seconds). 63
3.5 Comparison of protein compression. 65
3.6 Compression of the human genome. 66
3.7 The compressibility of various genomes. 67

4.1 *Plasmodium* genomes characteristics. 99
4.2 Sensitivity and specificity of exon detection from the *P. falciparum* genome. 100
4.3 Sensitivity and specificity of exon detection from the *P. knowlesi* genome. . 101
4.4 The target and computed substitution matrices in the experiment with simulated data. 108
4.5 The substitution matrices of different malaria genomes. 109

5.1 Numbers of possible topologies for a group of species. 116
5.2 Common models of nucleotide substitution. 118
5.3 Performance evaluation of tree building methods on the data set, grouped by tree sizes. 136
5.4 Performance evaluation of tree building methods on the data set, grouped by diversity levels. 137
5.5 *Plasmodium* genomes characteristics. 144

List of Figures

2.1 The double helix structure of DNA. 17
2.2 DNA Replication. 17
2.3 Biosysthesis of a protein. 20
2.4 The translation process. 21
2.5 Flow of information in the central dogma. 23

3.1 Mechanism of sequence compression by the expert model. 50
3.2 The estimated information content of HUMHBB sequence. 69
3.3 A dot matrix plot of the HUMHBB sequence. 70
3.4 The estimated conditional information content of a region on the human chromosome 22 in differing contexts. 71
3.5 The information content sequences of HUMHBB produced by various experts. 73

4.1 Transmission with and without a reference sequence. 84
4.2 Comparative Performance on Different Distances. 93
4.3 Comparison of Performance on Different Compositions. 94
4.3 Comparison of Performance on Different Compositions (cont.). 95
4.4 Relationship between compressibility and alignment performance. 96
4.5 Performance Comparison on Human-Mouse data set. 98
4.6 Visualisation of the alignment of the *P. vivax* contig ctg6843 against the *P. falciparum* genome. 104

5.1 Plots of the ratio $y = \frac{\mathcal{I}(X|Y)}{\mathcal{I}(X)}$ and the function $y = 1 - p^2$ where $p = e^{-8\alpha t}$. 130
5.2 The approximate linearity of the proposed distance measure D. 131
5.3 The generation of a random phylogeny. 134
5.4 The phylogeny of mammals inferred by XMDistance. 139

5.5 The phylogeny of primates inferred by XMDistance. 140
5.6 The inferred phylogenetic tree of the *Plasmodium* genus. 144
5.7 The inferred phylogenetic tree of the γ-Proteobacteria group. 146

Information Theoretic Approaches to Biological Sequence Analyses

Minh Duc Cao, BCompSc
minhduc@monash.edu
Monash University, 2010

Supervisors:
Dr. Trevor Dix *Trevor.Dix@monash.edu*
Dr. Lloyd Allison *Lloyd.Allison@monash.edu*

Abstract

Molecular biology is the first information processing system on the planet. The genome sequence of an organism stores the genetic information that virtually defines the organism. Analysis of genomic sequences can help elucidate many aspects of life. This thesis investigates approaches for sequence analysis that make use of the information content of the sequences.

The information content of a sequence can be estimated by lossless compression. The thesis develops the expert model, a fast and effective algorithm for compression of biological sequences. The expert model uses a novel adaptive technique to combine predictions from different sub-models for compression based on the well-founded Bayesian statistical framework. Experiments show that the expert model outperforms existing biological compression algorithms on standard DNA and protein sequence data sets while maintaining a practical running time. Moreover, the expert model is capable of compressing long sequences. It is applied to estimate the information content of the genomes of species at various organism levels, including viruses, bacteria, archaea, single cell eukaryotes, invertebrates, plants and mammals. Most importantly, the expert model can produce an estimate of the information content of every symbol in a sequence using background knowledge in the form of known sequences or contexts. This is useful for performing information extraction from genomic sequences.

The thesis suggests that since genomic sequences carry genetic information, sequence analysis can be performed at the *information level*. A method for pairwise local alignment of genomes, namely XMAligner, is presented. Instead of comparing sequences at the character level, XMAligner considers a pair of sequences to be related if their mutual information is significant. XMAligner is shown to be superior to conventional alignment methods, especially on distantly related sequences or statistically biased data. The

method aligns sequences of eukaryote genome size with only modest hardware requirements. Importantly, the method has an objective function which can obviate the need to choose parameter values for high quality alignment.

The information content of sequences can also be used for phylogenetic analysis. The thesis formulates XMDistance, a measure of genetic distances between sequences based on their information content estimated by lossless compression. The measure does not rely on an evolutionary model. It is shown to be proportional to elapsed time if the evolutionary rate is constant. The distance measure can be used for phylogenetic analysis of sequences that cannot be reliably aligned, for example, whole genomes. On a set of simulated data, phylogenetic analysis using XMDistance outperforms maximum parsimony method and the standard character-based distance measure. For small sequences, the maximum likelihood method, which requires much longer time to run, performs better. XMDistance successfully infers plausible trees from real data, and most importantly manages problematic sets of whole genome sequences.

Information Theoretic Approaches to Biological Sequence Analyses

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Minh Duc Cao
27^{th} September 2010

Chapter 1

Introduction

'When I use a word,' Humpty Dumpty said, in rather a scornful tone, 'it means just what I choose it to mean – neither more nor less.'
'The question is,' said Alice, 'whether you can make words mean so many different things.'
'The question is,' said Humpty Dumpty, 'which is to be master – that's all.'
–Lewis Carroll

The purpose of genomic sequences is to store genetic *information* of living organisms. In the light of information theory (Shannon, 1948) and the Minimum Message Length framework (Wallace and Boulton, 1968; Wallace and Freeman, 1987; Wallace, 2005), this thesis investigates information theoretic approaches to biological sequence analysis. As a departure from the traditional character-based analysis methods, these approaches perform biological sequence analysis at the *information content* level. A fast and effective algorithm for compression of genomic sequences is developed for the purpose of estimating the information content of the sequences. Importantly, the compression algorithm is capable of compressing long sequences. The information theoretic approaches presented in this thesis use the information content of biological sequences for several important sequence analysis tasks including sequence alignment and phylogenetic analysis. These approaches are shown to be effectively applied to analyse problematic data such as statistically biased data and distantly related sequences.

1.1 Motivation

The motivation for this research lies in two main directions. Firstly, while the compression of biological sequences is useful for the study of biological data and biological processes, existing compression methods do not perform well or are unable to handle long sequences. It is therefore necessary to develop a biological sequence compression algorithm that overcomes these limitations. Secondly, many conventional sequence analysis methods do not cope with the size and diversity of the increasing genomic databases. Information-theoretic approaches, with the aid of the developed biological sequence compression algorithm, provide a potential solution to the challenge.

1.1.1 Biological Sequence Compression

Molecular biology is the first information processing system on the planet. The genome of an organism, which is a long deoxyribonucleic acid (DNA) sequence, contains all genetic instructions to control the development of the organism. The double-helix structure of DNA facilitates the replication of DNA molecules and thereby allows the transfer of information from one cell to a new cell, and from an organism to its offspring. The genetic information in the genome, via intermediary ribonucleic acid (RNA), is decoded using the genetic code to synthesise proteins which are the main actors in the metabolic pathways of the organism. Virtually all the characteristics of the organism are defined by the set of proteins produced in its cells, and therefore by the information in its genome. There arises an important research question: how much information is contained in a genome and in every base of the genome. It is also intriguing to be able to see what information is shared between the genomes of any two organisms.

The genetic information of an organism, which can be as simple as a virus or as complex as a human, is contained in one or more DNA molecules, each of which is a sequence over four nucleotides, namely *adenine* (A), *cytosine* (C), *guanine* (G) and *thymine* (T). This is equivalent to a string over the alphabet of the four letters A, C, G and T. Thus the problem of measuring the information content of a genomic sequence can be viewed as the estimation of the information content of the corresponding string.

In information theory (Shannon, 1948), the information content of a string depends on the probability of the string given a statistical model: it is defined as the negative logarithm of the probability. Since biological processes are not fully understood, the probability of a genomic sequence cannot be computed correctly. Fortunately, the lossless compression of a string gives an approximation of its information content. The amount of information contained in a string is no more than the size of the compressed message of the string because the string can be restored precisely from the compressed message.

The precision of the approximation largely depends on the performance of the compression. The design of a compression scheme for biological sequences is, therefore, useful for studying the information content of these sequences. This research addresses this by developing the *expert model* (XM), a biological sequence compression algorithm that importantly can manage genome-sized sequences quickly with excellent compression (Cao et al., 2007; Giancarlo et al., 2009; Cao et al., 2010a; Nalbantoglu et al., 2010). The algorithm is presented in Chapter 3.

A related concept, the Kolmogorov complexity of a string (Solomonoff, 1964; Kolmogorov, 1965; Martin-Lof, 1966; Chaitin, 1966) is defined as the size of the shortest program that can run on some fixed universal computer and output the string. However, Kolmogorov complexity is incomputable; there exists no program that takes a string as input and outputs the Kolmogorov complexity of the string. Again, the compression of the string is often used as an estimate of the Kolmogorov complexity of the string.

Compression is often said to consist of two components: modelling and coding (Rissanen and Langdon, 1981). Modelling involves describing the redundancy in the data in the form of a predictive model which gives the probability of each symbol in the data stream. The coder then encodes each symbol with respect to the probability given by the model. There exist coding techniques, notably arithmetic coding (Rissanen and Langdon, 1979; Witten et al., 1987), that can encode a symbol with a code-word length arbitrarily close to the theoretical limit. Therefore, the performance of a compression scheme is largely determined by the prediction ability of the model. Compression performance, in a way, indicates how good the modelling is. In the present context, the key problem in the compression of biological sequences is the modelling of biological processes.

Apart from the apparent benefit of saving data storage and reducing communication bandwidth, the primary purpose of biological sequence compression in this research is for inductive inference. Any computer system doing automatic inference must guard against over-fitting. There is extensive practical and theoretical evidence (Solomonoff, 1964; Herzel, 1988; Wallace, 2005; Allison, 2005) in support of information theoretic compression methods being very effective at balancing model complexity against the fitting of the model to data. Model selection in the Minimum Message Length (MML) framework (Wallace and Boulton, 1968; Wallace and Freeman, 1987; Wallace, 2005) advocates the model that minimises the description of the model plus the description of the data given the model. Compression is a natural criterion to measure the goodness of models.

Molecular biology data are becoming available at an increasing rate. The competitive challenge is now to extract information from the data more quickly and more accurately. Conventional information extraction methods are often overwhelmed by volume and misled by statistical biases (Allison et al., 1999; Cao et al., 2009b). Information extraction methods based on compression (Dowe et al., 1996; Allison et al., 1998; Stern et al., 2001; Cao et al., 2009b, 2010c) are shown to minimise the effect of statistical biases in

data. Compression is also successfully applied to a wide range of data extraction tasks in bioinformatics ranging from sequence alignment (Allison et al., 1992; Powell et al., 1998a, 2004; Cao et al., 2010c) to pattern discovery (Milosavljevic and Jurka, 1993b; Stern et al., 2001; Cao et al., 2010a) and phylogenetic analysis (Allison and Wallace, 1994; Li et al., 2001; Otu and Sayood, 2003; Cao et al., 2009a). These methods require good compression techniques in order to perform well.

The per-element information content of a sequence using differing contexts (Dix et al., 2007) produced by compression is useful in many bioinformatics applications (Stern et al., 2001). The information content is relative; it depends on the *context* of the compression, i.e., the *background knowledge* upon which the compression is performed. If the context is related to the information contained in the sequence, better compression is achieved than otherwise. The resulting *conditional* information content obtained from the compression of a sequence on the background knowledge of some other sequences, can show the similarities between the sequence being compressed and the background sequences. Section 3.6 shows an application of the per-element information content of sequences obtained by the expert model compression algorithm.

Although genomic sequences can be represented as strings, compressing them is very challenging. General text compression algorithms do not perform well on biological sequences (Cao et al., 2007). A number of special purpose algorithms have been developed for compression of biological sequences (Grumbach and Tahi, 1994; Rivals et al., 1996; Allison et al., 1998, 1999; Apostolico and Lonardi, 2000; Matsumoto et al., 2000; Chen et al., 2000, 2002; Adjeroh et al., 2002; Tabus et al., 2003; Behzadi and Fessant, 2005; Korodi and Tabus, 2005, 2007). However, they either do not perform well or cannot handle sequences longer than a few million symbols. Furthermore, most of them do not produce the estimated per-element information content sequence, hence are not applicable for some information extraction tasks.

Designing adaptive compression models (Rissanen and Langdon, 1981) has been one of the quests of the compression community (Bell et al., 1989; Williams, 1991). It has been formally proved that for any non-adaptive compression model, there always be a more superior adaptive compression model (Cleary and Witten, 1984a). In practice, adaptive compression models overtake non-adaptive counterparts on most types of data such as texts and images (Cleary and Witten, 1984b; Williams, 1991). To the best of the author's knowledge, none of the existing special purpose compression algorithms for biological sequences is adaptive. The designing of an adaptive compression model, hence, should improve the compression performance and will push forward the field of biological sequence compression. The expert model presented in this thesis is an adaptive compression model.

1.1.2 Sequence Analysis

Most existing approaches to sequence analysis, such as those for sequence alignment and phylogenetic analysis, work at the character level; they examine characters (nucleotide bases) on each sequence for comparison. They normally do not perform well on problematic data such as those with statistical biases (Allison et al., 1999). Furthermore, the number of biological sequences collected is increasing rapidly, and the sequences are longer and longer. Developing novel analysis methods that are tolerant to the statistical biases of data and that scale well to the large amount of data is therefore a necessity.

One of the most common tools in comparative genomics is *pairwise local alignment* which finds similarities between a pair of genomic sequences. Existing alignment methods generally attempt to maximise the character matching score in an alignment based on some matching scheme that, for example, gives some positive score for a match and a penalty for a mismatch (Needleman and Wunsch, 1970; Smith and Waterman, 1981). This gives rise to the problem that alignments of unrelated but low information content sequences – such as statistically biased regions – can be given unreasonably high scores because of the abundance of random matches in these regions (Allison et al., 1999). This results in an excess of false positive matches. A partial solution is to "mask out" low information content regions before performing alignment but doing so could miss out some important features that may be present in these low information content regions. The effect could be minimised if the alignment is performed by examining the information content of the two sequences.

Most scoring schemes make the assumption that the score for a match of characters is the same in every position. However, in practice, the information content of each symbol often varies across the sequences (Allison et al., 1998; Cao et al., 2007) and therefore, the scoring scheme must be adapted accordingly. The compression technique developed in this thesis computes per-symbol information content through an adaptive scoring scheme enabling alignment that overcomes the problem.

This thesis takes the approach that since biological sequences carry genetic information, sequence comparison should be done at the information level. Instead of comparing each pair of characters from two sequences to find the matching score, the two characters or sequences should be compared to see how much information is shared between them. Moreover, the conditional information content of a sequence obtained by the compression of the sequence on the background knowledge of another (Dix et al., 2007) can show what the latter sequence tells about the former. An alignment method, namely *XMAligner*, which operates at the information content level, is presented in Chapter 4. Section 4.4 shows that XMAligner produces much less false positives than conventional alignment algorithms in low informative regions and in distantly related sequences.

Existing alignment methods often lack a well-principled objective function. Alignment generally involves a number of parameters and/or some evolutionary models. Without an objective function, it is not easy to select a good model or a good performing set of parameters. As a result, suboptimal solutions are sometimes obtained. Compression offers a natural criterion for selecting models as well as parameters, and hence is potentially useful for fine-tuning the sequence alignment process. Indeed, it is shown to be a plausible objective function for XMAligner.

While the similarity of information content in some regions within two sequences shows the local alignment of the two sequences, the total shared information between two sequences gives an indication of genetic sharing between them. This idea of using the compressibility of sequences to measure genetic distances has been investigated by several existing works (Allison and Wallace, 1994; Li et al., 2001; Otu and Sayood, 2003; Cao et al., 2009a). However, these works lack a formal analysis of the distance measure. The linearity of these distance measures with elapsed time is not guaranteed hence their reliability is still a question. In Chapter 5, a distance measure (XMDistance) is derived from the compressibility of sequences. XMDistance is shown to be approximately proportional to elapsed time in Section 5.4. The distance measure can be applied to inferring phylogenies from whole genomes of several problematic data sets (Section 5.5 and Section 5.6).

1.2 Research Objectives

There are two main aims of this thesis. The first is the development of a biologically related compression algorithm, that can be used effectively by biologists. The compressor needs to be fast and able to compress biological sequences well, i.e., be able to model biological processes. The second aim is to apply the developed compression tool to perform knowledge discovery from biological sequences. This section discusses these objectives in detail.

The first aim of this research is to develop an algorithm for compression of biological sequences. Since compression is closely related to modelling (Rissanen and Langdon, 1981), designing a biological sequence compression algorithm is essentially about modelling the biological processes. The underlying model of the compression algorithm should be a model that (i) relates to biological processes, (ii) has few parameters and is biologically interpretable, (iii) is able to estimate the per-element information content of a sequence, (iv) can utilise different "contexts" (background information), and (v) is computationally efficient to be able to handle long sequences.

Standard file compression algorithms fail to compress biological data at all (Grumbach and Tahi, 1993; Nevill-Manning and Witten, 1999; Cao et al., 2007). Most special purpose compressors treat biological sequences as strings containing approximate repeats (Allison et al., 1998). It is expected that better compression could be obtained with closer modelling of biological processes. The project aims to solve the compression problem by modelling real biological properties, such as approximate repeats involving long-distance similarities. Biological data are difficult to compress for a variety of reasons including the obscurity of significant local lexical and syntactic structure (Nevill-Manning and Witten, 1999), variations and mutations within repeated elements, and the lack of independence between different kinds of data. The model developed in this thesis (see Chapter 3) relates directly to biological processes so that inferred parameters and structures have biological interpretations, in contrast to standard or modified file compression methods (Loewenstern and Yianilos, 1999).

In order to fulfil the information extraction objective, the compression algorithm must be able to calculate the per-symbol information content of a biological sequence. This *information content sequence* allows biologists to examine areas of interest by zooming in and out. The generation of information content sequences is useful for the study of *conditional information content*. Comparing the information content of a sequence X with and without a sequence Y as the context shows what new information Y tells about X that was not already known about X without Y (Stern et al., 2001).

Motivated by the fact that biological sequences are means of carrying genetic information, this research's second objective is to investigate the use of the information content for analysis of sequences, especially long sequences such as genomes. One outcome of the compression is the information content sequence (Dix et al., 2007) which shows the amount of information at every position or every region in a sequence. By examining the information content sequence, one can identify important patterns from biological data (Stern et al., 2001).

Once considered to be "junk DNA", repeat elements are now recognised as "drivers of genome evolution" (Kazazian, 2004). Despite their abundant occurrences in eukaryotic genomes, locating repeat elements in genomes is challenging (Zhi et al., 2006) mainly because they are approximate (i.e., they contain mutations, insertions and deletions) and their occurrences can be far apart. If a pattern is repeated, even with errors, the second occurrence must contain less information than the first occurrence, since some information has been carried in the first occurrence. Therefore, the reduction of information content of a region due to repetition should be reflected in the information content sequence, and should indicate the fidelity of the repeat. Based on the observation, this research aims to develop a method for repeat element detection by examining the information content sequence. The method developed is presented in Section 3.6.

Chapter 4 also investigates the use of information content for sequence alignment. Information content of a sequence is relative; it depends on the background knowledge. The conditional information content of a sequence obtained from the compression of a sequence on the background knowledge of another related sequence differs, in general, from the information content of the sequence compressed alone. Examining the two information content sequences can identify regions where the two sequences share information. The identification of these regions is related to the local alignment of the two sequences.

More specifically, the research examines the premise that the best alignment of two sequences leads to the best compression of the two sequences together. Conversely, the best available compression of any two sequences reveals a plausible alignment of the two sequences. The premise can facilitate the alignment of sequences (see Section 4.3) and other sequence analyses such as estimating the rates of mutations between sequences (see Section 4.6).

The difference in the total conditional information content of a sequence on the background knowledge of another sequence and the total for the sequence alone indicates the amount of information that is shared between the two sequences. This shared information can give clues to the evolutionary history of the sequences. Chapter 5 investigates the use of information content for phylogenetic analysis. In particular, the connection between compressibility and the elapsed time that the two species split from each other is studied in section 5.4. The research derives a genetic distance measure between species from the compressibility of the genetic sequences. Because compression by the expert model is fast and is not dependent on the multiple alignment of sequences, the distance measure can be used to infer the evolutionary history of various species from their genomes (see Section 5.6).

1.3 Contributions

The research presented in this thesis has made several important contributions to the fields of bioinformatics and data compression. These contributions are briefly described below.

Although a number of special purpose algorithms (Grumbach and Tahi, 1993, 1994; Allison et al., 1998; Loewenstern and Yianilos, 1999; Chen et al., 2000; Rivals et al., 1996; Apostolico and Lonardi, 2000; Matsumoto et al., 2000; Chen et al., 2002; Adjeroh et al., 2002; Tabus et al., 2003; Manzini and Rastero, 2004; Korodi and Tabus, 2005; Behzadi and Fessant, 2005; Korodi and Tabus, 2007; Cao et al., 2007) to compress biological sequences have been described in the literature, there has not been a systematic taxonomy of these methods. This thesis presents a categorisation of these algorithms in Subsection 3.3.2. A

detailed analysis of the features that make biological sequences compressible, and the techniques that biological sequence compression algorithms can use to exploit these features are presented in Section 3.3.

In chapter 3, the thesis introduces the expert model (XM) (Cao et al., 2007), an algorithm for the compression of biological sequences. Experiments, presented in Section 3.5, show that the expert model outperforms existing biological sequence compression algorithms on standard benchmarks of both DNA and protein sequences. Not only does the expert model achieve better compression, but it also runs faster than most other algorithms. Moreover, the expert model is the first algorithm reported to be able to handle sequences in length of up to a billion bases on a desktop computer. The superiority of the expert model over other biological sequence compression methods is confirmed in recent reviews on biological sequence compression (Giancarlo et al., 2009; Nalbantoglu et al., 2010).

The expert model uses a novel adaptive technique to combine predictions from different sub-models. The technique is based on the well-founded Bayesian framework. Designing adaptive models has been one of the most pursued directions in data compression (Rissanen and Langdon, 1981; Cleary and Witten, 1984a; Williams, 1991). Not only does the expert model adapt its parameters to fit to the data stream, but it also adaptively adjusts the weight of each sub-model to obtain an optimal blending given the data seen so far. The framework of the algorithm can be generalised to combine different kinds of models, especially models for different data sources. The expert model is the first adaptive compression technique applied to biological data, and adaptive compression methods generally perform better than non-adaptive ones. This explains the superiority of the expert model.

Using the expert model, this research performs a study on the information content of the genomes of several species in various organism levels. These species include viruses, bacteria, archaea, single cell eukaryotes, invertebrates, plants and mammals. The compressibility of these genomes is presented in Subsection 3.5.3. The thesis suggests that, the collection of these genomes should be a benchmark for future biological sequence compression algorithms since the existing benchmark composed by Grumbach and Tahi (1994) does not reflect the diversity and the scale of biological sequences in current and future databases.

The expert model provides several attractive features that are useful for many information extraction tasks. Such an example is the estimation of the information content of every symbol in a sequence using different contexts. In Section 3.6, this thesis shows the use of the information content sequence to locate repeat elements in a sequence. Repeat elements can be recurrent patterns in the sequence or repeat elements from an existing repeat database such as RepBase (Jurka et al., 2005).

This thesis argues that, since genomic sequences are means of carrying genetic information, many sequence analysis tasks can be done at the *information level*. Compression provides a natural criterion for model selection in performing these tasks. The research investigates this approach in two of the main problems in bioinformatics: sequence alignment and phylogenetic analysis.

Chapter 4 of this thesis presents XMAligner, a method for local alignment of genomes based on compression. The method works on the premise that the best alignment of two sequences leads to the best compression of the two sequences together. Unlike conventional alignment approaches that perform alignment at the symbol (character) level, XMAligner aligns sequences at the information level. The compressibility provides a natural objective function for the alignment, which is lacking in most existing alignment methods. XMAligner does not rely on a substitution matrix (a matrix of matching scores) like most other alignment algorithms. A substitution matrix can even be calculated if the sequences are sufficiently long (Cao et al., 2009b). This is presented in Section 4.6.

Experiments show that XMAligner outperforms most conventional character matching alignment methods, especially on problematic data such as distantly related sequences or sequences with statistically biased compositions. The method can be applied for alignment of long sequences such as eukaryotic genomes while requiring only modest hardware. The experiments are described in Section 4.4. The output from XMAligner can be imported into InfoV (Dix et al., 2007) for visualisation. An example of the visualisation is shown in Section 4.5.

Phylogenetic analysis using information content is also investigated in this research (Chapter 5). A distance measure of genetic distances between any pair of sequences, namely *XMDistance* is derived. Section 5.4 presents a formal analysis showing that XMDistance is approximately proportional to the elapsed time, which is a desirable property for phylogenetics. The calculation of XMDistance between two sequences is based on compression of the two sequences, separately and together. This approach does not require an evolutionary model as for most existing phylogenetic analysis methods. XMDistance does not rely on a multiple alignment of the sequences: it can be used for analysis regardless of whether or not the sequences have been aligned. If an alignment of the sequences is available, a simple compression technique based on the Markov model is used. In the case that the sequences are not aligned, or even cannot be reliably aligned as in the case of whole genomes, the expert model is used.

Experiments on simulated data presented in Subsection 5.5.1 show that phylogenetic analysis by XMDistance performs comparably with the best existing approaches. Also XMDistance is much faster and can handle long sequences and large phylogenies. On real data (Subsection 5.5.2), XMDistance is found to infer plausible phylogenies. XMDistance can be successfully applied to phylogenetic analysis from whole genomes, especially on

difficult data such as statistically biased genomes or genomes that contain horizontal gene transfer (Cao et al., 2009a) (Section 5.6).

1.4 A Tour of the Thesis

The rest of the thesis is organised as follows. After this introduction, Chapter 2 presents the background that this thesis is based on and introduces the important concepts used in this thesis. In particular, Section 2.1 describes the processing of genetic information in a living cell. The mechanisms for evolution and diversity of life are also discussed. Section 2.2 introduces the basic concepts of information theory and the MML framework and gives an overview of data compression.

Chapter 3 deals with the compression of biological sequences. It reviews some general applications of compression for solving bioinformatics problems in Section 3.2. Section 3.3 discusses the main features of redundancy in biological sequences and reviews the state of the art algorithms for biological sequence compression. Section 3.4 then presents the expert model and some of its variants. Subsection 3.5.1 and Subsection 3.5.2 show experiments to compare the performance of the expert model against the most effective existing compression algorithms on a standard DNA data set and a standard protein data set respectively. The expert model is then used to compress the 24 human chromosomes – 22 autosomes and two sex chromosomes – and the genomes of various species of varying organism levels. The compression of these genomes is described in Subsection 3.5.3.

The chapter also demonstrates that the expert model can produce the information content of every symbol in the sequence as well as the conditional information content of a sequence on the background of some other sequences. An application of this is the detection of repeat elements in a sequence and is presented in Section 3.6. Section 3.7 presents an asymptotic analysis of the expert model and discusses the advantages of the expert model over other existing biological sequence compression algorithms.

Chapter 4 presents the application of the expert model to one of the most important problems of bioinformatics: alignment of genomes. Instead of comparing sequences at the character level as most existing alignment algorithms, XMAligner aligns sequences at the *information* level. The methodology is based on the premise that the best alignment of two sequences leads to the best compression of the two together. Compression is therefore a natural objective function for alignment and for estimating parameters. The chapter presents experiments to evaluate the performance of XMAligner on an simulated data set (in Subsection 4.4.1), on a set of sequence pairs from the mouse and human genomes (Jareborg et al., 1999) (in Subsection 4.4.2), and on a set of *Plasmodium* genomes (in Subsection 4.4.3). The objective function is also validated experimentally. Visualisation of the alignment using InfoV (Dix et al., 2007) is also illustrated in Section 4.5.

In Section 4.6, the alignment methodology is then extended to computing the substitution matrix between any genomes. Here, the substitution matrix is considered as parameters of the alignment program. An expectation maximisation approach is used to find the matrix that maximises the alignment score, which is measured by the compression objective function. Experiments on simulated data, and on various malaria parasite genomes, which form a difficult case, are presented in Subsection 4.6.2.

Chapter 5 shows how the information content obtained from compression can be used to infer phylogenetic trees. In the chapter, Section 5.4 presents XMDistance, a measure of genetic distance between two sequences based on the compressibility of the two sequences, and the shared information content between them. The proposed distance measure is shown to be approximately proportional to elapsed time given a constant evolutionary rate. This is a desirable property of a distance measure in phylogenetics analysis. The section presents two compression techniques to compute the distance measure in two cases: where the sequences are aligned and where the sequences cannot be reliably aligned, as for genomes.

Section 5.5 presents experiments of phylogenetic analysis using XMDistance. Subsection 5.5.1 describes the comparison of XMDistance to several standard phylogenetic analysis methods including the *maximum parsimony* method (Camin and Sokal, 1965), the *maximum likelihood* method (Felsenstein, 1981) and the standard distance measure using the F84 evolutionary model (Kishino and Hasegawa, 1989; Felsenstein and Churchill, 1996) on a set of simulated data. Subsection 5.5.2 shows experiments of XMDistance on two sets of mitochondrial DNA. Section 5.6 presents the use of XMDistance to infer phylogenies from whole genomes on two problematic data sets: a set of 13 genomes of the γ-Proteobacteria group which is well known for the abundance of horizontal gene transfer, and a set of eight *Plasmodium* genomes which shows variation of evolutionary rates and statistical biases in the sequences.

Each chapter contains its own summary and ideas for future works. Chapter 6 recaps the key contributions of the research along with future directions before concluding the thesis.

1.5 Publications

Parts of material presented in this thesis have previously appeared in the following publications, in chronological order:

- Minh Duc Cao, Trevor I. Dix, Lloyd Allison and Chris Mears, *A Simple Statistical Algorithm for Biological Sequence Compression*, in: IEEE Data Compression Conference, pages 43-52, 2007.

- Minh Duc Cao, Trevor I. Dix and Lloyd Allison, *A Distance Measure for Phylogenetic Analysis of Genomes*, in: The 19th International Conference on Genomic Informatics (Poster), 2008.
- Minh Duc Cao, Trevor I. Dix and Lloyd Allison, *Computing Substitution Matrices for Genomic Comparative Analysis*, in: the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), LNCS 5476, Springer, pages 647-655, 2009.
- Minh Duc Cao, Lloyd Allison and Trevor I. Dix, *A Distance Measure for Genome Phylogenetic Analysis*, in: Australian Conference on Artificial Intelligence, LNCS 5866, Springer, pages 71–80, 2009.
- Minh Duc Cao, Trevor I. Dix and Lloyd Allison, *A Biological Compression Model and its Applications*, book chapter in: Software Tools and Algorithms for Biological Systems, Springer, 2010. In press.
- Minh Duc Cao, Trevor I. Dix and Lloyd Allison, *A Genome Alignment Algorithm Based on Compression*, BMC Bioinformatics, 2010. Submitted.

1.6 Summary

The motivation for the research lies in two main directions: the compression of biological sequences, and the use of compression for biological sequence analyses. The first objective is to design a biological sequence compression algorithm with several desired features that are useful for studies of biological sequences. The second objective is to investigate the use of compression for sequence analysis. The two objectives of the research have been achieved. A compression algorithm, namely expert model (XM) (Cao et al., 2007), has been developed. The expert model is superior to existing counterparts in most aspects: compression performance, speed and scalability (Cao et al., 2007; Giancarlo et al., 2009; Cao et al., 2010a; Nalbantoglu et al., 2010). The research has also applied successfully the expert model to two important research problems in bioinformatics, sequence alignment and phylogenetic analysis.

Chapter 2

Background

In biology, there are no rules without exceptions.
–David B. Searls

Computers are to biology what mathematics is to physics.
– Harold Morowitz

2.1 Biological Background

Biology reached a major milestone in the 1950s when Watson and Crick (1953) discovered DNA's double helix structure and proposed the central dogma (Crick, 1958, 1970). The finding has been the foundation of much of the study of biological processes, especially the elucidation of the processing of genetic information. This section summarises how genetic information is processed in cells.

2.1.1 Molecules of the Cells

Living organisms are those with the ability to reproduce, to metabolise and to adapt to their environment. Reproduction refers to the ability to create new individuals of the same kind, either asexually from a single parent, or sexually from two parent organisms. Living organisms can metabolise; they can capture materials and energy from the environment, and transform these materials into their components to grow and to maintain their living states. Metabolism happens via series of chemical reactions called *metabolic pathways*. Organisms are also able to respond to changes in their environment to maintain their living states and to promote the continuation of their species.

A living organism has a structure of one or more cells where essential biochemical reactions and metabolic activities take place. A new cell arises from a pre-existing cell by cell division. Each cell of an organism stores a copy of the organism's *genetic material*, which is duplicated and passed to new cells during cell division. The genetic material contains the instructions for the cell to perform all biological activities within the cell. The genetic material is passed on to the next generation in reproduction. This genetic material is, therefore, considered to define the organism.

In the current biological classification system (Woese et al., 1990), living organisms are categorised into three domains, namely *eukarya*, *bacteria* and *archaea*. The classification is based on morphology and evolutionary history. Eukarya, sometimes called *eukaryotes* (literally meaning *with nucleus*), are organisms whose cells contain a distinct membrane-bound *nucleus* where the genetic material of the organism is kept. Apart from the nucleus, a eukaryotic cell has some other membrane-bound, specialised subunits called *organelles*. Eukaryotic organisms can be unicellular, as in amoebae, or multicellular, as in plants and animals. On the other hand, bacteria are unicellular organisms. Bacterial cells are also simpler and smaller than eukaryotic cells. They do not have a membrane-bound nucleus or other organelles as do eukaryotic cells. Archaea resemble bacteria in that they are unicellular organisms and their cells do not contain a nucleus or other organelles. They are however different from both bacteria and eukarya in certain aspects of their chemical structure, such as the composition of their cell membranes. Archaea usually live in extreme environments such as hot mineral springs or deep-sea hydrothermal vents. Originally, archaea were considered to be a sub-group of bacteria, namely *archaebacteria*, but recent analyses show that they are genetically and metabolically more closely related to eukarya than to bacteria (Woese et al., 1990). Both archaea and bacteria are regarded as *prokarya* (or *prokaryotes*) in effect meaning *without nucleus*.

Viruses are considered to be "at the edge of life" (Rybicki, 1990) but do not really live. They have no cellular structure and hence do not metabolise. However, they possess genetic material that is made from the same materials as in living organisms, and they can adapt to their environments. They develop inside host cells of living organisms and rely on the biochemical machinery of the host to survive and to reproduce.

Despite the diversity in cell structures and metabolic pathways among species, all living organisms, and some viruses, have their genetic information stored by the same material, *Deoxyribonucleic acid* or *DNA* for short. Some viruses use *Ribonucleic acid* (RNA), a similar material, to store their genetic information. DNA is a nucleic acid which consists of two long polymers (or two strands) with backbones made of sugars and phosphate groups. Attached to each sugar in a backbone is one of four *bases*, namely *adenine* (abbreviated as A), *cytosine* (C), *guanine* (G) and *thymine* (T). Adenine and guanine are chemically similar and are thus classified into one group called *purine* while guanine and thymine are in the *pyrimidine* group. The combination of a base and a sugar/phosphate

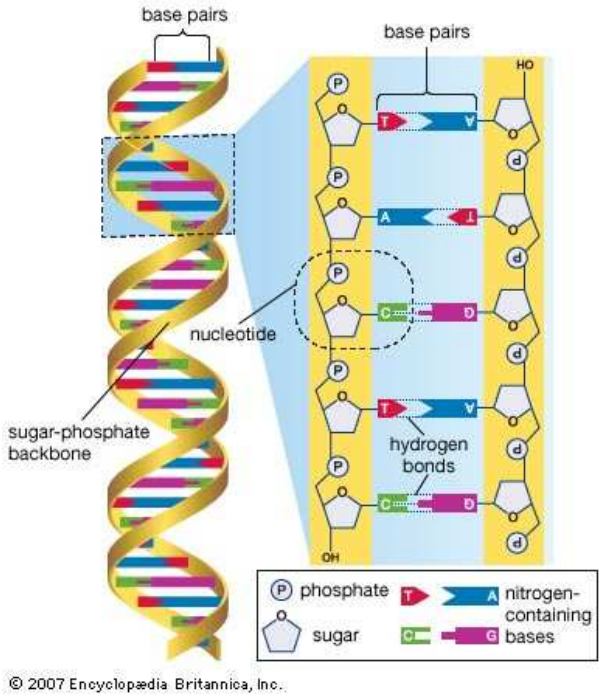


Figure 2.1: The double helix structure of DNA. (By courtesy of Encyclopaedia Britannica, Inc., copyright 2007; used with permission.)

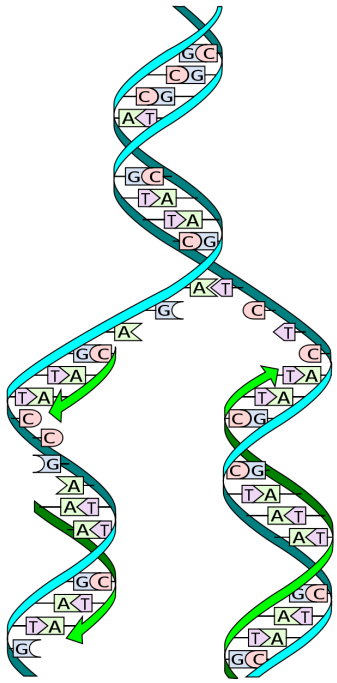


Figure 2.2: DNA Replication. (Source: Wikipedia.)

group makes up a *nucleotide*. The two strands of DNA run in opposite direction to each other. The asymmetric ends of DNA strands are called the 5′ (upstream) and 3′ (downstream) ends. The two strands twist together into a double helix form (Watson and Crick, 1953). The double helix structure is stabilised by hydrogen bonds between *complementary* nucleotides in the two strands; these bonds pair A to T, and C to G. As a result of this pairing mechanism, the information on a strand is duplicated on the complementary strand. The structure of DNA is illustrated in Figure 2.1. The length of a DNA sequence is often measured in the number of bases (or *base pairs*), kilobases (Kb), or megabases (Mb).

The double helix structure of DNA provides a mechanism for the replication of DNA, which takes place at the beginning of the cell division process. During DNA replication, the two strands of the DNA sequence are separated in an unzipped fashion. Each strand is used as a template to create its new complementary strand. An enzyme called *DNA polymerase* performs the replication by finding the complementary base for each base on the strand in turn and bonding it to the base on the template. The process is generally extremely accurate, making less than one error for every 10^7 nucleotides (McCulloch and Kunkel, 2008). Some errors may even be corrected by the DNA polymerase. The replication process produces two DNA sequences nearly identical to the original DNA sequence. The two newly created sequences are the genetic material for the two cells resulting from the cell division process. Figure 2.2 illustrates the replication of a DNA sequence.

The entire hereditary genetic information of an organism is called the *genome* which consists of one or more DNA molecules, each is called a *chromosome*. A prokaryotic genome generally contains one chromosome. The two ends of a prokaryotic chromosome commonly bind together to form a circular chromosome. The genome is stored in a structure called the *nucleoid*. On the other hand, a eukaryotic genome generally consists of multiple linear chromosomes and is stored in the nucleus of the cell. Apart from the nuclear genome, a eukaryotic cell may contain smaller genomic material in other organelles such as *mitochondria* (mitochondria DNA or mtDNA) and *chloroplasts* (chloroplast genome). Each of these genomes, called a *organelle genome*, is often a circular chromosome.

Typically, a sexually reproducing organism is *diploid*, that is the genome has two sets of chromosomes, each obtained from a parent. Each diploidy chromosome has a *centromere*, with one or two arms projecting from it. The two copies of a chromosome in a cell most resemble each other (and hence are called *homologous chromosomes*) and bind to each other at their centromeres. Some organisms, such as some plants and amphibians, have more than two sets of chromosomes and hence are called *polyploid*. Having one set of chromosomes of an organism is called *haploid*.

Another type of material similar to DNA, *ribonucleic acid* (RNA), stores the genetic information of some viruses, and is the carrier of information within a cell. RNA is also a sequence of nucleotides but each nucleotide contains a ribose sugar instead of a deoxyribose as in DNA and the base *uracil*(U) takes the place of thymine (T). RNA is single-stranded but an RNA sequence may contain self-complementary parts that bind to form folded structures.

The primary role of DNA is to carry the instructions to produce *proteins* which are the main actors that are involved in virtually every process within the cell. Proteins include the enzymes that catalyse most of the chemical reactions involved in metabolism and regulate DNA manipulation processes such as DNA replication, DNA repair, and the translation from genetic material to proteins themselves. They provide facilities for communication among cells and among components within a cell. They coordinate basic cellular activities. They form antibodies in the immune system whose function is to bind to antigens and foreign substances in the body, and target them for destruction. They transport substances and energy, to the appropriate locations in the organism body.

A protein is a chain of *amino acids* joined together by peptide bonds. An amino acid in a protein sequence is called a *residue*. There are 20 types of amino acids, each contains an amine group, a carboxylic acid group and a side chain that determines the specific amino acid. The function of a protein is determined by its three-dimensional structure which is also known as its *native conformation* or *tertiary structure*. The conformation of a protein is uniquely determined by the sequence of amino acids which is referred to as the *primary structure* of the protein. Once a protein sequence is formed, it spontaneously folds to its conformation, given a suitable environment. The folding of a protein sequence

Table 2.1: Properties of 20 amino acids.

Amino Acid	3-Letter Code	1-Letter Code	Polarity	Charge	Hydropathy index	Volume
Alanine	Ala	A	nonpolar	neutral	1.8	67
Arginine	Arg	R	polar	positive	-4.5	148
Asparagine	Asn	N	polar	neutral	-3.5	96
Aspartic acid	Asp	D	polar	negative	-3.5	91
Cysteine	Cys	C	nonpolar	neutral	2.5	86
Glutamic acid	Glu	E	polar	negative	-3.5	109
Glutamine	Gln	Q	polar	neutral	-3.5	114
Glycine	Gly	G	nonpolar	neutral	-0.4	48
Histidine	His	H	polar	positive	-3.2	118
Isoleucine	Ile	I	nonpolar	neutral	4.5	124
Leucine	Leu	L	nonpolar	neutral	3.8	124
Lysine	Lys	K	polar	positive	-3.9	135
Methionine	Met	M	nonpolar	neutral	1.9	124
Phenylalanine	Phe	F	nonpolar	neutral	2.8	90
Proline	Pro	P	nonpolar	neutral	-1.6	90
Serine	Ser	S	polar	neutral	-0.8	73
Threonine	Thr	T	polar	neutral	-0.7	93
Tryptophan	Trp	W	nonpolar	neutral	-0.9	163
Tyrosine	Tyr	Y	polar	neutral	-1.3	141
Valine	Val	V	nonpolar	neutral	4.2	105

The chemical properties of 20 amino acids. The hydropathy index of an amino acid shows hydrophobic or hydrophilic properties of its side-chain. The larger the number, the more hydrophobic and the less hydrophilic the amino acid's side chain is. The hydropathy index in this table follows work by Kyte and Doolittle (1982). The volume of an amino acid is in *Van der Waals volume* (Bondi, 1964). Side chain charge is measured at pH 7.4.

is thought to be dependent on the chemical and physical properties of each amino acid in the sequence, such as polarity, being hydrophilic (ability to bond with water) or hydrophobic (water-repellent), electrical charge, and the volume of the side chain of the amino acid. However, although these properties of amino acids are well known, understanding the folding process and predicting the three-dimensional structure of a given protein sequence is still a significant challenge. The names, codes and properties of each amino acid are presented in Table 2.1. The set of all protein sequences created by an organism is called the *proteome*.

2.1.2 The Genetic Code and the Central Dogma

If proteins provide the machinery to perform most of the operations within a cell then the genome is the controller of these machines. Not only does the genome specify the primary structure of each protein, and thence the conformation and the function of the

protein, but it also regulates the production of the protein. The genome contains information to create various RNA molecules that are involved in protein synthesis. It ultimately controls the expression of each protein, that is it specifies which proteins are to be produced in a particular cell at a particular time. The biosynthesis of a protein from the information in the genome consists of three main stages, namely *transcription*, *splicing* (i.e., *post-transcriptional modification*), and *translation*, as illustrated in Figure 2.3 and described in the text below.

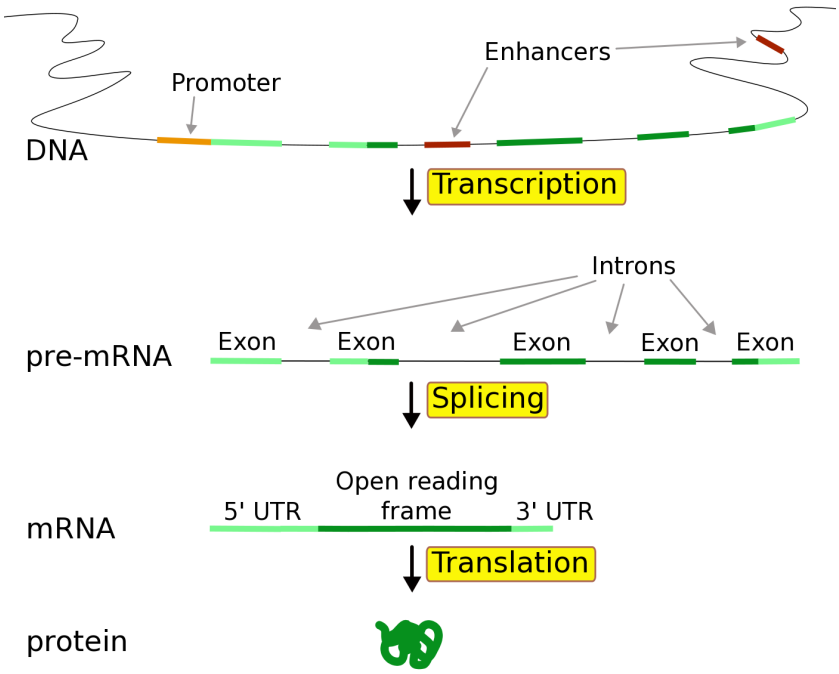


Figure 2.3: Biosysthesis of a protein. (Source: Wikipedia.)

A segment in the genome that stores the genetic information to produce a *messenger* RNA (mRNA) molecule and thence a protein is called a *gene*. A gene is a "unit of heredity." The complete set of genes in an organism, its *genotype*, specifies the characteristics of the organism. Since DNA and RNA use essentially the same "language", that is nucleotides, the process of producing RNA from DNA is called *transcription*. The process of using the genetic information in a gene to synthesise a protein is known as *gene expression*.

Messenger RNA molecules are intermediaries in the process of protein synthesis, but other RNAs also have functional roles, such as *transfer RNA* involved in transportation of amino acids, and *micro RNA* (miRNA) involved in the regulation of gene expression. These functional RNA molecules are called *non-coding RNA*, and are transcribed from DNA. The segments coding for these molecules in the genome are called *RNA genes*.

From the 5′ end, a gene starts with a regulatory region, called the *promoter*, which facilitates the transcription of the gene. A gene may have other regulatory regions, such

as *enhancers*, to strengthen the signal from the promoter. The transcription process starts when an enzyme called *RNA polymerase* recognises and binds to the promoter of the gene. The RNA polymerase then unwinds the two DNA strands, and sequentially hooks together the complementary RNA nucleotide for each base on the DNA template from the non-coding (3′) DNA strand. The resulting RNA sequence is an exact copy of the coding (5′) strand, except that thymine is replaced with uracil. Transcription proceeds until the RNA polymerase encounters a *terminator* DNA sequence, which effectively signals the end of the gene. Transcription in eukaryotic cells occurs in the nucleus of the cell. Transcription of all kinds of RNA is generally the same.

Messenger RNAs in eukaryotic cells are often modified before being translated into protein. Only certain segments in a eukaryotic mRNA may code for protein. A coding segment is called an *exon* (standing for "expressed region") and a non-coding region between exons is called an *intron* (for "intragenic region"). The RNA polymerase transcribes both exons and introns from DNA, resulting in an immature mRNA, also called *pre-mRNA*. The pre-mRNA then undergoes a process called *splicing* where introns are removed and the remaining exons are connected to form the *mature mRNA*, which contains a contiguous mRNA sequence coding for a protein. A gene may have its pre-mRNA spliced in several ways (alternative splicing) and hence can give rise to two or more proteins. Generally, prokaryotic genes do not contain introns and thus their mRNAs are mature upon transcription.

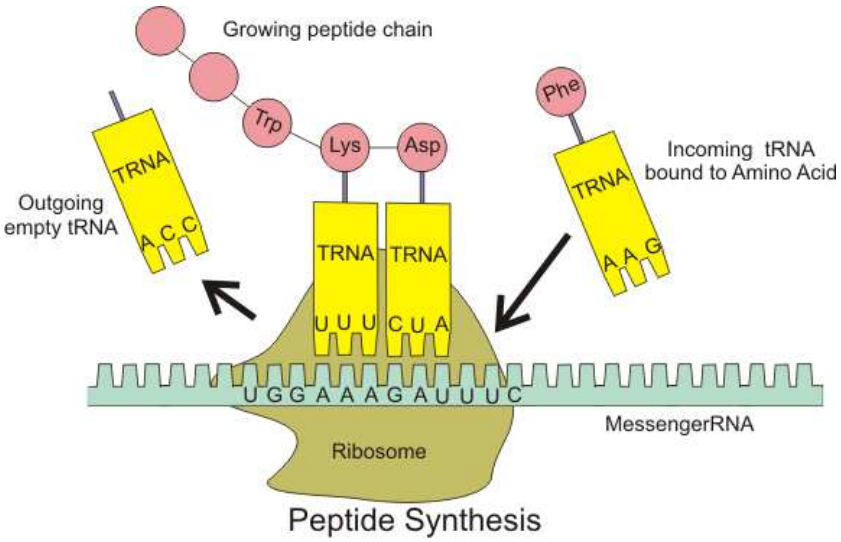


Figure 2.4: The translation process. (Source: Wikipedia.)

Each mature mRNA is brought to the *cytoplasm* where a *ribosome* interprets the genetic message in the sequence and builds a protein accordingly. The message is a series of RNA triplets that are called *codons*. The set of rules for the translation of each of the 64 codons to the appropriate amino acid is called the *genetic code* and is presented in Table 2.2. Since there are 64 possible triplets, i.e., 4^3, there are 64 codons but only 20 types of

Table 2.2: The dictionary of the genetic code.

First base		Second base: U (pyrimidine)		Second base: C (pyrimidine)		Second base: A (purine)		Second base: G (purine)		Third base
	U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
		UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys	C
		UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop	A
		UUG	Leu	UCG	Ser	UAG	Stop	UGG	Trp	G
	C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
		CUC	Leu	CCC	Pro	CAC	His	CGC	Arg	C
		CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	A
		CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	G
	A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
		AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	C
		AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	A
		AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg	G
	G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
		GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly	C
		GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly	A
		GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly	G

amino acids exist. It is often the case that several codons code for one amino acid. For example, both UUU and UUC code for the amino acid phenylalanine (Phe), and six codons (UUA, UUG, CUU, CUC, CUA and CUG) code for leucine (Leu). Apart from three codons UAA, UAG and UGA, which signal the end of the coding region and thus are named *stop codons* (or *non-sense codons*), each of the other codons (*sense codons*) codes for an amino acid. The codon AUG codes for methionine (Met), and is also the indicator for the beginning of the coding region – the start codon – but can occur elsewhere.

For a particular mRNA sequence, there are three possible ways to translate it to protein, depending on the offset, modulo 3, of the starting nucleotide, 0, $+1$ or $+2$. Any such contiguous set of codons is called a *reading frame*. An *open reading frame* (ORF) is a reading frame that contains a start codon at the beginning and a stop codon at its end. For a double-stranded DNA molecule, there are six possible reading frames, three on each strand. This makes overlapping genes possible; one in each reading frame. Indeed some viruses, such as *hepatitis B*, employ such a very compact genetic encoding.

In the cytoplasm, there is a pool of amino acids in the 20 different types. An amino acid binds to a tRNA molecule which brings the amino acid into the ribosome. At one end, a tRNA molecule has an *anticodon* nucleotide triplet that pairs with its complementary codon on the mRNA. For example the anticodon AAA pairs with UUU due to the A-U and C-G complementary pairing for RNA, and UUU codes for the amino acid phenylalanine. At its other end, the tRNA molecule can bind with a particular type of amino acids. For example, the tRNA with an anticodon AAA can only bind to a phenylalanine amino acid. Hence, this tRNA in effect translates a UUU codon on the mRNA to the

amino acid phenylalanine which is then appended to the growing protein sequence. The translation process is illustrated in Figure 2.4.

The translation process starts when the ribosome encounters the first start codon on an mRNA. The mRNA moves through the ribosome and brings the next codon into the ribosome. The corresponding amino acid is then added to the protein while the next codon is brought into the ribosome. The translation proceeds until the ribosome reaches a stop codon whereupon the protein sequence is released from the ribosome. During and immediately after being synthesised, a protein sequence folds to its conformation which is essential to its function. The protein is then transported to the appropriate location to perform its function.

In 1956 Crick coined the term *central dogma* (Crick, 1958, 1970), which deals with the transfer of genetic information stored in the form of macromolecular sequences. Basically, the central dogma describes the flow of genetic information as in Figure 2.5.

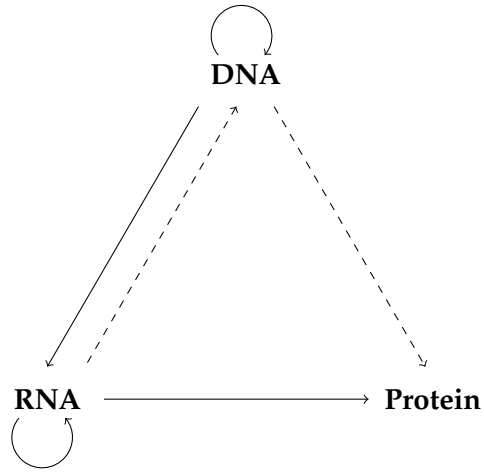


Figure 2.5: Flow of information in the central dogma.

In this figure, the arrows show the directions of information transfer. Solid arrows represent general transfers and dotted arrows show special transfers. Genetic information can be transferred from DNA to DNA via DNA replication. This process involves proteins but these proteins (i.e., enzymes) only provide machinery and structural support in a non-template manner. Likewise, the transcription process transfers information from DNA to RNA, which acts as carriers to bring coded information to ribosomes. During translation, the ribosomes interpret this information and use it to produce proteins. The transfer of genetic information from RNA to RNA occurs in some viruses with RNA genomes during genome replication. The central dogma was proposed in the period that molecular biology was not yet established. The proposed direction of information transfer from DNA to protein was based on a suggestion from Gamow (Segre, 2000) before the discovery of transcription and translation. There is no evidence thus far about the direct

transfer of information from DNA to protein except in a special free-living system (McCarthy and Holland, 1965). The predicted transfer from RNA to DNA, however, was later confirmed by the discovery of viral infection. The main point of the central dogma is, in Crick's words, "once information has got into a protein, it can't get out again", and still underpins biology research today.

2.1.3 Mechanisms of Evolution

Despite the diversity of species in the world, all organisms are believed to be descendants of a common ancestor living about 3.5 billion years ago (Maher and Stevenson, 1988). The variety of organisms on earth is due to accumulated changes in the inherited traits over successive generations. The theory is supported by substantial evidence, and has been central to most biological studies. This section discusses the evolution of species from the point of view of molecular biology.

Evolution of species is the result of the interaction of two processes. The first process is the introduction of variation in organism characteristics that are inherited by successive generations. The second process is natural selection, which makes a particular variant become more prevalent if the variant enhances the organism's survival and reproduction in a specific environment.

Since every organism is virtually defined by its genome, the main cause of variation is changes in genomes. Specifically, changes within genes can cause variations in the resulting proteins which in turn affect particular characteristics of the organism. Different variants, at a specific location (or *locus*), in the genome of a species are called *alleles*. A distinct variant of a characteristic of an organism is a *trait* and the set of observable traits that make up the structure and behaviour of an organism is called its *phenotype*. The replication of DNA during reproduction makes certain inheritable traits to be passed from one generation to the next. The main causes of genome changes are *mutation* and *recombination*.

Changes to genomes in the form of *mutations* occur as a result of errors in DNA replication and of damage to DNA by various factors. Although the process of DNA replication is highly accurate, errors sometimes creep in. An organism inherits the erroneous genome from its parents and passes the genome to the next generation. Collectively after many generations, the number of errors becomes substantial. Environmental factors such as ultraviolet and ionising radiations, mutagenic chemicals and viral infections can also cause damages to DNA. Some viruses insert their genomes into the genome of a host organism causing the host DNA to change. These spontaneous mutations, if occurring in the sex cells and not causing the termination of the organism, have the possibility of being inherited by the next generation.

Some types of DNA mutation involve the change of one nucleotide. The so-called *point mutations* or *substitutions* are caused by the replacement of one nucleotide by another. Certain substitutions to a codon in coding DNA, often to the third position of the codon, do not change the coded amino acid coded, and hence do not affect the resulting protein. Such a substitution is called a *synonymous* or *silent* mutation. These mutations do not cause a change of the phenotype of the organism. A substitution that causes a change in the corresponding amino acid is called a *non-synonymous* mutation. Not only does a non-synonymous mutation alter the coded amino acid in the protein, but it can also result in a stop codon which causes the premature termination in the production of the protein. A substitution that changes a purine (A and G) to pyrimidine (C and T) or vice versa is called a *transversion* while the substitution of a purine by a purine or a pyrimidine by a pyrimidine is called a *transition*. Since intragroup nucleotides (i.e., purine or pyrimidine) are more chemically similar than intergroup nucleotides, transitions tend to cause less severe damage than transversions, and thus are more viable and more likely to be retained. As a result, transitions are known to occur more frequently than transversions in most DNA sequences, even though there are twice as many types of transversions (A↔C, A↔T, G↔C and G↔T) than types of transitions (A↔G and C↔T).

Other point-mutations include those that lead to the *insertion* and *deletion* of nucleotides. These are collectively called *indels*. These mutations significantly modify the signal in a DNA sequence. For example, an indel in a gene may alter splicing of the mRNA (*splice site mutation*) or cause a shift in the reading frame (*frame-shift mutation*). Such changes often result in major modification in the protein, and are the causes of many genetic diseases (Ogura et al., 2001; Baase et al., 2009).

Mutations can also occur at the chromosomal level, which results in changes to the chromosome structure. Some mutations cause the breaking and rearrangement of a segment of DNA within a chromosome, some broken segment may even join another chromosome. Such a mutation is called *translocation*. On some occasions, mutation may reverse the orientation of the segment, which results in a chromosome *inversion*. Other mutations lead to the deletion or duplication of large regions in the chromosome. If these regions contain genes, such mutations respectively cause the loss or the redundancy of the genes located within them. In some cases, mutations at this level may result in the breaking of a chromosome into two, or the fusion of several chromosomes into one. For example, the fusion of two ancestral chromosomes in an ape species is believed to have led to the divergence of human species, *Homo sapiens* from other primates (Hillier et al., 2005). Chromosomal mutations may also change the number of chromosomes in some polyploidy species, resulting in an abnormal number of chromosomes (*aneuploidy*). An example of this is the redundant copy of chromosome 21 in humans which causes Down syndrome.

Another mechanism for producing variation of traits is genetic *recombination* where a DNA molecule is broken and joined to another DNA molecule. Recombination between similar sequences such as copies of the same chromosome is called *homologous recombination*. In sexual reproduction in eukaryotes, recombination occurring during the forming of reproductive cells results in *chromosomal crossover* where the offspring obtains random combinations of genes from its parents and thus is different to each of its parents. Although genetic recombination does not change individual genes, recombination in sexual reproduction breaks up allele combinations and thus allows the removal of harmful mutations and the retention of beneficial mutations (Otto, 2003). Furthermore, recombination can produce individuals with new and advantageous gene combinations. In prokaryotes, homologous recombination allows an organism to incorporate foreign DNA from another similar organism without being the offspring of that organism. This process is called *horizontal gene transfer* or *lateral transfer* (Thomas and Nielsen, 2005).

The three aspects namely variations in genetic material (i.e., genotypes), *inheritance* and *natural selection* make evolution possible. Natural genetic variation means that different organisms have different characteristics, and some of these characteristics can enhance the chances of survival and reproduction of some groups of organisms. Since the world is limited in resources and organisms often reproduce more offspring than their environment can support, natural selection allows organisms with advantageous characteristics to survive and reproduce more successfully. Through inheritance, reproductive advantages are passed on to the offspring over many generations. This leads to the dominance of certain traits in the population.

2.1.4 Genome – Structure, Diversity and Size

The genomes of different species vary greatly in their organisation and structure. Most prokaryotic cells have a single circular chromosome, but some have several chromosomes or a single linear chromosome, or even multiple linear chromosomes. On the other hand, eukaryotic cells always have multiple linear chromosomes. The number of chromosomes in a eukaryotic cell generally ranges from 2 to under 50, but there are exceptions which contain thousands of chromosomes such the genus *Ophioglossum* (Raven et al., 2005). Prokaryotes typically are *haploid*, that is their cells contain one set of chromosome(s), while most eukaryotes are diploid (two sets of chromosomes). Some eukaryotes are haploid (one set) or polyploid (more than two sets) and some species even have thousands of copies of each chromosome in their cell (Watson et al., 2008).

Genome sizes also vary enormously between prokaryotic and eukaryotic species. In this context, the genome size refers to the size of one set of haploid complement chromosomes since the copies of a chromosome in a cell are nearly identical. Typically, prokaryotic genomes are smaller than 10 Mb while most eukaryotic genomes can be as small as

10 Mb (some fungi) and can be as large as 670,000 Mb (amoeboid) as shown in Table 2.3. Even organisms with similar properties have very different genome sizes. For example, the rice genome is about 40 times smaller than the wheat genome (Arumuganathan and Earle, 1991). Diversity is also found in the number of genes in the genomes of different species. Table 2.3 shows the genome sizes and the numbers of genes in several species' genomes.

Table 2.3: Approximate genome sizes, gene numbers and gene densities of various organisms (data taken from (Watson et al., 2008) and (McGrath and Katz, 2004)).

Species	Genome size (Mb)	Gene number	Gene density (Genes/Mb)
Bacteria			
Mycoplasma genitalium	0.58	500	860
Streptococcus pneumoniae	2.2	2,300	1,060
Escherichia coli K-12	4.6	4,400	950
Agrobacterium tumefaciens	5.7	5,400	960
Sinorhizobium meliloti	6.7	6,200	930
Eukaria			
Saccharomyces cerevisiae	12	5,800	480
Schizosaccharomyces pombe	12	4,900	410
Tetrahymena thermophila	125	27,000	220
Caenorhabditis briggsae	103	20,000	190
Drosophila melanogaster	180	14,700	82
Ciona intestinalis	160	16,000	100
Locusta migratoria	5,000	–	–
Fugu rubripes (puffer fish)	393	22,000	56
Homo sapiens (human)	3,200	20,000	6.25
Mus musculus (mouse)	2,600	22,000	8.5
Arabidopsis thaliana	120	26,500	220
Oryza sativa (rice)	430	45,000	100
Zea mayes (corn)	2,200	45,000	20
Triticum aestivum (wheat)	16,000	–	–
Fritilaria assyriaca (tulip)	120,000	–	–
Polychaos dubium (amoeboid)	670,000	–	–

It has been suggested that there is a correlation between the genome size and the complexity of an organism (Vendrely and Vendrely, 1948). Though this observation is reasonable to some extent, such as eukaryotic genomes typically being larger than prokaryotic genomes, it is found to be not true. For instance, the genome of the single cell amoeboid *Polychaos dubium* is more than 200-fold larger than the human genome, while it is clearly less complex than human. The genome sizes are not even proportional to the numbers of genes. The genome of the puffer fish *Fugu rubripes* is eight times smaller than the

human genome while the two species have similar numbers of genes. The extensive variation in nuclear genome size among species is known as the *C-value paradox* (Thomas, 1971).

The C-value paradox was resolved with the discovery of *non-coding DNA* – parts of the genome that do not code for proteins. In a prokaryotic species, the majority of the genome codes for proteins. In *E. coli*, for example, only a few hundred of base pairs of the 4.6 Mb genome are non-coding DNA, and the majority of them are dedicated to regulating gene transcription (Watson et al., 2008). On the other hand, only a small portion of a eukaryotic genome codes for proteins. It is estimated that coding regions make up only 1.5% of the human genome (Lander et al., 2001). Non-coding DNA are sequences between genes (*intergenic regions*) and sequences interspersed between protein coding regions within genes (intragenic regions or *introns*).

Introns are regions within a eukaryotic gene that are transcribed to immature mRNA, but are removed from the mRNA during the splicing process. Parts of an intron are signals for splicing such as *acceptor* and *donor* sites at the two ends of the intron. Some introns are known to enhance the expression of the gene that they are contained in by a process known as intron-mediated enhancement (Mascarenhas et al., 1990). Introns are important for alternative splicing which results in multiple possible proteins from a single gene. The numbers and the lengths of introns vary considerably among species and among genes within the same organism. Prokaryotic mRNAs do not contain introns. Simple eukaryotes have few short introns in a gene. For example, only 3.5% of *Saccharomyces cerevisiae* genes have introns and their introns are generally shorter than 1 kilobase. On the other hand, most human genes have introns, and introns make up 95% of these genes (Watson et al., 2008).

Regions in the genome that do not contain genes are called intergenic regions. Intergenic regions make up 60% of the human genome. Some sequences in intergenic regions are functional. These functional sequences include regulatory sequences such as promoters and enhancers which control gene expression. Other non-coding DNA sequences are transcribed to *functional non-coding RNA* that are not translated to proteins but are generally involved in the translation of mRNA to proteins. Examples of non-coding RNA include ribosomal RNA, transfer RNA and microRNA. Intergenic regions also contain *pseudogenes* which are related to known genes, but have lost their protein-coding ability and are no longer expressed in the cell. The functions of the remaining intergenic regions and of most introns have not been identified and they are often arguably called *junk DNA* (Ohno, 1972).

A large proportion of DNA in eukaryotic genomes is composed of *repetitive DNA* which are sequences that can amplify themselves. About 45% of the human genome is made up of repetitive DNA (Lander et al., 2001). Repetitive DNA sequences can be broadly categorised into two classes (Jurka, 2003; Berg and Howe, 1989), namely *tandem*

repeats and *interspersed repeats*. A tandem repeat is two or more short repeated DNA sequences residing adjacent to each other. Their length ranges from 5-15 bp in *microsatellites*, to 10-100 bp in *minisatellites*. Tandem repeats typically appear in introns.

Interspersed repeat elements are generally longer than tandem repeats. They normally increase their copy number by copying themselves to different positions in the genome. They are often created in DNA by being copied into RNA and reverse transcribed back to DNA (*retrotransposon* elements) or by being copied from other parts of the genome by *transposase enzymes* (*transposon* elements). Nucleotides in repeat elements can be changed, inserted or deleted during these *transposition* events. The class of interspersed repeat elements is subdivided into two smaller classes: *long interspersed nuclear elements (LINEs)* and *short interspersed nuclear elements (SINEs)*.

LINEs are long repeat DNA sequences that range in length from a few hundred to as many as 9,000 base pairs. A LINE element generally codes for several proteins, one of which is the *reverse transcriptase*. Once the RNAs transcribed from a LINE element have been translated to proteins, the reverse transcriptase copies the RNA molecules back into the DNA and forms a new LINE element in the genome. Because LINEs move by copying themselves, they enlarge the genome. The human genome, for example, contains about 900,000 LINEs, which make up roughly 21% of the genome (Lander et al., 2001).

SINEs are shorter DNA sequences (< 500 bases) that represent reverse-transcribed RNA. SINEs do not encode a functional reverse transcriptase protein and rely on other mobile elements for transposition. The most common SINEs in primates' genomes are Alus, which are about 300 base pairs long. They do not contain any coding sequences, and can be recognised by the restriction enzyme Alu, hence the name. With about 1.5 million copies, SINEs make up about 13% of the human genome (Lander et al., 2001). While previously believed to be "junk DNA", recent research suggests that both LINEs and SINEs have a significant role in gene evolution, structure and transcription levels (Hess et al., 1983; Kazazian, 2004).

2.2 Information Theory and Inference

This section presents the basic concepts of information theory and its application to inductive inference.

2.2.1 Probability and Information

Probability theory has been the subject of study since as early as the seventeenth century following the study of games of chance (Arnauld and Baynes, 1962; Bayes, 1763; Laplace,

1814). However it was not properly formulated until the twentieth century when Kolmogorov (1933) proposed his axiomatisation which has become the foundation of probability theory. For a sample space of all possible outcomes denoted $\Omega = \{x_1, x_2...\}$, the function giving the probability of an outcome x, denoted $Pr(x)$, must satisfy the following axioms:

- $0 \leq Pr(x) \leq 1$ for all $x \in \Omega$
- $\sum_{x \in \Omega} Pr(x) = 1$
- $Pr(x_i \cup x_j) = Pr(x_i) + Pr(x_j)$, if two outcomes x_i and x_j are mutually exclusive, that is $x_i \cap x_j = \emptyset$

There are two main schools of interpretation of the concept of probability. The *frequentist* school holds to *frequency probability* (also called *physical* or *objective* probability), being the relative frequency of an event occurring over an infinite number of trials. Under this view, each event is assumed to be governed by some random physical phenomenon, which can be estimated with sufficient information. The limitation of the frequency interpretation is that it is impossible to perform an infinity of trials of an experiment to determine the probability of an event. Some experiments can be performed once only (such as it rains tomorrow) or not at all (such as the cause of dinosaur extinction).

The second, *Bayesian* school of probability interpretation holds to *Bayesian probability* (*subjective probability*), viewing probability as the *degree of belief* of an individual assessing the uncertainty of a particular event. The individual has some belief about the event before seeing any evidence. The degree of belief in this case is called the *prior probability*. The individual can update her or his belief in the light of new relevant data to get the *posterior probability*. This is formalised in Bayes's famous theorem (Bayes, 1763):

$$Pr(h|D) = \frac{Pr(D|h)Pr(h)}{Pr(D)} \tag{2.1}$$

in which, $Pr(h|D)$ is the posterior probability of a hypothesis h after some data D are observed, $Pr(D|h)$ is the probability of observing data D if the hypothesis h is true, $Pr(h)$ is the prior probability of h before seeing D, and $Pr(D)$ is the probability of D under all possible hypotheses. This thesis adopts the Bayesian interpretation of probability.

The outcome of an event adds something to the observer's knowledge; it contributes a reduction in uncertainty in a human mind or changes the state of a system. The outcome is said to carry some *information*. The occurrence of an obvious event brings little surprise to the observer and hence carries little information. On the other hand, a rare event has a greater effect on the observer. The amount of information carried by an event decreases as its probability increases.

Information is not directly linked to any physical quality and hence cannot be measured with an instrument. A decreasing function of probability is required to quantify the amount of information associated with an event. As pointed out by Shannon (1948), any decreasing monotonic function can be used as a measure of information, but the logarithm function is the most natural choice (Hartley, 1928). The Shannon information content of an outcome x is defined to be the negative logarithm of the probability of its happening:

$$\mathcal{I}(x) = -log_2 Pr(x) \tag{2.2}$$

In this function, the logarithm is to base 2, and the information content is measured in *bits* (for binary units). Other choices of base can be used. For example, base 10, which corresponds to the information unit *ban*, was used by Turing to measure the amount of information deduced by his codebreakers (Good, 1979). Boulton and Wallace (1970) measured information in *nats* (originally *nits*) for natural logarithms (base e); this is often mathematically convenient.

Communication of information generally involves the transfer of some physical material or some form of energy from a *sender* to a *receiver*. At an abstract level, information can be considered to be represented by a sequence of *symbols* drawn from a discrete set called an *alphabet*. Both the sender and the receiver in the communication agree on a coding scheme to represent the information. The sequence of symbols under a certain coding scheme is called a *message*. For example, DNA is used as the medium to transmit genetic information from a cell to a daughter cell. The message in this transmission consists of a sequence over the alphabet {A, C, G, T}. The same information is perceived by the SOLiD sequencing system (Pandey et al., 2008) as a sequence over the colour space ({red, green, blue, yellow}) in which each data point represents two adjacent nucleotides, and each nucleotide is interrogated twice. The representation using colour space can be transformed to the sequence of bases (Ondov et al., 2008). One can even conveniently represent the information by a binary message (over the alphabet {0,1}) in which each nucleotide is represented by two bits (such as using a coding scheme that codes the nucleotides adenine, cytosine, guanine and thymine by the bit patterns 00, 01, 10 and 11 respectively).

Sometimes the representation of information contains *redundancy*. For example, if the distribution of a nucleotide sequence is skewed (e.g., the frequencies of A, C, G and T are $1/2$, $1/4$, $1/8$ and $1/8$ respectively) the 2 bits per symbol encoding scheme mentioned above is not as efficient as another code that gives a shorter codeword for a more frequent symbol (i.e., A) and a longer codeword for a less frequent symbol (i.e., T). In information theory, the coded message must be *optimal* and must not contain any *redundancy*. Hence the message will be as short as possible. The length of such a binary message encoding a sequence is equal to the amount of information in the message.

$$\mathcal{L}(S) = \mathcal{I}(S) = -log_2 Pr(S) \tag{2.3}$$

In the above example, if the probability of each symbol is independent in the position of the symbol, the amount of information content of each symbol A, C, G and T is 1 bit, 2 bits, 3 bits and 3 bits respectively. An optimal coding scheme would give the respective codeword length for each symbol (i.e., 0 codes for A, 10 codes for C, 110 codes for G and 111 codes for T).

In transmitting a message containing some information, it is preferable to choose the most compact representation of the information all other things being equal. However, the sender and receiver cannot just select the optimal coding scheme for one particular piece of information because the information is not known when they negotiate the choice of codes. They must estimate the probabilities of possible messages based on some grounds, and agree on a representation scheme that minimises the *expected* length of a message:

$$E(\mathcal{L}(S)) = -\sum Pr(S)log_2Pr(S) \tag{2.4}$$

The lower bound on the expected length of a message generated by a data source is known as the *entropy* of the source. The expected length of encoding one symbol is called the *entropy rate* of the source.

The length of the optimal message representing the information of an event, which is equal to the amount of information of the event, depends on the probability of the event being emitted by its source, and hence on the nature of the source. If the statistical nature of the source is known, the probability of the event can be computed. In the above example, if the probabilities of the events of the occurrences of these symbols are known to both the sender and the receiver, the mentioned optimal coding scheme can be used. However, the statistical nature of many sources such as that generating DNA sequences is not fully understood. For such a source, a model is required to estimate the probability of the occurrence of a symbol generated by the source.

The information content conveyed in an event (or a sequence of events) is the negative logarithm of the probability of its being emitted. Since probability is subjective, the information content of a message is relative; it depends on what is known to both the sender and the receiver. This common knowledge is called the *background knowledge* of the communication. If part of the information in the message is already available to both the sender and the receiver, the information content of the message is less than otherwise. In the above example, if the probability of each symbol is known, the sender and the receiver can agree on an optimal code for transmission. However, if the probability distribution of symbols is not available to both parties, the sender can estimate the probability distribution, and then transmit the statement of the distribution along with the coding of the sequence. This results in a longer message than in the case that the probability distribution is included in the background knowledge.

Background knowledge can also be built up from other events. Consider the example of transmitting a sequence S using an optimal code and thus the length of the message in the transmission is the amount of information $\mathcal{I}(S)$ contained in S. Now suppose that prior to the transmission of S, the sender and the receiver exchanged another sequence T which is *related* to S, and hence covers some of the information in S. The cost of transmitting S is now reduced due to the change in background knowledge. The information content conveyed by S in this case is called the *conditional information content* of S on the background knowledge of T (or given T) and is denoted by $\mathcal{I}(S|T)$. The reduction in information content indicates the amount of information *shared* between S and T. This shared information is called *mutual information*, denoted $\mathcal{I}(S;T)$. It has been formally proved (Cover and Thomas, 1991) that the mutual information of two messages S and T is equal to the reduction:

$$\mathcal{I}(S;T) = \mathcal{I}(S) - \mathcal{I}(S|T) \tag{2.5}$$

2.2.2 Statistical Inference and Minimum Message Length

With the availability of data, one wishes to infer the properties of the source that generated the data. *Inductive inference* is the process of drawing conclusions about the data source from observations of data. For many data sources, and notably biological processes, the generation of data involves random variation. That is, identical repetitions of an experiment do not necessarily produce the same data. Inductive inference on these data sources requires the use of probability and statistics, and is referred to as *statistical inference*.

In statistical inference, the observer of data generally makes assumptions about the generation of the observed data. The mathematical formulation of these assumptions makes up a *statistical model* of the data source. The outcome of the statistical inference can be the rejection or acceptance of a model, an indication of how well a model describes the data, or a set of parameters that best fits the model to the data observer. The outcome is interpreted as an "understanding" of the data source for further decisions on the data.

Since statistical inference is built upon a foundation of probability and statistics, different interpretations of probability lead to different paradigms of statistical inference. *Bayesian inference* applies the Bayesian view of probability. A hypothesis is initially assigned a *prior* probability which is then revised with the observation of data. The resulting *posterior* probability of the hypothesis is then used as the basis for making statistical propositions. Bayesian inference is encapsulated in Bayes's theorem (Equation 2.1).

The *Minimum Message Length* (MML) principle devised by Wallace and Boulton (1968) is an information-theoretic approach to inductive inference. MML is a formalisation of

Ockham's Razor which states that, other things being equal, one should favour the simplest solution. The principle combines information theory with Bayes's theorem to trade a model's complexity against its goodness of fit to data. The MML objective function is the length of a message which contains (i) a statement of a model and (ii) a concise encoding of the data using that model. The goodness of a model can be evaluated from the length of the message. The most probable hypothesis is the one that gives the shortest message length. The message must contain two parts: the first part codes the model, and the second part codes the data assuming that the model is true.

MML is related to the concept of Kolmogorov complexity (Solomonoff, 1964; Kolmogorov, 1965; Martin-Lof, 1966; Chaitin, 1966) and Minimum Description Length (MDL) (Rissanen, 1978). MML differs from Kolmogorov complexity in offering a *practical* computation of the goodness of a model through compression. In contrast to MDL which searches for a model-class, MML advocates selecting a fully parameterised model where parameters are stated to *optimal* precision (Baxter and Oliver, 1994).

2.2.3 Compression: Coding and Modelling

As discussed previously, the entropy of a data source is the lower bound on the expected length of a message by any practical representation of information. Hence, data compression, which attempts to remove any redundancies in data, is useful for the study of the data source. This is the main motivation for this research.

Recent advances view compression as a two stage process: modelling and coding (Rissanen and Langdon, 1981). Modelling involves studying the characteristics of the data source to estimate the probability distribution of messages. In coding, the transmitter transforms a message into a sequence of bits with regard to its probability under the distribution. A similar process is performed by the receiver. The same probability distribution is obtained by the receiver using an identical modelling stage. Using this distribution, the bit sequence is decoded to get the original message.

Optimal coding is achieved when the length of the code assigned to a symbol is the negative logarithm of the probability of the symbol (Shannon, 1948). *Huffman coding* (Huffman, 1952) is one of the best known coding techniques. It constructs a code table for symbols in the alphabet given their probabilities. Huffman coding is guaranteed to use the smallest *integer* number of bits to encode a symbol. Because Huffman coding codes each symbol independently, if the optimal code length of a symbol is not an integer number of bits, it is rounded up. This sub-optimality of Huffman coding becomes more noticeable the longer the sequence being encoded. In particular, it needs at least 1 bit to code a symbol, even if the probability of the symbol is greater than 0.5 in which case an optimal code would encode it in less than one bit!

The limitations of Huffman coding are addressed by arithmetic coding (Rissanen and Langdon, 1981; Witten et al., 1987). Although arithmetic coding codes each symbol in turn, it effectively codes the entire message as a real number between 0 and 1. The number of bits coding for a symbol need not be an integer. Arithmetic coding maintains an internal state which can be carried forward from the encoding of one symbol to influence the coding of the next symbol. By this means, more than one consecutive symbol can share a bit in the encoded stream. Arithmetic coding can therefore code a message arbitrarily closely to the theoretically limit (the negative logarithm of the probability).

Since arithmetic coding achieves an efficient code, compression is largely determined by modelling. Modelling of a data source involves making use of knowledge about the data source for predicting the message being emitted. It is therefore often said that compression of a data source is about modelling the data source, e.g., in (Allison et al., 1999). This thesis focuses on modelling biological processes to compress biological sequences.

Modelling techniques can be broadly classified into two groups, namely *substitutional* modelling and *statistical* modelling. In substitutional modelling, compression is achieved by replacing common phrases (groups of consecutive symbols) with indexes into some dictionary. The first method in this group, the LZ77 (Ziv and Lempel, 1977) uses the previously seen symbols in a recent window as the implicit dictionary. The LZ77 encoder searches in the dictionary for the longest match to the next block of symbols ahead, and encodes the block with the index of match found. LZ77 approach makes an implicit assumption that patterns generally occur together and thus is less effective in data where patterns recur over a long period. The LZ78 approach Ziv and Lempel (1978) resolves this problem by explicitly keeping a dictionary of frequently occurring patterns. The central decision to make in designing this approach is what phrases are to be included in the dictionary. Although substitutional compression techniques are generally not as effective as statistical compression techniques, they are faster and hence are used in many practical applications such as the UNIX tools *compress* and *gzip*. A substitutional modelling often results in a list of indexes of patterns rather than the probabilities, and thus does not strictly fit in the modelling-coding framework considered in this thesis.

Statistical compression techniques such as those in the PPM family (Cleary and Witten, 1984b) compress a sequence of symbols by estimating the probability of each symbol in turn, and encoding the symbol by one of the coding approaches such as Huffman coding or arithmetic coding. A predictive model is employed in these compression techniques. The model predicts the next symbol to be encoded on the background of all the symbols encoded previously. A simple example of a predictive model is an *order-k Markov model* (sometimes referred to as a *finite context model*), which makes predictions based on the *context* of the previous k symbols. An order-0 Markov model allocates a fixed probability to a symbol regardless of the position of the symbol in the message. For example,

the probability of a character in English may be better estimated if some preceding symbols are considered: The probability of a 'u' following a 'q' is estimated to be over 99% compared to just 2.4% if the preceding symbol is not considered (Bell et al., 1989). A model that uses a context of one preceding symbol is an order-1 Markov model.

It may be tempting to think that a higher-order Markov model should give better compression. However, such a model requires the estimation of a large number of probabilities covering all possible contexts, and the number of possible contexts increases exponentially with the order number. The model has a large number of parameters to be estimated from a limited amount of data. This can result in sub-optimal compression, sometimes worse than a lower order Markov model.

Sometimes several models are used together for compression. This is done by *combining* these models into a single predictive model. For example, the *prediction by partial match* (PPM) (Cleary and Witten, 1984b) algorithm uses several Markov models, from order-0 up to order-k where k is a specified number. Each Markov model maintains a table of probabilities for all of its contexts. To estimate the probability of occurrence of a symbol given the current context, the algorithm finds the model with the highest order q ($0 \leq q \leq k$) that has seen the symbol following the current context. Each of the higher order ($q + 1$ to k) models, not having seen the symbol following the current context, assigns an *escape probability* to the symbol. The final estimated probability of the symbol is the product of these escape probabilities multiplied by the probability estimated by the order-q Markov model.

Instead of combining models based on some heuristics as in the PPM algorithm, other approaches blend models based on their recent performance. In the multi-modal data compression algorithm (MMDC) (Williams, 1991), the local performance of each model is measured by the compression of recent symbols by the model. The best performed model at a position is used to predict the current symbol. The TMW (Meyer and Tischer, 1998) algorithm, employs a linear blending technique in which the negative logarithm of the weight given to a model is proportional to the number of bits needed to encode some recent symbols by the model.

Outside the two main groups mentioned above, some compressors use the Burrow-Wheeler transform (BWT) (Burrows and Wheeler, 1994). The BWT is a particular reversible permutation of the sequence of symbols that is easy to compress using a simple compression method. Such a transformation does not explicitly *model* the data source. Instead of encoding the stream of symbols as in the substitutional and statistical techniques, the (straight-forward) BWT requires the whole sequence be available to the encoder prior to compression. The BWT has been applied to many bioinformatics applications ranging from indexing (Ferragina and Manzini, 2000) to short read alignment (Li and Durbin, 2009). However it is unsuitable because modelling is the main aim in this thesis.

It is crucial that both the sender and the receiver have access to the same model. Three mechanisms, namely *static*, *semi-adaptive*, and *adaptive* (Boulton and Wallace, 1969; Bell et al., 1989) can be used to achieve this. In static modelling, the same model is used for compression of all messages. The model is available to both the sender and the receiver. Static modelling can perform poorly when the message does not correspond to model. Semi-adaptive modelling constructs a model by scanning the message prior to compression. The model is transmitted to the receiver before any compressed message is sent. The extra cost of transmitting the model generally pays off because the model is well suited to the messages.

Adaptive modelling provides a mechanism that can avoid the transmission of the model. In this mechanism, the model is built incrementally by both the sender and the receiver. At first, they assume some bland model, and use this model to encode and to decode the first symbol. After one symbol is encoded and decoded, both parties update the model in an identical manner so that the model is the same at each end. The new model is then used to encode and to decode the next symbol, and so on, until the entire message is transmitted. Adaptive modelling is attractive in that the model does not have to be transmitted explicitly and yet the model becomes well-suited for the message after a while. In has been formally proved by Cleary and Witten (1984a) that for every non-adaptive compression model, there exists an adaptive model that performs at least as well. In other words, the performance of adaptive modelling is at least as good as static and semi-adaptive modelling. Indeed, the most effective compression techniques in most data domains in practice are adaptive.

Chapter 3

Biological Sequence Compression

Simplicity is the ultimate sophistication.
–Leonardo da Vinci

3.1 Introduction

The genome stores all the genetic information necessary for the development and functioning of a living organism. It contains the instructions needed to construct other macromolecules such as RNAs and proteins which virtually control all processing within a cell. The genome, hence, ultimately defines the organism. The information is stored in a simple sequence over the alphabet of four nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T) and yet it fully accounts for the organism's assemblage of inherited traits. This naturally leads to a very intriguing question: How much information is in a genome? It is also very interesting to be able to measure the information content of every symbol in a genomic sequence, and to measure the information that is shared between any two genomes.

In information theory (Shannon, 1948), the *information content* of an event is defined as the negative logarithm of the probability of the event. In other words, it measures how unpredictable the event is. The *mutual information content* of two events measures the amount of information that can be obtained about one event by observing the other. This can be used to measure the relatedness of the two events. While the exact information content is not known, it can be approximated by lossless compression. A compression model which can capture the redundancy of the data is required for this process. The closeness of the approximation depends on the compression performance of the model.

The exponential increase of biological data poses a challenge to the development of tools to analyse bioinformatics data. Conventional sequence analysis methods are often

overwhelmed by volume and misled by statistical biases. A class of methods based on information theory are shown to overcome these issues. Work by Stern et al. (2001) recognises the importance of mutual compressibility for discovering patterns of interest from *Plasmodium* genomes. Chen et al. (2000) and Powell et al. (2004) show that compressibility is a good measurement of relatedness between sequences and can be used effectively in sequence alignment and evolutionary tree construction. These methods often require a compression model, the performance of which is crucial to the performance of the analysis methods. It is therefore important to develop compression techniques that are effective and are able to handle long sequences.

The massive growth in the amount of biological data recently has also motivated the development of effective compression techniques which can be used for biological sequences. Biological databases contain a great many highly similar genetic sequences from related organisms, and thus exhibits a high level of redundancy. Removing the redundancy would significantly reduce the cost of data storage and communication, which is likely to become a significant issue in the near future. However in this thesis, compression is primarily used as a criterion for statistical analysis.

Since DNA is the instruction-set of life, it is expected that DNA sequences are not random and should be compressible. Some DNA sequences are indeed highly repetitive. In the human genome for example, it is estimated that "repeat sequences account for at least 50% and probably much more" (Lander et al., 2001). A repeat sequence is a copy of a previous sequence in the genome in either forward or reverse complementary sense. Most DNA repeats are not exact as nucleotides can be changed, inserted and deleted. As an example, the Alu family are repeats in length of about 300 bases, and many Alu sequences are only about 70–80% similar to the consensus sequence (Deininger et al., 1992; Mighell et al., 1997).

Interestingly, biological sequence compression is highly challenging. Most general purpose text compression algorithms fail to compress DNA better than the naive 2 bits per symbol encoding. Proteins have even been considered "incompressible" (Nevill-Manning and Witten, 1999). This is because regularities in biological sequences are different from those in texts and are rarely modelled by general compression algorithms. A number of special purpose compression algorithms for biological sequences have been developed recently. They generally treat biological sequences as strings containing approximate repeats. Although most biological sequence compressors to date are able to beat the 2 bits per symbol boundary, they lack biological interpretations and have high time and space complexities. Therefore, they are unable to compress sequences longer than one megabase.

This chapter presents the *expert model*, an effective algorithm for compression of biological sequences. The algorithm outperforms all special purpose biological sequence

compression algorithms available in the literature. The algorithm is capable of compressing sequences as long as the human genome on a normal desktop computer. The algorithm provides many novel features useful for analysing sequences, which are generally not available in other biological sequence compression algorithms. The features include estimating the per symbol information content and performing compression of a sequence on the background knowledge of other sequences.

The chapter is organised as follows. Following this introductory subsection, Section 3.2 reviews several applications of compression to bioinformatics problems. Section 3.3 presents a survey of existing biological sequence compression algorithms and analyses techniques commonly used in these algorithms. Section 3.4 describes the expert model algorithm and Section 3.5 shows experimental results. A preliminary knowledge discovery task, repeat detection, using the expert model is presented in Section 3.6. Section 3.7 presents an analysis of the algorithm. Section 3.8 summarises the chapter and discusses ideas of future research. Further applications of the expert model to sequence analyses are presented in subsequent chapters.

3.2 Applications of Biological Sequence Compression

Since medieval times, it has been argued that nature follows the simplest rules. This principle, referred to as "Ockham's Razor" after the fourteen-century philosopher William of Ockham, suggests that scientific methods should prefer simpler hypotheses. This is the basic idea behind the *Minimum Message Length* (MML) principle discovered by Wallace and Boulton (1968). The principle combines information theory with Bayes's theorem to trade a model's complexity against its goodness of fit to data. The MML objective function is the length of a message which contains (i) a statement of a model and (ii) a concise encoding of the data using that model. The most probable hypothesis is the one that gives the shortest message length. A similar approach to MML, *Minimum Description Length* (MDL) was developed by Rissanen (1978) a decade later. While the two are closely related, MDL advocates selecting a model *class*, while MML advocates selecting a fully parameterised model with parameters stated to *optimal* precision.

MML is related to the concept of *algorithmic complexity* which was discovered independently by Solomonoff (1964), Kolmogorov (1965) and Chaitin (1966), and is often referred to as Kolmogorov complexity. The algorithmic complexity of a string is the length of the shortest description of the string in some programming language such as C, Pascal, or a universal Turing Machine. Kolmogorov complexity is, unfortunately, not computable; there exists no program that takes a string as input and outputs the algorithmic complexity of the string. It is therefore not applicable in practice. A related measure, *entropy*, was formulated by Shannon (1948). The value of the entropy of a source that emits

a sequence of data depends on the statistical nature of the source and is the fundamental limit to lossless compression of the data emitted by the source. If the statistical nature of the source is known, the entropy of the source can be computed. This is unfortunately, rarely the case in the context of biological sequences. The entropy of a biological source, however, can be estimated by modelling the source. The goodness of the model is measured by the ability of the model to compress the data generated by the source. In other words, compression of biological sequences is equivalent to modelling of biological processes, and thus is essential to inductive inference from biological data.

Inductive inference using compression has been used in a number of bioinformatics applications. The minimum message length encoding method by Allison and Yee (1990) is applied to comparison of biological sequences. Unlike conventional alignment methods which compare two strings by the edit distance between them, the MML encoding method computes the posterior odds-ratio of a string alignment under a mutation model. Suppose a sender must transmit two strings to a receiver. If the strings are truly related under some model, the alignment message length is shorter than sending them independently. The best model results in the shortest message length.

Chen et al. (2000) define a genetic distance measure of two biological sequences based on the compressibility of the two sequences using a compression algorithm named *GenCompress*. The measure is demonstrated in an application in evolutionary tree construction. Similarly, Burstein et al. (2005) apply Kullback-Leibler relative entropy for measurement of genetic distances where a simple Markov model is used for sequence compression. An information theoretic approach is also used in work by Otu and Sayood (2003) which uses a Lempel-Ziv model for compression. Though these compression models generally do not compress biological sequences well, these works successfully construct plausible phylogenetic trees. A better compression model is expected to improve the accuracy and reliability of the phylogenetic trees constructed.

Information theoretic approaches to measure string similarity is applied in protein sequence classification (Kocsor et al., 2005). The work claims that measurement of distances using a generic compression algorithm such as Lempel-Ziv and PPMZ is inferior to the conventional alignment distance measure, but the combination of both approaches outperforms alignment-based distance measure. It is expected that a biology related compression model will measure the relatedness of protein sequences better than those general purpose compression schemes do.

Compressibility is also applied for discovering patterns in biological sequences. The underlying assumption of the method is that patterns are discovered by finding an encoding method which effectively compresses observed data. A compression method based on encoding of previously occurring runs is used in (Milosavljevic and Jurka, 1993a) and (Milosavljevic and Jurka, 1993b) to discover tandem repeats and Alu families respectively. The compression method is subsequently improved by Powell et al. (1998b).

The *information content sequence* generated by a compressor (Dix et al., 2007) is useful for pattern discovery. Analysis by Stern et al. (2001) on *Plasmodium falciparum* chromosomes 2 and 3 detects low complexity regions in telomeric and central regions, long repeats in the sub-telomeric regions, and shorter repeat areas in dense coding regions. It is also revealed that the telomeric regions of the two chromosomes are similar since the conditional information content, i.e., the information content of a chromosome given the other, in those areas is significantly lower than information content obtained by compressing the sequence alone.

The above discovery tasks rely on a compression model to perform knowledge discovery. Their performance is largely dependent on the effectiveness of the underlying compression model. The designing of more effective is therefore important to the study of biological sequences using information theoretic approaches.

3.3 Biological Sequence Compression Review

The rapid increase in biological data in the last two decades has naturally led to research in DNA and protein compression. Since the introduction of the first biological sequence compression algorithm, *BioCompress* (Grumbach and Tahi, 1993), there have been over ten other algorithms proposed to date. This section discusses the regularities of biological data and reviews the main techniques generally employed by biological sequence compression algorithms. It also presents a systematic review of existing biological sequence compression methods.

3.3.1 Compressible Features of Biological Sequences

A biological sequence can be considered as a text over an alphabet of size four in the case of DNA and RNA, and of size 20 in case of protein. Compression of biological sequences, however, appears to be much more difficult than that of natural language texts though biological sequences contain a great amount of repeats. This is because regularities in biological sequences are more difficult to model. General text compression algorithms often fail to compress biological sequences better than the base lines, while existing biological algorithms can compress only marginally better than a simple Markov model (Cao et al., 2007; Nevill-Manning and Witten, 1999). This subsection shows an analysis of features that make biological sequences compressible and reviews how biological sequence compression algorithms can exploit these features.

The compressibility of many sources is due to a skewed distribution of the alphabet. The distribution of characters in a natural language such as English is usually distinct. For example, in most English texts, the frequencies of the letters E and T, (usually the most

frequent), and Q and Z (typically the least frequent) are often identifiable. A zero-order Markov model thus can be used to compress English texts reasonably well. Analysing English texts using a longer context is also feasible due to certain spelling rules such as the character following a 'q' is almost always a 'u'. On the other hand, the distribution of letters in biological sequences is not universally distinct. For example, letters A and T make up 80% of the malaria parasite *Plasmodium falciparum* genome, and even over 90% in introns and intergenic regions, while the genome of another human malaria parasite, *Plasmodium vivax*, contains only 60% A and T. The unpredictability of DNA composition distribution argues against the use of a static compression mechanism.

Not only does the character composition vary from sequence to sequence but the distribution of characters in a sequence often also changes along the sequence. Different areas of a genome may have different functions and thus have different character compositions. As an example, the AT content in exons of the *Plasmodium falciparum* genome is 60% while that in introns is over 90%, even though exons and introns sit side by side in a gene. A semi-adaptive compression algorithm is not suitable for data with such changes in composition.

The presence of biological repeats (discussed in Subsection 2.1.4) leads to recurring patterns in biological sequences. These patterns are instances of information redundancy that can be exploited by compression techniques. However, modelling biological repeat patterns appears to be more difficult than modelling repeating patterns in natural language text. This is mainly because of the variability of lengths and the inexact nature of biological repeats.

Recurring patterns in text are exploited by most compression algorithms. Compression algorithms in the Lempel-Ziv family (Ziv and Lempel, 1977, 1978) store frequently occurring patterns in a dictionary, and encode each occurrence of a pattern by a reference to the pattern in the dictionary. The building blocks in a natural language are words which are rarely longer than 10 characters. In other words, repeat patterns in general texts are relatively short and hence are easy to store and to locate in a dictionary. Compression algorithms in the PPM family (Cleary and Witten, 1984b; Moffat, 1990) take advantage of this feature to estimate the probability of a symbol given the context of a small number of preceding symbols. Repeats from biological sequences, however, are much longer. An element of the long interspersed repetitive element family (LINE) which makes up over 20% of the human genome (Lander et al., 2001), can be as long as 10 kilobases (Singer, 1982). Maintaining a full probability distribution over all such high order contexts is infeasible. Furthermore, lengths of DNA repeats vary greatly, ranging from a few symbols in short tandem repeats to the size of a LINE. The variability of repeat lengths makes it hard to model repetition well.

Not only are biological repeats variable in size, but they are also generally not exact. Copying of DNA is subject to mutations, insertions and deletions and hence, copies of the

same repeat can be very different from each other. As an example, each copy of an Alu repeat in the human genome can be up to 20–30% different to the consensus sequence (Deininger et al., 1992; Mighell et al., 1997). DNA repeats are sometimes even reversed and complemented. Although biological repeats are not exact copies of each other, they do contain redundant information. As a result of their inexact nature, locating biological repeats for compression is very difficult. In a long sequence over a small alphabet, one would expect many "random" repeats. For a uniformly distributed DNA sequence, a character is expected to occur every four places, on average. Similarly, a 10-mer would be expected to occur about once every 1 million symbols ($4^{10} = 1048576$). In other words, a given 10-mer might be expected to occur 3000 times in a sequence of the human genome size. On the other hand, significant repeats are approximate and also variable in length. It is therefore difficult to detect "true" significant repeats without introducing many false positives.

The large amount of biological sequence data today also poses another challenge to compression: Sequences are very long, and this prevents the use of data structures and algorithms of high complexity. An exhaustive search for repeats is almost impossible to be applied to a sequence of human genome size, let alone the availability of thousands of genomes now available.

3.3.2 Review of Biological Sequence Compression Algorithms

Most compression algorithms fall into one of two categories, namely *substitutional* compression and *statistical* compression. Those in the former class replace a long repeated sequence by a pointer to an earlier instance of the sequence or to an entry in a dictionary. Examples of this category are the popular Lempel-Ziv compression algorithms (Ziv and Lempel, 1977, 1978) and their variants. As DNA sequences are known to be highly repetitive, a substitutional scheme is a plausible approach to take. Indeed, most biological sequence compression algorithms to date are in this category.

On the other hand, a statistical compression method such as the *prediction by partial match* (PPM) (Cleary and Witten, 1984b) predicts the probability distribution of each symbol. Statistical compression algorithms depend on assumptions about how the sequence is generated to calculate the distribution. These assumptions are said to be the *model* of the sequence. If the model gives a high estimated probability to the actual value of the symbol, good compression is obtained. A model that produces good compression makes good predictions and is a good description of the data.

The earliest special purpose DNA compression algorithm found in the literature is *BioCompress* developed by Grumbach and Tahi (1993). BioCompress detects an exact repeat in DNA using an automaton, and uses Fibonacci coding (Fraenkel and Klein, 1996) to encode the length and position of the previous occurrence of the repeat sequence. If

a region is not a repeat, it is encoded by the naive 2 bits per symbol code. An improved version, *BioCompress-2* (Grumbach and Tahi, 1994) uses a second order Markov model to encode non-repeat regions. The *Cfact* DNA compressor developed by Rivals et al. (1996) also searches for the longest exact repeats but is a two-pass algorithm. It builds the suffix tree of the DNA sequence in the first pass, and performs the actual encoding in the second pass. Non-repeat regions are also encoded by the 2 bits per symbol encoding method. The *Off-line* approach by Apostolico and Lonardi (2000) iteratively selects repeated substrings for which encoding would gain maximum compression. These algorithms are in the substitutional class.

A similar substitution approach is used in *GenCompress* by Chen et al. (2000) except that approximate repeats are exploited. An inexact repeat sequence is encoded by a pair of integers, as for *BioCompress-2*, plus a list of edit operations for mutations, insertions and deletions. Since most repeats in DNA are approximate, *GenCompress* obtains better compression ratios than *BioCompress-2* and *Cfact*. The same compression technique is used in the *DNACompress* algorithm by Chen et al. (2002), which finds significant inexact repeats in one pass and encodes these repeats in another pass.

Most other compression algorithms employ techniques similar to *GenCompress* to encode approximate repeats. They differ only in the encoding of non-repeat regions and in detecting repeats. The *CTW+LZ* algorithm developed by Matsumoto et al. (2000) encodes significantly long repeats by the substitution approach, and encodes short repeats and non-repeat areas by context tree weighting (Willems et al., 1995). At the cost of higher time complexity, *DNAPack* Behzadi and Fessant (2005) employs a dynamic programming approach to find repeats. Non-repeat regions are encoded by the best choice from a second order Markov model, context tree weighting, and the naive 2 bits per symbol methods.

Adjeroh et al. (2002) propose a substitution method that explicitly builds a dictionary of repeat patterns. The method uses the *Burrows-Wheeler Transform* (Burrows and Wheeler, 1994) to locate repeats in a preprocessing pass. In the second pass, each repeat is examined based on its length and the number of its occurrences. Only repeats with guaranteed compression gain are added to the dictionary. A very fast DNA compressor reported by Manzini and Rastero (2004) obtains its speed by employing a fingerprint approach to locate matched seeds. Matches are greedily extended for the longest exact matches. Although its compression ratio is inferior to other predecessors, it is the fastest special purpose DNA compressor found in the literature.

Several DNA compression algorithms combine substitutional and statistical styles: An inexact repeat is encoded using (i) a pointer to a previous occurrence and (ii) the probabilities of symbols being copied, changed, inserted or deleted. In the *MNL* algorithm (Tabus et al., 2003) and its improvement, *GeMNL* (Korodi and Tabus, 2005, 2007), the DNA sequence is split into fixed size blocks. To encode a block, the algorithm searches

the history of seen symbols for a regressor, which is a sequence having the minimum Hamming distance from the current block, and represents the block by a pointer to the regressor and a bit mask for the differences between the block and the regressor. The bit mask is encoded using a probability distribution estimated by the normalised maximum likelihood of similarity between the block and the regressor.

Probably the only two pure statistical DNA compressors published so far are *CDNA* (Loewenstern and Yianilos, 1999) and *ARM* (Allison et al., 1998). The former algorithm obtains the probability distribution of each symbol by combining predictions based on previously seen approximate partial matches. Each approximate match is a previous block having the smallest possible Hamming distance to the context preceding the symbol to be encoded. Predictions are combined using a set of weights which are learnt adaptively by an expectation maximisation process. Conceptually, CDNA can learn its parameters as each symbol is compressed and thus is an adaptive algorithm. However, due to the high complexity of the expectation maximisation, the learning process is only invoked after every segment of the sequence is compressed.

ARM algorithm forms the probability of a sequence by summing the probabilities over all explanations of how the sequence is generated. This is done by a modified dynamic programming algorithm that sums the probabilities of all alignments of possible repeats, within the string compression model. ARM has a set of parameters modelling the events in the replication of DNA such as the rates of mutations, insertions and deletions. The dynamic programming algorithm can be used as the step in an expectation maximisation algorithm to estimate the parameters. ARM therefore is a semi-adaptive algorithm.

As statistical approaches, CDNA and ARM yield significantly better compression ratios than those in the substitutional class and can also estimate the per element information content sequences. *CDNA* has many parameters which do not have obvious biological interpretations while *ARM* has few parameters and each relates directly to biology. Parameters in both approaches are estimated by an expectation maximisation process. Both algorithms are high in time complexity and hence are not practical for compressing long sequences.

Compression of protein sequences has been a long running challenge (Nevill-Manning and Witten, 1999). developed a sophisticated compression scheme that uses up all contexts up to a certain length, weighted by their similarity to the current context. The results however, are not better than a simple, low order Markov model, which leads to the negative conclusion. The ProtComp algorithm, developed by Hategan and Tabus (2005) considers a substitution probability matrix of amino acids, and produces more optimistic results.

The expert model presented in this chapter is a statistical algorithm. It maintains a panel of experts and combines them for prediction but a much simpler and computationally cheaper mechanism than CDNA and ARM is used. The framework allows any kind of experts to be used, though only experts obtained from statistics and repetitiveness of sequences are reported here. Weights of experts are assigned based on their performance. Generally, this algorithm is superior to any compression algorithms to date and its speed is practical. The algorithm can produce an estimate of the per element information content of a sequence (Dix et al., 2007) which facilitates biological knowledge discovery. That is the primary purpose of this research.

3.3.3 Analysis of Available Techniques for Biological Sequence Compression

Most redundancy in biological sequences comes from repetition. Therefore, in order to compress a sequence, a biological sequence compressor generally needs to (i) identify repetitive regions and (ii) encode these regions to obtain a compact representation of the sequence. This subsection analyses techniques which are potentially useful for biological sequence compression. Some of these techniques have been employed by many existing algorithms.

A biological sequence compression algorithm must be able to deal with repeats. Early biological algorithms are modifications of Ziv-Lempel dictionary scheme and they consider only exact repeats. Since biological repeats are approximate, handling only exact repeats will miss out approximate long repeat sequences. On the other hand, in a long sequence, there can be a great many random matches longer than a typical "genuine" repeat. For example, a typical member of the Alu repeat family in the human genome can be over 20% different from the consensus. In other words, in a pair of Alu elements, one can expect a difference every 5 symbols on average. Therefore, an algorithm using exact repeats can only find repeats with an average length of 5. Every 5-mer, however, would be expected to randomly occur about every 1024 bases (4^5) and thus would occur 3 billion times by chance in a sequence of human genome size. These algorithms therefore, do not yield very good compression. Later algorithms consider approximate repeats, and as a result, perform much better.

Identification of repeats is important to a biological sequence compression algorithm. *Biocompress* and *Biocompress-2* employ an automaton to detect repeats, both forward copies and reverse complements. The depth of the automaton is limited to a size h ($h = 8$ in their experiments) in order to prevent the automaton from becoming too large. These algorithms start with finding repeats of the size h or smaller using the automaton. Longer repeats are found by searching both the automaton and the sequence at the expense of much more processing time. The automaton technique restricts these algorithms to working on only exact repeats.

Suffix structures such as suffix trees and suffix arrays are employed in a number of biological sequence compression algorithms. There are two main approaches to using suffix structures. In the first approach, a suffix tree or a suffix array is used to index a sequence to locate repeats which are then encoded by some special scheme (Rivals et al., 1996). In the second approach, as adopted by several algorithms such as (Adjeroh et al., 2002), a suffix tree or a suffix array is used in the computation of a sequence's Burrows-Wheeler Transform (BWT) (Burrows and Wheeler, 1994) which is more easily compressed. The advantage of a suffix structure over an automaton is that the suffix structure can locate the longest possible repeats. However, compression algorithms using these suffix structures are only able to reliably find exact repeats. Suffix trees have been known for high memory requirement and a large constant even for linear time complexity. Over the last decade, development on suffix arrays has made them better than suffix trees in both time and space complexities (Abouelhoda et al., 2006; Kärkkäinen et al., 2006; Schürmann and Stoye, 2007).

Several biological sequence compression algorithms (Behzadi and Fessant, 2005; Venugopal et al., 2009) employ the dynamic programming approach to locate repeats. A direct implementation of dynamic programming would be able to find all matches but has a quadratic time complexity which is too high for long sequences. The ARM (Allison et al., 1998) searches for all approximate matches using dynamic programming. It learns a statistical model from the matches found. The algorithm requires several expectation-maximisation iterations of quadratic complexity each and thus is extremely slow. Nonetheless, ARM produces the best results among existing algorithms. To reduce the execution time, a hash table is employed to find repeats that share a common seed. Repeats are then extended using dynamic programming. This allows ARM to ignore a large area of the search space to trade off speed for compression.

Works in recent years turn attention to indexing data structures that can handle mismatches. Spaced seeds are employed in PatternHunter (Ma et al., 2002) which is the core of the DNACompress (Chen et al., 2002) algorithm. PatternHunter is reported to be more sensitive than ordinary contiguous seeds for sequence searching. The effect of spaced seeds for biological sequence compression, however, has not been investigated previously.

Generally statistical approaches perform better than substitutional approaches, albeit they are normally more computationally expensive. This observation coincides with compression of other types of data such as texts and images. Most existing biological sequence compression algorithms belong to the substitutional approach. As a departure from the substitutional approach, the GeMNL (Korodi and Tabus, 2005, 2007) uses a hybrid method combining both approaches and, as a result, is superior to other substitutional methods. The two pure statistical methods, CDNA (Loewenstern and Yianilos,

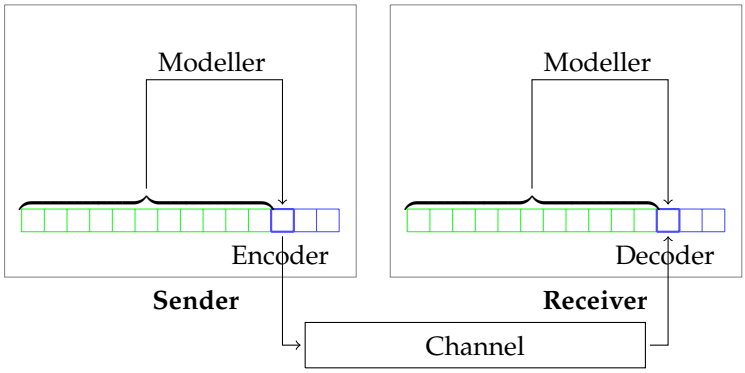


Figure 3.1: Mechanism of sequence compression by the expert model.

1999) and ARM (Allison et al., 1998) perform the best among existing algorithms. However, their high complexity in time and space prohibits them from practically compressing long sequences.

3.4 The Expert Model Compression Algorithm

The expert model is a statistical compression method. It comprises two components, a modeller and a coder. To compress a symbol in a sequence, the modeller assigns a probability distribution to the symbol and the coder constructs a compressed representation of the actual symbol with respect to the probability distribution. A high probability symbol is coded with a short code word. Therefore, the better the modeller predicts symbols, the shorter the compressed message is. There exist efficient coding schemes, such as arithmetic coding (Rissanen and Langdon, 1979; Witten et al., 1987), that can produce code words that are arbitrarily close to the theoretically optimal length, which is the negative binary logarithm of the probability of the symbol (Shannon, 1948):

$$\mathcal{I}(x_i) = -log_2 Pr(x_i) \tag{3.1}$$

The compression is, therefore, dependent on the prediction of the modeller.

The algorithm compresses a sequence by compressing each symbol in turn. To compress a symbol, the modeller forms the probability distribution over the symbol's possible values based on the information from all symbols previously seen. The encoded message of the symbol is written to the channel and sent to the decompressor, which maintains an identical modeller. The decompressor, having seen all previous decoded symbols, and hence having the same information as the compressor, is able to compute the identical probability distribution and can thus recover the symbol at the position. The mechanism is depicted in Figure 3.1.

In order to form the probability distribution of a symbol, the algorithm maintains a set of *experts*. An expert is any model that can provide a probability distribution for the symbol. The experts' opinions about the symbol are *blended* to give a combined prediction. The statistics of symbols can vary over the sequence. One expert may perform well in one region, but could give bad advice in other regions. The reliability of an expert is evaluated based on its recent performance. A reliable expert is given a high weight in the blending while an unreliable one has little influence on the final prediction or may even be ignored.

3.4.1 Types of Experts

An expert can be any entity that can provide a reasonably good probability distribution for the symbol at a position in the sequence. An example is the *Markov expert* of order k which uses a Markov model learnt from the statistics of the sequence "so far" to give the probability of a symbol in the context of k preceding symbols. Initially, the Markov expert does not have any prior knowledge of the sequence and thus gives a uniform distribution to the prediction. The probability distribution adapts as the encoding proceeds. Essentially, the Markov expert provides the background statistical distribution of symbols over the sequence.

Different areas of a DNA sequence may have differing functions and thus may have different probability distributions. For example, in the *Plasmodium falciparum* genome, the probability distribution of symbols in exons is more uniform than in non-coding regions. Another type of expert, the *local Markov expert*, is employed to model this situation, It calculates the probability distribution of a symbol from the statistics of the *local* history rather than the entire history of the sequence.

As biological sequences are highly repetitive, it is important to include *repeat experts* that make use of repeat patterns for compression. The first type of repeat expert is the *copy expert* which considers the current symbol to be part of a copied region at a particular offset: A copy expert with offset f suggests that the symbol at position i is copied from the symbol at position $i - f$. A copy expert does not blindly give a high probability to its suggested symbol. It uses an adaptive code (Boulton and Wallace, 1969) based on its correct and incorrect predictions. It reviews its own performance over some recent history and accordingly builds a probability distribution for the mutation rate. The copy expert gives a probability to its predicted symbol of:

$$p = \frac{r + 1}{w + 2} \tag{3.2}$$

where w is the window size over which the expert reviews its performance and r is the number of correct predictions the expert has made. The remaining probability, $1 - p$, is distributed evenly to the other letters in the alphabet.

If there is a mutation within a repeated region, the corresponding copy expert gives a bad prediction at mutated symbol and its weight is decreased as a result. However, given its subsequent good predictions, the expert will regain its influence in the combined prediction. On the other hand, if the repeated region ends, the copy expert will make many mistaken predictions. After a number of bad predictions, the weight of the expert drops to below a threshold and the expert is then removed from the panel. That also happens when an insertion or a deletion occurs, the copy expert is no longer able to make good predictions and is eventually excluded to make room for a new copy expert; the newly proposed expert could possibly have a similar offset to the excluded expert and thus the rest of the information from the repeat region can be used for compression. By these means the algorithm is able to use approximate matches for compression.

For modelling complementary reverse repeats in DNA, another type of repeat experts, *reverse experts* is used. These experts work exactly the same as copy experts, except that they suggest the complementary symbol to the one from the earlier instance and proceed in the reverse direction.

3.4.2 Proposing Repeat Experts

At position $i + 1$ in the sequence, there are i possible copy experts and i possible reverse experts. This is too many to combine efficiently and anyway most are not *genuine* and thus would be ignored. The algorithm therefore, must use at most a small number of repeat experts at any one time. The algorithm has a parameter L, which specifies the maximum number of repeat experts to be employed at a time. When the expert panel size is less than L, the algorithm may recruit more potential repeat experts. Since the number of experts must be small, it is desirable that the experts proposed are those that are most likely to be genuine experts.

A simple technique to propose potential experts uses a hash table. The hash table associates each position in the sequence with the hash key composed of k symbols *preceding* the position. At a position, the hash table proposes experts which correspond to positions that share the same hash key with the current position. The choice of hash key size k and the expert limit L is a trade-off between running time and compressibility. Generally, a small k and a large L allow the model to search for repeats more thoroughly and thus give better compression at the cost of more time.

3.4.3 Combining Experts' Predictions

Not only do individual experts adapt themselves based on the context of seen symbols, the algorithm also adaptively adjusts experts' weights to reflect the "quality" of each expert given the context. Good experts are assigned high weights. Even being nominated by the hash table, some copy and reverse experts are due to mere random matches and thus their predictions are not significantly better than a Markov expert in a reasonably long run. The algorithm must be able to exclude the by-random copy and reverse experts to reduce noise and to be more time efficient. Furthermore, a "genuine" repeat expert performs well only within a repeated region. Beyond this, it provides random predictions and thus should also be excluded. It is important that the algorithm must be able to evaluate the quality of each expert to assign its weight accordingly, and to exclude it if necessary.

The core part of the expert model is the evaluation and combination of expert predictions. Suppose at position n, a panel of experts E is available to the compressor. Expert θ_e gives the probability $Pr(x_n|\theta_e, x_{1..n-1})$ of symbol x_n based on its observations of the preceding $n-1$ symbols. It is assigned a weight w_e which reflects the reliability of expert θ_e. The expert model performs a linear blending of expert predictions to give the probability distribution of the symbol x_n:

$$Pr(x_n|x_{1..n-1}) = \sum_{\theta_e \in E} Pr(x_n|\theta_e, x_{1..n-1}) w_{\theta_e} \tag{3.3}$$

in which the sum of all weights is equal to 1:

$$\sum_{\theta_e \in E} w_{\theta_e} = 1 \tag{3.4}$$

On the other hand, if $Pr(x_n|x_{1..n-1})$ is considered as the probability of x_n under all possible hypotheses θ_e, then by marginalisation:

$$Pr(x_n|x_{1..n-1}) = \sum_{\theta_e \in E} Pr(x_n|\theta_e, x_{1..n-1}) Pr(\theta_e|x_{1..n-1}) \tag{3.5}$$

Therefore, a sensible way to combine the experts' predictions is based on Bayesian model averaging (Hoeting et al., 1999):

$$w_{\theta_e} = Pr(\theta_e|x_{1..n-1}) \tag{3.6}$$

In other words, the weight w_{θ_e} of expert θ_e in encoding x_n equals the posterior probability $Pr(\theta_e|x_{1..n-1})$ of the expert after encoding $n-1$ symbols. Assuming the background of context $x_{1..n-2}$, $Pr(\theta_e|x_{1..n-1})$ can be considered as the posterior probability of expert θ_e

after encoding symbol x_{n-1} and thus, by Bayes's theorem:

$$\begin{aligned} Pr(\theta_e|x_{1..n-1}) &= Pr(\theta_e|x_{1..n-2}, x_{n-1}) \\ &= \frac{Pr(x_{n-1}|\theta_e, x_{1..n-2})Pr(\theta_e|x_{1..n-2})}{Pr(x_{n-1}|x_{1..n-2})} \end{aligned} \tag{3.7}$$

in which $Pr(\theta_e|x_{1..n-2})$ is the prior probability of expert θ_e *before* encoding symbol x_{n-1}, $Pr(x_{n-1}|\theta_e, x_{1..n-2})$ is the probability of x_{n-1} estimated by expert θ_e and $Pr(x_{n-1}|x_{1..n-2})$ is the marginal probability of x_{n-1}. Recursively applying Equation 3.7 for x_{n-2}, x_{n-3},...,x_1 gives

$$\begin{aligned} w_{\theta_e} &= Pr(\theta_e|x_{1..n-1}) \\ &= \frac{Pr(x_{n-1}|\theta_e, x_{1..n-2})}{Pr(x_{n-1}|x_{1..n-2})} Pr(\theta_e|x_{1..n-2}) \\ &= \frac{Pr(x_{n-1}|\theta_e, x_{1..n-2})}{Pr(x_{n-1}|x_{1..n-2})} \frac{Pr(x_{n-2}|\theta_e, x_{1..n-3})}{Pr(x_{n-2}|x_{1..n-3})} Pr(\theta_e|x_{1..n-3}) \\ &= \frac{\prod_{i=1}^{n-1} Pr(x_i|\theta_e, x_{1..i-1})}{\prod_{i=1}^{n-1} Pr(x_i|x_{1..i-1})} Pr(\theta_e) \end{aligned} \tag{3.8}$$

Normalising Equation 3.8 by the common factor $M = \prod_{i=1}^{n-1} Pr(x_i|x_{1..i-1})$ obtains

$$w_{\theta_e} = \frac{1}{M} \prod_{i=1}^{n-1} Pr(x_i|\theta_e, x_{1..i-1}) Pr(\theta_e) \tag{3.9}$$

The normalisation factor M, in fact does not matter since Equation 3.3 could be again normalised to have $\sum w_{\theta_e} = 1$. Therefore

$$w_{\theta_e} \propto \prod_{i=1}^{n-1} Pr(x_i|\theta_e, x_{1..i-1}) Pr(\theta_e) \tag{3.10}$$

Taking the negative logarithm of Equation 3.10 gives

$$-log_2(w_{\theta_e}) \sim -\sum_{i=1}^{n-1} log_2 Pr(x_i|\theta_e, x_{1..i-1}) - log_2 Pr(\theta_e) \tag{3.11}$$

Since $Pr(x_i|\theta_e, x_{1..i-1})$ is the probability of symbol x_i estimated by expert θ_e, the value $-log_2 Pr(x_i|\theta_e, x_{1..i-1})$ is the cost of encoding symbol x_i by the expert and thus $-\sum_{i=1}^{n-1} log_2 Pr(x_i|\theta_e, x_{1..i-1})$ is the length of encoding sequence $x_{1..n-1}$ by expert θ_e. Instead

of evaluating experts on the whole history $x_{1..n-1}$, the expert model algorithm considers a recent history of size h; only the message length of encoding symbols $x_{n-h..n-1}$ is used to determine the weights of experts. The final formula of w_{θ_e} is

$$\begin{aligned} -log_2(w_{\theta_e}) &\sim -\sum_{i=n-h}^{n-1} log_2 Pr(x_i|\theta_e, x_{1..i-1}) - log_2 Pr(\theta_e) \\ &= \mathcal{L}(x_{n-h..n-1}|\theta_e) - log_2 Pr(\theta_e) \end{aligned} \tag{3.12}$$

or

$$w_{\theta_e} \propto 2^{-\mathcal{L}(x_{n-h..n-1}|\theta_e)} Pr(\theta_e) \tag{3.13}$$

In this equation, $\mathcal{L}(x_{n-h..n-1})$ is determined by measuring the message length of encoding symbols $x_{n-h..n-1}$ if only expert θ_e were consulted. This leaves the prior probability $Pr(\theta_e)$ to be estimated. All repeat experts are proposed by the same hash table and thus can be assumed to have the same prior probability. Therefore, only the ratio between the prior probability of a repeat expert and that of the Markov expert is required. A simple estimation of this is the ratio of the number of symbols that are part of a repeat element and the number of symbols that are not. A symbol is considered part of a repeat element if there exists a repeat expert that can encode the symbol significantly better than the Markov expert. Determining the number of repeat symbols is done by keeping a counter as the compression proceeds.

If a symbol is part of a significant repeat sequence, the copy or reverse expert making use of that repeat sequence must predict significantly better than a general prediction such as that from the Markov expert. A *listen threshold*, T, is therefore defined to determine the reliability of a repeat expert. A repeat expert is considered reliable if its encoding of the last h symbols is smaller than that of the Markov expert by T bits. T is a parameter of the algorithm.

Suppose there are three hypotheses about how a symbol is generated: by the distribution of the whole sequence, by the distribution of the local region, and by repeating something earlier. Three experts corresponding to the three hypotheses are used: (i) a Markov expert for the whole sequence distribution, (ii) a local Markov expert for the local distribution, and (iii) a combined repeat expert, which is the combination of any available copy and reverse experts, for the third hypothesis. The copy and reverse experts are first blended as in Equations 3.3 and 3.13 to produce the *combined repeat expert*, which is then blended with the Markov expert and the local Markov expert in the same manner. A formal description of the algorithm is given in Algorithm 1.

The expert model algorithm can also be used as an entropy estimator of biological sequences. The information content of every single symbol is estimated by the negative

Algorithm 1 Expert Model Compression Algorithm

XM($X_{1..n}$)
param L: limit on size of the expert panel E
param k: size of the hash key
param h: size of the window to evaluate experts
param T: threshold to discard repeat experts
$E \leftarrow$ empty set
for $n \leftarrow 1$ to $|X|$ **do**
  while $|E| < L$ and an expert θ_e which matches x_{n-f} to x_n is proposed **do**
    add θ_e into E
  end while
  $Pr(x_n) \leftarrow \sum_{\theta_e \in E} w_{\theta_e} Pr(x_i|\theta_e)$ where $w_{\theta_e} = 2^{-\mathcal{L}(x_{n-h..n-1}|\theta_e)}$
  code x_n based on $Pr(x_n)$
  for all $\theta_e \in E$ **do**
    $\mathcal{L}(x_n|\theta_e) \leftarrow -log_2 Pr(x_n|\theta_e))$
    update θ_e
    if $\mathcal{L}(x_{n-h+1}..x_n|\theta_e) > \mathcal{L}(x_{n-h+1}..x_n|\theta_{Markov}) - T$ **then**
      remove θ_e from E
    end if
  end for
  store $x_{n-k+1}..x_n$ in the hash table
end for

logarithm of its probability. To compress the sequence, the arithmetic coding scheme (Rissanen and Langdon, 1979; Witten et al., 1987) is used to code each symbol based on the probability distribution derived from the combination of experts.

3.4.4 Variants of the Expert Model

The model presented above is a general framework. Different components of the algorithm are independent from each other and can be extended in several ways. A variant of the algorithm can be created to suit the data being analysed. This subsection describes a number of variants of the expert model. These variants have been implemented and incorporated into the algorithm.

The hash table used to suggest repeat experts is based on the heuristics that a match of k symbols has a good chance of being a repeat, and hence a repeat expert is nominated. The expert is then evaluated based on its performance in compression of the next several symbols. There is a trade off in selecting the hash size k. A small k can result in many false positives from random matches while a large k may miss out many "genuine" experts because biological repeats generally contain many mutations. One method to improve the trade off is to use a *spaced seed* which allows a longer seed but only a number of symbols at certain places in the seed are required to match. In work by Ma et al. (2002), the use of

a single deterministic spaced seed is found to be more sensitive than a contiguous seed in biological database search. The authors report that, the optimal spaced seed of length 11 is 111010011001010111 in which only places marked 1 are required to match. The spaced seed hash table is incorporated in the expert model to allow users to select a seed suitable for data.

There are two important groups of nucleotides – purine (C and T) and pyrimidine (A and G). The biological properties of two nucleotides in one group are more similar than those from different groups. Therefore, substitutions changing nucleotides within a group (*transitions*) are more common than those that change the group (*transversions*). A variant of the hash table operates on the alphabet of just two letters, purine and pyrimidine. This hash table is less sensitive to mutations when detecting repeats and can be used with a longer seed.

Suffix trees and suffix arrays provide another heuristic approach for proposing repeat experts: they can suggest experts corresponding to the longest possible matches. However, they will miss out long approximate matches containing mutations. The suffix array and the suffix tree of the same sequence are essentially stores the same data and thus will suggest the same experts. A suffix array structure was implemented in this project as a possible alternative to a hash table.

Since repeat experts make predictions along the sequence, they can learn from their experience to make better predictions for later symbols. As presented in Section 3.4.1, the repeat experts assume a mutation model that specifies the rate of mutations in repeats. In particular, a repeat expert keeps track of the number of mutations and gives the probability of the next symbol based on the number of matches and mismatches. A more sophisticated model of mutation is for each repeat expert to keep track of the substitution matrix that describes the rate of change of one symbol to another. Such a repeat expert predicts the probability of the next symbol according to the substitution matrix. After each prediction, the expert updates its matrix according to the value seen. This type of expert is useful for compressing multiple sequences which are from related species (Cao et al., 2009b). Such compression is useful for sequence alignment as shall be presented in Chapter 4.

3.5 Experimental Results

The expert model algorithm was implemented in Java. All experiments with the expert model were performed on a workstation equipped with Pentium Duo Core CPU 2.33Ghz (E6550) and 8GB of memory. The compression results are calculated from the size of real encoded files in bits per symbol (bps). It was noted that the figures for actual compression and the estimated information content of a sequence are equal to four decimal places. The

subtle difference between the information content estimated and the actual compression is due to rounding in arithmetic coding and padding the last byte of the encoded files.

In order to compare the performance of the expert model with other biological sequence compression algorithms, the expert model was run on a standard data set that most other workers have used. Subsection 3.5.1 describes the comparison in detail. To demonstrate the scalability of the expert model, it is run on the 23 chromosomes of the human genome and a set of genomes of various species from different organism levels. The experiment is presented in Subsection 3.5.3.

3.5.1 Comparison of DNA Compression Results

A standard data set of DNA sequences was used to compare between compression algorithms. The data set has been used as a benchmark in most other DNA compression publications. The sequences in the data set come from a variety of species and sequence types, and their lengths range from 38 kilobases to 230 kilobases. They include five human genes (HUMDYSTROP, HUMGHCSA, HUMHBB, HUMHDABCD and HUMHPRTB), two mitochondria genomes (MPOMTCG and MTPACG), two chloroplast genomes (CHMPXX and CHNTXX) and the genomes of two viruses (HEHCMVCG and VACCG). The lengths (in *base pair*) and descriptions of these sequences are described in Table 3.1.

Table 3.1: Description of the sequences in the DNA data set.

Sequence	Length	Sequence Description
CHMPXX	121024	Marchantia polymorpha chloroplast genome
CHNTXX	155844	Nicotiana tabacum chloroplast genome
HEHCMVCG	229354	Human cytomegalovirus strain AD169
HUMDYSTROP	38770	Human syntrophin gene
HUMGHCSA	66495	Human chorionic somatomammotropin gene
HUMHBB	73308	Human beta globin region on chromosome 11
HUMHDAB	58864	Human three cosmids: HDAB, HDAC and HDAD
HUMHPRTB	56737	Human hypoxanthine phosphoribosyltransferase gene
MPOMTCG	186609	Marchantia polymorpha mitochondrion genome
MTPACG	100314	Podospora anserina mitochondrion genome
VACCG	191737	Vaccinia virus Copenhagen, complete genome

Firstly, various general compression algorithms of different styles were applied to the data set. The algorithms used in this experiment were: *GZIP* (of the Lempel-Ziv 77 family), *LZMA* (Lempel-Ziv Markov Chain algorithm), *BZIP2* (Burrows-Wheeler transform style), *PPMZ* (an implementation of the PPM algorithm) and *CTW* (the context switching algorithm). These algorithms are highly optimised and are frequently used for general text compression. Since they are designed mainly for compressing ASCII texts, each algorithm was run on two configurations of the data set: on "raw" files that represent each base by a character, and on files that pack every four nucleotides into an ASCII character.

The performance of these compression algorithms on the data set is reported in Table 3.2. The performance of each algorithm in bits per symbol is presented in two columns. The columns marked *raw* show the compression results on the raw data set, and columns marked *pack-4* show that of the packed data. The last column of the table presents the compression performance of *expert model zero* (XM-0) which does not employ any repeat expert. Instead, it employs only a Markov expert and a local Markov expert for compression. XM-0 essentially does not utilise the repetition property of the sequences. The overall compression result (computed as the average of the compression result in each sequence) of each algorithm is presented in the last row.

Of these general compressors, those in the LZ family (GZIP and LZMA) performed poorly on the data set. They all failed to compress the sequences better than the baseline 2 bps. However, they compressed the packed data down to 1.9 bps on average. This is probably because they are designed for compression of ASCII text and thus the dictionaries are designed for larger alphabets. The PPMZ algorithm performed better on the data set, with an average 1.88 bps. However it was inferior to others on the packed data set with an average compression of 1.97 bps. This is because when DNA nucleotides are packed, characters in repeat areas are less likely to be matched. Finally, the performance of CTW algorithm was similar in both configurations.

The performance of the XM-0 algorithm, which does not exploit the repetitive property of DNA, was comparable to the other general text compression algorithms on the data set. Its average compression result was 1.91 bps, which was only outperformed by PPMZ. It should be borne in mind that the XM-0 algorithm uses only two simple Markov models to compress the data – one learned from the global statistics and one from the local statistics. In other words, the general text compression algorithms fail to model repetition in DNA and, because of this, barely compress biological sequences.

Table 3.3 compares the compression results of XM with that of other biological sequence compression algorithms on the data set. XM was configured with a hash table key of size 10. The expert limit was set to 200 and the context for evaluation of experts had size 12. As will be seen later in Table 3.4, this configuration does not produce the best compression. Instead, it compromises the compression performance for the running time. For comparison, the most effective algorithms found in the literature, including BioCompress-2 (BioC) (Grumbach and Tahi, 1994), GenCompress (GenC) (Chen et al., 2000), DNACompress (DNAC) (Chen et al., 2002), DNA2 (Manzini and Rastero, 2004), DNAPack (DNAP) (Behzadi and Fessant, 2005), CDNA (Loewenstern and Yianilos, 1999) and GeMNL (Korodi and Tabus, 2005) are presented. The results for CDNA were reported for only nine sequences and to a precision of two decimal places. The GeMNL results were reported without the sequence HUMHBB and to two decimal places. Higher precision results for GeMNL were obtained by downloading the encoded files from the

Table 3.2: DNA compression by general purpose algorithms.

Sequence	GZIP		BZIP		LZMA		PPMZ		CTW		XM -0
	raw	pack-4	raw	pack-4	raw	pack-4	raw	pack-4	raw	pack-4	
CHMPXX	2.2823	1.8644	2.1223	1.9654	2.0793	1.8782	1.8952	1.9444	1.8231	1.8498	1.8228
CHNTXX	2.3349	1.9525	2.1847	2.0088	2.1582	1.9672	1.9806	2.0015	1.9271	1.9447	1.9257
HEHCMVCG	2.3300	1.9822	2.1687	2.0097	2.1574	1.9858	2.0006	2.0010	1.9502	1.9717	1.9526
HUMDYSTROP	2.3635	1.9500	2.1858	2.0651	2.2110	1.9749	2.0123	2.0061	1.9233	1.9489	1.9131
HUMGHCSA	2.0656	1.7387	1.7305	1.8709	1.1304	1.6650	1.2803	1.8078	1.8827	1.7953	1.9247
HUMHBB	2.2464	1.8976	2.1474	1.9965	2.0587	1.9115	1.9256	1.9846	1.9132	1.9175	1.9025
HUMHDAB	2.2400	1.9159	2.0715	1.9912	2.0118	1.8985	1.9045	1.9865	1.9026	1.9227	1.9174
HUMHPRTB	2.2674	1.9226	2.0936	2.0025	2.0599	1.9241	1.9275	1.9970	1.9171	1.9282	1.9143
MPOMTCG	2.3291	1.9732	2.1737	2.0122	2.1376	1.9808	1.9760	2.0013	1.9634	1.9733	1.9583
MTPACG	2.2926	1.8838	2.1285	1.9839	2.0873	1.9101	1.9190	1.9687	1.8674	1.8816	1.8657
VACCG	2.2520	1.8745	2.0913	1.9518	2.0593	1.8891	1.9046	1.9556	1.8767	1.8795	1.9011
Average	2.2731	1.9050	2.0998	1.9871	2.0137	1.9077	1.8842	1.9686	1.9043	1.9103	1.9089

Table 3.3: DNA compression by special purpose algorithms.

Sequence	DNA2	CTW-LZ	BioC	GenC	CTW-LZ	DNAC	DNAP	CDNA	GeMNL	DNAMem	XM
CHMPXX	1.6733	1.6690	1.6848	1.6730	1.6690	1.6716	1.6602	-	1.6617	1.6601	1.6583
CHNTXX	1.6162	1.6129	1.6172	1.6146	1.6129	1.6127	1.6103	1.65	1.6101	1.6101	1.6103
HEHCMVCG	1.8487	1.8414	1.8480	1.8470	1.8414	1.8492	1.8346	-	1.8420	1.8349	1.8416
HUMDYSTROP	1.9326	1.9175	1.9262	1.9231	1.9175	1.9116	1.9088	1.93	1.9085	1.9084	1.9055
HUMGHCSA	1.3668	1.0972	1.3074	1.0969	1.0972	1.0272	1.0390	0.95	1.0089	1.0311	0.9579
HUMHBB	1.8677	1.8082	1.8800	1.8204	1.8082	1.7897	1.7771	1.77	-	1.7765	1.7499
HUMHDAB	1.9036	1.8218	1.8770	1.8192	1.8218	1.7951	1.7394	1.67	1.7059	1.7395	1.6623
HUMHPRTB	1.9104	1.8433	1.9066	1.8466	1.8433	1.8165	1.7886	1.72	1.7639	1.7884	1.7263
MPOMTCG	1.9275	1.9000	1.9378	1.9058	1.9000	1.8920	1.8932	1.87	1.8822	1.8925	1.8725
MTPACG	1.8696	1.8555	1.8752	1.8624	1.8555	1.8556	1.8535	1.85	1.8440	1.8533	1.8446
VACCG	1.7634	1.7616	1.7614	1.7614	1.7616	1.7580	1.7583	1.81	1.7644	1.7582	1.7683
Average	1.7891	1.7389	1.7838	1.7428	1.7389	1.7254	1.7148	-	-	1.7139	1.6907

author's website. The average compression result of each algorithm is presented in the last row.

With the given parameter configuration, XM outperformed all other algorithms on most sequences from the standard data set. XM's average compression result on 11 sequences was 1.6907 bps while that of its closest competitor, DNAMem, was 1.7139 bps. Since CDNA and GeMNL had missing compression results for several sequences, the same average compression results could not be computed. Instead, the average of only the available results was compared. The average compression result across nine sequences reported for CDNA was 1.6911 bps, while XM's average on the same set was 1.6775 bps. On the ten sequences excluding HUMHBB, GeMNL's average compression result was 1.6980 bps, compared to XM's 1.6848 bps.

Total time for XM to encode these 11 sequences (total 1.3 megabases) was 5.5 seconds. Decoding time was similar since both the encoder and the decoder do substantially the same computation. This is considerably faster than most special purpose compression algorithms to date. Although it is hard to compare XM's running time with other algorithms because either running times were not reported for some algorithms or the hardware capacities were different, a rough comparison is made here. Apostolico and Lonardi (2000) reported a running time of 2-3 minutes for a sequence of 80 kilobases on a 300 Mhz machine. The *CTW-LZ* algorithm (Matsumoto et al., 2000) took 8 minutes to compress HUMDYSTROP (38 kilobases) and several hours to compress HEHCMVCG (229 kilobases) on a slightly faster machine. On a 700Mhz machine, GenCompress (Chen et al., 2000) and DNACompress (Chen et al., 2002) took 53 seconds and 4 seconds respectively to compress HEHCMVCG. These running times are clearly longer than about 0.82 seconds needed by XM to compress the same sequence, albeit a faster machine was used. The authors of NML (Korodi and Tabus, 2005) reported a running time of 6.14 seconds for ten sequences excluding HUMHBB on a Pentium 4 processor running at 2.8 GHz, which is similar to XM in this experiment given the difference in hardware capacity. The fastest reported algorithm appears to be the *DNA2* (Manzini and Rastero, 2004) which compressed the sequence HEHCMVCG (229,354 bases) in 0.42 seconds on a 1Ghz Pentium III processor while XM took twice as long on a much faster machine. However, the compression performance of DNA2 is inferior to other biological special purpose compression algorithms.

In fact, the running time of XM can be reduced by setting the parameters of the algorithm to sacrifice some compression performance. Particularly, the hash key size, k, and the expert limit, L, specify how exhaustively the algorithm searches for matches and hence can control the running time and the compression performance of the algorithm. Generally, the more exhaustively the algorithm searches for matches, the better compression can be obtained and the longer time it takes. To illustrate this, XM was run on the data set using varying values for k and L. The compression performance and running

Table 3.4: Compression results (in bps) and running time (in seconds).

Sequence	L=2, k=8		L=10, k=8		L=50, k=10		L=200, k=10		L=1000, k=8		L=10000, k=5	
	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
CHMPXX	1.6639	0.257	1.6615	0.307	1.6619	0.332	1.6583	0.699	1.6570	3.084	1.6528	5.9041
CHNTXX	1.6122	0.335	1.6125	0.377	1.6114	0.422	1.6103	0.541	1.6110	2.459	1.6081	56.674
HEHCMVCG	1.8454	0.322	1.8433	0.460	1.8429	0.548	1.8416	0.719	1.8410	4.023	1.8372	104.150
HUMDYSTROP	1.9095	0.161	1.9094	0.206	1.9069	0.201	1.9055	0.231	1.9064	0.331	1.9047	4.365
HUMGHCSA	1.1412	0.206	1.0352	0.239	0.9913	0.296	0.9580	0.358	0.9530	0.607	0.9598	11.125
HUMHBB	1.8012	0.217	1.7782	0.246	1.7678	0.285	1.7499	0.343	1.7464	0.603	1.7517	11.941
HUMHDAB	1.7874	0.201	1.7377	0.229	1.7055	0.274	1.6618	0.339	1.6566	0.580	1.6541	7.621
HUMHPRTB	1.8224	0.176	1.7819	0.227	1.7601	0.238	1.7263	0.344	1.7219	0.566	1.7163	7.524
MPOMTCG	1.9180	0.289	1.9060	0.398	1.8939	0.554	1.8725	0.738	1.8690	2.451	1.8647	68.275
MTPACG	1.8592	0.266	1.8544	0.322	1.8518	0.375	1.8446	0.492	1.8404	1.853	1.8363	3.3011
VACCG	1.7726	0.354	1.7702	0.452	1.7698	0.550	1.7683	0.736	1.7683	4.035	1.7654	100.557
Average	1.7394	2.784	1.7173	3.463	1.7058	4.075	1.6907	5.540	1.6883	20.592	1.6865	464.284

times of these configurations are shown in Table 3.4. The results of each configuration are presented in two columns showing the compression results in bits per symbol and the running times in seconds. The configurations are presented in order from the fastest, and hence the worst compression, to the slowest and the best compression.

In the first configuration, the expert limit was set to $L = 2$ and the hash key size was set to $k = 8$. With this configuration, XM took a total of only 2.8 seconds to compress the 11 sequences, and achieved an average compression result of 1.7394 bps, which is as good as most other biological sequence compression algorithms such as BioCompress and GenCompress. The third configuration, which limited to 50 experts and used a hash key size of $k = 10$, achieved an average compression result of 1.7058 bps, which is comparable to the best existing algorithms, in only 4 seconds. The forth configuration took about 5.5 seconds and produced an average compression result of 1.6907 bps, better than any existing algorithm. In the last configuration, which set the hash key size to 5 and the expert limit to 10000, XM achieved better compression results, with an average of 1.6865 bps, at the cost of more time. This took about 8 minutes to compress the total of over one megabase of DNA.

3.5.2 Comparison of Protein Compression Results

Compression of protein sequences has been considered as a challenging problem. The protein alphabet consists of 20 symbols and thus the base line of protein entropy is $log_2 20 = 4.322$ bits per symbol. General text compressors like PPM and GZIP are found not to be able to compress protein sequences and even to *expand* protein data set by a factor of 10% in comparison to the base line.

A special purpose compression algorithm, *Compress Protein* (CP), developed by Nevill-Manning and Witten (1999), takes into consideration the mutation properties of protein sequences. The algorithm uses different contexts for compression as in the PPM algorithm, except that it considers all contexts up to a certain length, and blends the contexts to form a probability distribution. Despite using domain specific knowledge by incorporating substitution matrices, the CP algorithm does not perform significantly better than a simple Markov model. This led to the conclusion of the "incompressibility" of protein.

Probably discouraged by the "incompressibility" claim, very few attempts have been made to compress protein. Only three works on protein compression are found in the literature since then. The ProtComp (Hategan and Tabus, 2004) compresses a protein sequence in two passes. It builds a substitution matrix by searching for *regressor blocks* in the first pass. In the second pass, blocks marked in the first pass are encoded based on the substitution matrix. The *CTW-LZ* algorithm (Matsumoto et al., 2000) uses a context tree weighting approach for protein compression while the BW (Adjeroh and Nan, 2006) algorithm applies the *Burrows-Wheeler transform* (Burrows and Wheeler, 1994) approach.

Table 3.5: Comparison of protein compression.

Sequence	PPM	GZIP	Markov-0	CP	ProtComp	LZ-CTW	XM
HI	4.881	4.672	4.156	4.143	4.108	4.118	**4.102**
SC	4.854	4.640	4.163	4.146	3.938	3.951	**3.885**
MJ	4.734	4.588	4.068	4.051	4.008	4.028	**4.000**
HS	4.639	4.605	4.133	4.112	3.824	4.006	**3.786**
Average	4.777	4.626	4.130	4.113	3.970	4.026	**3.943**

The expert model was also used to compress protein. It was applied to the protein corpus gathered by (Nevill-Manning and Witten, 1999) which consists of the proteomes of four species: *Haemophilus influenzae* (HI), *Saccharomyces cerevisiae* (SC), *Methanococcus jannaschii* (MJ) and *Homo sapiens*(HS). The proteome of a species is the concatenation of all proteins of the species. As an amino acid is coded by three nucleotides in DNA, a shorter hash key, of length 6 was used. Table 3.5 shows the compression results of the discussed algorithms on the four protein sequences. Note that an incorrect protein corpus that was more compressible somehow got into circulation at some point, resulting in significantly lower compression figures being reported by ProtComp (Hategan and Tabus, 2004) and BW (Adjeroh and Nan, 2006). The compression results of ProtComp on the *correct* protein corpus were obtained from the author's website. The authors of BW have "moved to new projects" (Nan, 2006) so proper compression results for BW are not available.

The two general text compressors, PPM and GZIP, performed poorly on the four proteomes, at an average of 4.777 bps and 4.626 bps respectively, much worse than the base line 4.322 bps. The simple order-0 Markov model achieved an average of 4.130 bps, only slightly better than the baseline. The CP algorithm improved on the order-0 Markov model by very little, to a 4.113 bps average. The three later algorithms, ProtComp, LZ-CTW and XM, performed much better than CP – by more than 0.1 bps. Among the three, XM marginally outperformed the others on all four sequences.

3.5.3 Compressibility of Genomes

The standard DNA data set presented in Subsection 3.5.1 was designed to compare early biological sequence compression algorithms, most of which are unable to handle long sequences. The data set is clearly not suitable for the current situation. Biological databases are now much larger than before, and sequences are much longer. Modern biological sequence compression algorithms should be able to work on much longer sequences. This work proposes new data sets for comparison of present and future compression algorithms. These new data sets contain sequences that reflect the sizes of sequences available in biological databases today. The sequences also reflect the diversity of life: they are extracted from various species levels, and also have different degrees of compressibility.

Table 3.6: Compression of the human genome.

Sequence	Length	MNL-1	XM - 200		XM - 500		XM - 1000		XM - 5000		XM - 10000	
	(Mb)	Rate	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
Chr 1	218.71	1.6440	1.6259	1h34m	1.6128	3h22m	1.6055	6h33m	1.5956	18h10m	1.5940	31h04m
Chr 2	237.04	1.6640	1.6464	1h42m	1.6345	3h40m	1.6279	7h05m	1.6187	20h57m	1.6174	37h52m
Chr 3	193.61	1.6720	1.6517	2h03m	1.6397	4h35m	1.6331	8h33m	1.6246	23h20m	1.6234	36h01m
Chr 4	186.58	1.6530	1.6337	1h20m	1.6219	2h52m	1.6154	5h17m	1.6075	14h56m	1.6064	23h07m
Chr 5	177.52	1.6500	1.6300	1h51m	1.6188	4h10m	1.6126	7h53m	1.6050	20h24m	1.6040	30h07m
Chr 6	166.88	1.6640	1.6432	1h41m	1.6315	3h44m	1.6251	6h40m	1.6174	17h43m	1.6164	26h38m
Chr 7	154.55	1.6140	1.5927	1h08m	1.5805	2h31m	1.5739	4h27m	1.5661	11h28m	1.5650	18h53m
Chr 8	141.69	1.6700	1.6521	1h27m	1.6408	3h17m	1.6345	5h47m	1.6273	14h00m	1.6263	20h45m
Chr 9	115.19	1.6080	1.5891	0h45m	1.5787	1h37m	1.5730	2h49m	1.5667	6h08m	1.5658	9h55m
Chr 10	130.71	1.6410	1.6252	1h19m	1.6140	2h58m	1.6076	5h10m	1.6004	12h15m	1.5994	19h06m
Chr 11	130.71	1.6470	1.6280	0h52m	1.6166	1h53m	1.6105	3h19m	1.6036	7h35m	1.6027	12h19m
Chr 12	129.33	1.6530	1.6322	1h19m	1.6201	2h55m	1.6135	5h09m	1.6058	12h24m	1.6046	19h27m
Chr 13	95.51	1.6890	1.6721	0h53m	1.6624	1h57m	1.6572	3h18m	1.6522	6h55m	1.6243	9h36m
Chr 14	87.19	1.6670	1.6476	0h51m	1.6367	1h51m	1.6309	3h04m	1.6251	6h21m	1.5768	9h49m
Chr 15	81.12	1.6180	1.5992	0h31m	1.5889	1h07m	1.5834	1h48m	1.5777	3h35m	1.5768	6h02m
Chr 16	79.89	1.5740	1.5573	0h32m	1.5458	1h08m	1.5396	1h54m	1.5329	4h03m	1.5318	7h39m
Chr 17	77.48	1.5990	1.5812	0h45m	1.5687	1h36m	1.5622	2h36m	1.5544	6h20m	1.5530	11h36m
Chr 18	74.53	1.7090	1.6932	0h28m	1.6841	1h00m	1.6795	1h33m	1.6752	2h55m	1.6747	4h09m
Chr 19	55.78	1.4820	1.4533	0h30m	1.4368	1h03m	1.4285	1h08m	1.4181	4h47m	1.4160	6h46m
Chr 20	59.42	1.6940	1.6777	0h33m	1.6674	1h08m	1.6622	1h43m	1.6569	4h05m	1.6562	5h36m
Chr 21	33.92	1.7010	1.6806	0h16m	1.6728	0h30m	1.6698	0h41m	1.6668	1h13m	1.6666	1h37m
Chr 22	34.35	1.6100	1.5886	0h16m	1.5785	0h31m	1.5739	0h45m	1.5683	1h59m	1.5675	3h15m
Chr X	147.69	1.5500	1.5241	1h01m	1.5085	2h15m	1.4999	4h02m	1.4906	10h24m	1.4897	15h55m
Chr Y	22.76	1.1490	1.1213	0h07m	1.1159	0h12m	1.1143	0h15m	1.1133	0h23m	1.1132	0h26m
Average	2832.18	1.6176	1.5978	23h55m	1.5865	52h04m	1.5806	91h40m	1.5738	232h33m	1.5697	367h51m

Table 3.7: The compressibility of various genomes.

Species	Type	Relevance	Genome Size	Known bases	Bps	Time
Human immunodeficiency virus	Virus	Virus causes AIDS	9181	9181	1.8907	430
Thermoplasma volcanium	Archaea	Archaea	1584804	1584804	1.9279	6362
Methanococcus jannaschii	Archaea	Archaea	1664970	1664957	1.8129	11054
Haemophilus influenzae	Bacteria	Bacteria	1830138	1830023	1.8776	10917
Bacillus licheniformis	Bacteria	Bacteria	4222645	4222645	1.9135	37212
Escherichia coli	Bacteria	Bacteria	4643538	4643537	1.9079	41302
Mycobacterium tuberculosis	Bacteria	Bacteria	4419977	4419977	1.8339	57633
Mycobacterium avium	Bacteria	Bacteria	4829781	4829781	1.8021	82574
Saccharomyces cerevisiae	Baker's yeast	Single cell eukaryote	12156679	12156679	1.8176	210304
Dictyostelium discoideum	Slime mold	Model organism	33928503	33906464	1.5061	626124
Plasmodium falciparum	Parasitic protozoan	Malaria parasites	23264338	23263391	1.5126	430254
Anopheles gambiae	Mosquito	Vector of malaria	230466657	225028590	1.7483	6850454
Drosophila melanogaster	Fruit fly	Model animal	120381546	120290946	1.8332	2408892
Caenorhabditis elegans	Nematode worm	Model animal	100269917	100269917	1.7053	2047436
Arabidopsis thaliana	Wild mustard	Model plant	119186497	118960067	1.6587	2576292
Vitis vinifera	Grapevine	Fruit crop	303085820	290237009	1.4037	6199596
Oryza sativa	Rice	Crop & model organism	370792118	370733456	1.3494	8532162
Ciona intestinalis	Sea squirt	Simple chordate	173499994	141233565	1.3972	3340646
Tetraodon nigroviridis	Puffer fish	Compact genome	358601784	302298326	1.7745	6495504
Gallus gallus	Chicken		1100463666	1042566360	1.7573	43576727
Mus musculus	Mouse	Model mammal	2654895218	2558509480		
	Mouse Chromosomes 1-5		846710681	825871442	1.6106	17946492
	Mouse Chromosomes 6-12		930951274	896762952	1.6124	28267991
	Mouse Chromosomes 13-29,X and Y		877233263	835875086	1.5505	26545914

A natural choice for a data set is the human genome. The 22 autosomes and two sex chromosomes were obtained from GenBank Release 36 (NCBI, 2003). Wildcards – N, R, Y, etc – were deleted. The longest sequence, chromosome 1, is about 218 megabases and the shortest is the Y sex chromosome with 22 megabases.

Table 3.6 shows the compression of 24 human chromosomes using the expert model. The second column gives the length of the chromosomes in megabases (Mb). Different expert limit values, including 200, 500, 1000, 5000 and 10000, were tried. Each expert model configuration is presented in two columns, one giving the compression result and the other giving the running time. The table also shows the compression of the same data by another compression algorithm, NML-1 (Korodi and Tabus, 2007), which is the only other whole genome compression algorithm found in the literature. The last row shows the average compression results and the total running time of each configuration.

The XM-200 (XM with maximum 200 experts) configuration took about one day on a 2.33 Ghz processor (1 CPU-day) to compress the chromosomes while the NML-1 was reported to take 3.5 hours on a cluster of 12 workstations with 3.2 GHz processors (Korodi and Tabus, 2007) (1.75 CPU-days). Not only being faster, XM-200 also outperformed the NML by about 0.02 bps on every chromosome. The XM-10000 configuration performed even better. It made another improvement of 0.03 bps over the XM-200 configuration at the cost of more running time (over 15 CPU-days).

The second data set proposed by this thesis contains the genomes of 18 species from different organism levels including bacteria, archaea, single cell eukaryotes, worms, plants and vertebrates. The genomes were downloaded from Genbank (Benson et al., 2009), except for the genome of *Plasmodium falciparum* which was obtained from the PlasmoDB database (PlasmoDB, 2009b). The compression of these genomes by the expert model is shown in Table 3.7. As the current implementation of the expert model and the hardware available in this project are only capable of handling sequences of up to 1 billion bases, the mouse genome is therefore split into three parts. The lengths of these genomes are shown in the second column of the table. The expert model in this experiment was run with hash key size 11 and expert limit 200. The compression results in bits per symbol and the running times are given in the third and fourth columns respectively.

3.5.4 Information Content of Sequences

Not only does the expert model outperform existing biological sequence compression algorithms in terms of both speed and compression, but it also can provide an estimate of the information content of each symbol. Such a sequence of information content under a compression model has been found to be useful for many knowledge discovery tasks (Dix et al., 2007). While some previous methods such as the ARM model (Allison et al., 1998) are capable of producing the information content sequence, they are very slow

and hence cannot be practically applied to analysing long sequences. Having a practical compression algorithm like the expert model, therefore is important.

Figure 3.2 shows a graph of information content along the HUMHBB sequence, produced by the expert model. The data in the graph is smoothed with a window size of 300 for viewing purposes. One can notice spikes in the graph corresponding to areas of repeats in the sequence. Each significant spike corresponds to a repeat element in the sequence. A more detailed analysis of the information content sequence of HUMHBB is presented in Section 3.6.

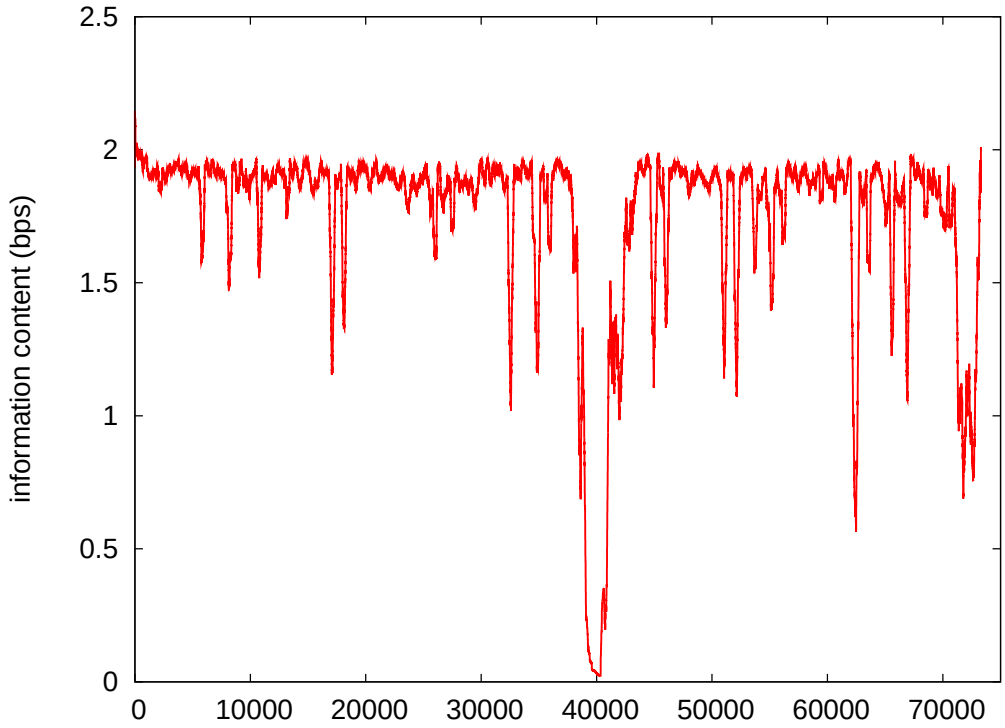


Figure 3.2: The estimated information content of HUMHBB sequence.

Figure 3.3 presents a *dot matrix* (Gibbs and Mcintyre, 1970) that shows the two dimensional plots of repeat elements from the HUMHBB sequence detected by XM. The dot matrix shows not only where a repeat element is but also where it is repeated from. Each pair of repeat elements is presented by a diagonal line of dots. A diagonal line parallel to the primary diagonal (going down to the right) represents a pair of forward repeats, while a line along the secondary diagonal (going up to the right) shows a pair of reverse repeats. The projection of a diagonal line on the two axes shows the positions of the pair in the sequence. From the plot, it is easy to see a long forward repeat element near the center of the sequence. A smaller region near the end of the sequence is a reverse repeat from two earlier locations.

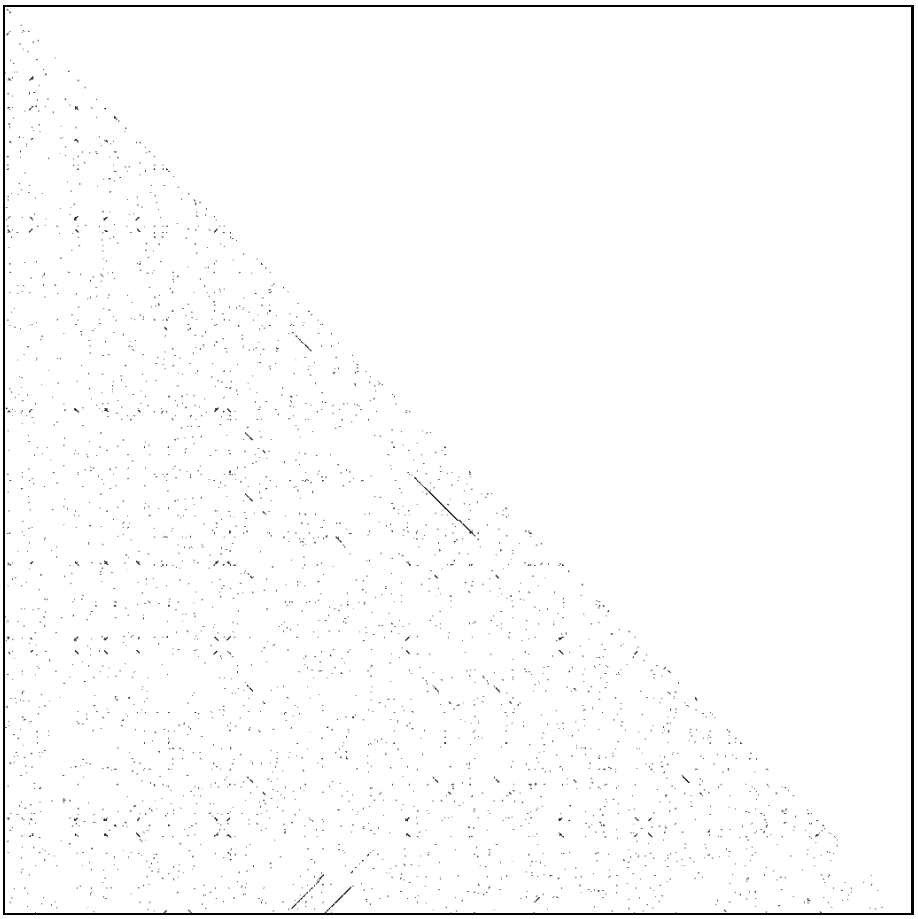


Figure 3.3: A dot matrix plot of the HUMHBB sequence.

3.5.5 Conditional Compression of Sequences

The compression of a sequence depends on the context, that is the background knowledge of the compressor. The conditional information content of a sequence on the background of a particular context shows the amount of information about the sequence that is not contained in the context. Compression in a context related to the sequence results in low conditional information content. The conditional information content of a sequence in a context, therefore, is an indication of how related the context is to the sequence.

The expert model is able to estimate the conditional information content of a sequence in a context. An experiment was performed to illustrate this. In this experiment, the human chromosome 22 was compressed on the background of the chimpanzee genome, the mouse genome and the chicken genome, respectively. For the context of the chimpanzee genome, only the chimpanzee chromosome 22 was used since it corresponds to the human chromosome 22. The human chromosome 22 is thought to be mapped to parts of chromosomes 5, 6, 8, 11 10, 15 and 16 in the mouse genome (Dunham et al., 1999) due

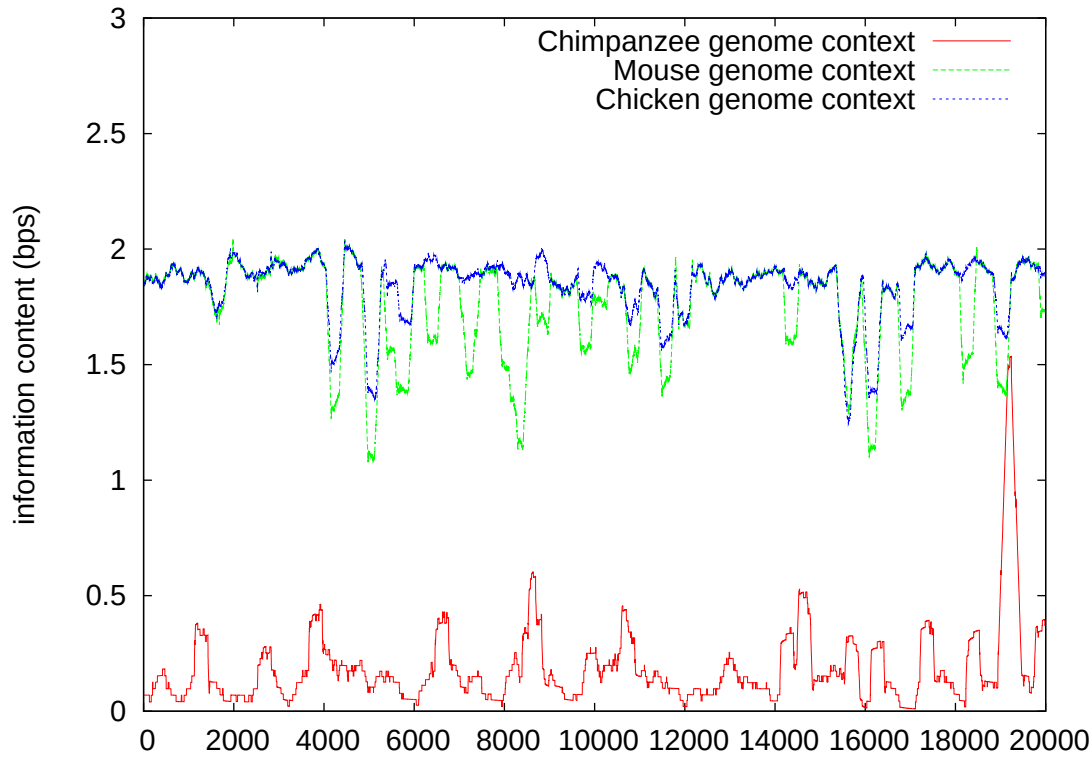


Figure 3.4: The estimated conditional information content of a region on the human chromosome 22 in differing contexts.

to chromosomal translocation. Therefore, these mouse chromosomes were used as the context of the mouse genome. Finally, the human chromosome 22 was compressed in the context of the whole chicken genome.

Figure 3.4 presents the estimated conditional information content of a region (from position 40,050,000 to position 40,070,000) in the human chromosome 22 in these three contexts. Since chimpanzee is the most related to human among the three species, the conditional information content of the human genomic sequence in the context of the chimpanzee genome is very low, mostly under 0.5 bps. The conditional information content in the context of the mouse genome is lower than that in the context of the chicken genome in several areas.

The conditional information content obtained by compression of one sequence on the background of another can be used to estimate the shared information content of the two sequences. Among the human chromosomes, the two sex chromosomes show a great deal of mutual information. The information content of chromosome X is estimated to be $1.5085 * 147.69 = 222.7903$ megabits. If chromosome X is compressed on the background of chromosome Y, it is compressed to 1.465440 bits per symbol. In other words, its conditional information content given chromosome Y is estimated to be

$1.465440 * 147.69 = 216.430833$ megabits. The shared information content between the two sequences from the estimation is $222.7903 - 216.430833 = 6.359467$ megabits.

Another way to estimate the shared information content of the two sex chromosomes is to compress chromosome Y instead of chromosome X. The information content of chromosome Y is $1.1159 * 22.76 = 25.3978$ megabits while the conditional information content of chromosome Y given chromosome X is $0.832073 * 22.76 = 18.937981$ megabits. In other words, the shared information content between the two chromosomes is $25.3978 - 18.937981 = 6.459819$ megabits. The two values differ, by 1.5%, because of arithmetic rounding and the random features of the algorithm (e.g., selecting experts randomly). In some applications such as for phylogenetic analysis (see Chapter 5), the average of the two values is used for the mutual information content of two sequences.

3.6 A Side Application: Repeat Detection

Repeat elements are abundant in eukaryotic genomes. As much as 55% of the human genome is made up of repeat elements (Lander et al., 2001). Many human diseases such as *Huntington's disease* and *Schizophrenia* are associated with repeat elements in the genome (Buard and Jeffreys, 1997). Locating repeats in the genome is therefore, very important. It is, however, very challenging. Long sequences over a small alphabet contain many random matches while most "genuine" repeat elements are approximate. For example, each Alu repeat element in the human genome is only 70% – 80% similar to each other (Deininger et al., 1992; Mighell et al., 1997). In other words, one would expect a mutation every 5 symbols on average in an Alu repeat element whereas any 5-mer could be expected to occur randomly every $4^5 = 1024$ bases and thus to occur about 3 billion times in the human genome just by chance.

Many biological sequence compression algorithms rely on locating repeat elements. They use a special coding scheme to compress repeated areas. The expert model, on the other hand, does not require the explicit identification of repeats from non-repeats nor a separate compression scheme for them. Instead, it blends the encoding of non-repeat areas and that of repeat areas. A substantial repeat would result in a repeat expert that compresses significantly better than the Markov experts on the region. Therefore, a region is considered as a repeat if it can be compressed better by combined experts than by only the Markov experts.

The expert model is able to estimate the information content (Dix et al., 2007) of every symbol in the sequence. The information content of a symbol reflects how well the model predicts the symbol based on available contexts i.e., the background knowledge known by the model. The information content sequence produced by the Markov experts is based on the knowledge from Markov models. By combining Markov experts

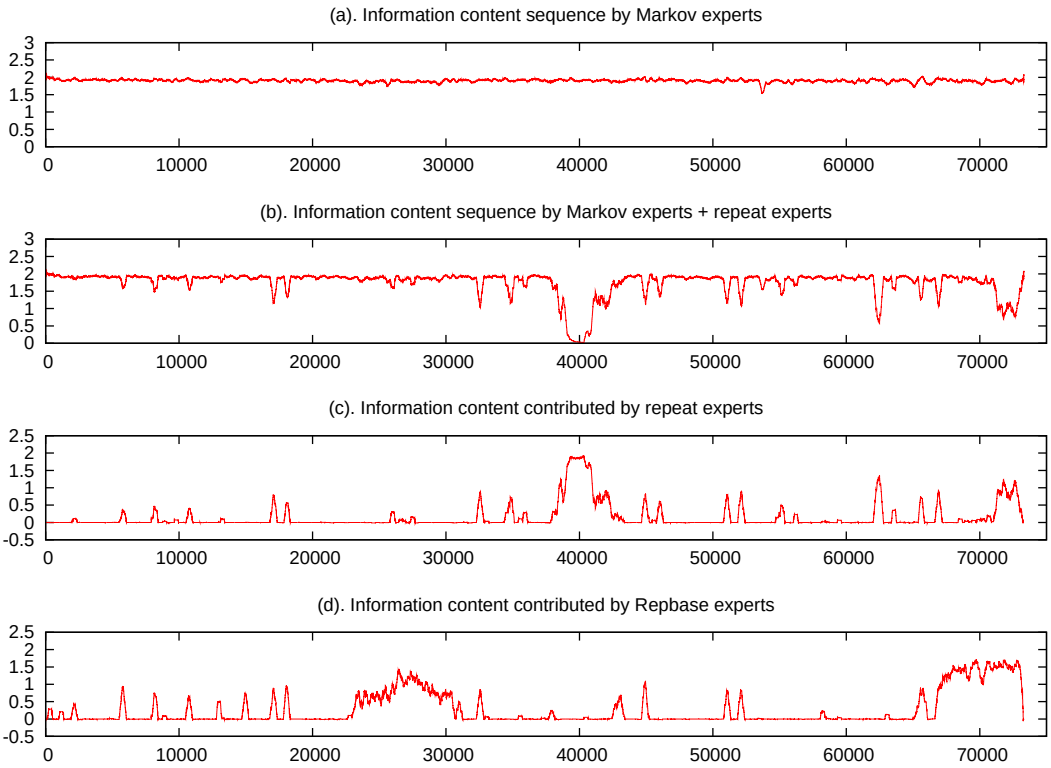


Figure 3.5: The information content sequences of HUMHBB produced by various experts.

with other knowledge in the forms of other experts, the expert model can produce the information content sequence given different sources of knowledge. The *difference* in information content with and without additional experts shows the contribution of these experts in prediction of the sequence.

Figure 3.5(a) shows the graph of information content along the HUMHBB sequence produced by the two Markov experts. Their predictions are only slightly better than the baseline 2 bits per symbol. The information content of HUMHBB produced by combining the Markov experts with repeat experts, is shown in Figure 3.5(b). The difference of the two information content sequences is given in Figure 3.5(c). This shows the amount of information contributed by repeat experts that perform prediction based on repetition. In other words, this information is gained by modelling repetition in the sequence. Each spike in the plot corresponds to a region that is a repeat of an earlier region. This is an example of finding repeats *ab initio*, e.i., discovering patterns that are repeated in a sequence.

The expert model can also be used to find repeat elements by compressing a sequence on the background knowledge of a curated library of precomputed motifs such as the RepBase (Jurka et al., 2005) database which is a collection of repeat elements from the human genome. The HUMHBB sequence is compressed by combining the Markov

experts with *RepBase experts* which are repeat experts that base their prediction on the knowledge from RepBase. Figure 3.5(d) shows the information gained by having RepBase as the background knowledge. The spikes in the plot show the regions that are approximate copies of some elements in RepBase.

Interestingly, the previous *ab initio* repeat finding exercise locates several repeat regions in HUMHBB that are not included in RepBase. These regions correspond to spikes in Figure 3.5(c) but not in Figure 3.5(d). One can notice a region of about 5 kilobases long starting at around position 38000 and a shorter region of about 700 bases at position 62000.

3.7 Discussion

The expert model is very simple and has few parameters. This is useful for modelling genomic and proteomic sequences, which is the main purpose of this research. Most parameters such as the hash key size and expert limit are for controlling the compromise between the speed and the compression performance of the algorithm. Generally, a small hash key size and a large expert limit enable the algorithm to propose more expert candidates and thus improve the compression quality but at the cost of longer running time.

Assuming constant-size integers, the expert model has linear space complexity with a very low constant: The algorithm needs to store all positions in the hash table as well as the sequence for references. Storing a nucleotide requires 2 bits and storing a position in a sequence of length n needs $log_2 n$ bits. Some extra memory is needed for processing at a position, but is bounded by the expert limit, which is small in comparison to n. In other words, the space complexity of the algorithm is $O(n(log_2 n + 2))$ bits. In practice, for the sake of speed and simplicity, the implementation uses an integer (4 bytes) to store a position and a byte to store a symbol. Therefore, the space required for compression of a sequence of length n is $O(5n + LC)$ where L is the expert limit and C is a constant. There are also two bit-arrays, each of size n, to mark if a certain copy or reverse expert has been employed to further facilitate management of repeat experts. The program therefore actually uses $O(5.25n + LC)$ bytes of memory for compression of a DNA sequence of length n. For example, the program requires 6GB of memory to compress a sequence of 2^{30} bases.

The expert model compress a sequence by scanning the sequence and, at a position, it consults the hash table for potential experts. It recruits experts up to the limit if there are sufficient candidates. If a position is part of a repeat, a repeat expert is potentially employed. The compressor therefore, has to process as many as the sum of the lengths of all repeat elements, $\sum |r|$ where $|r|$ is the length of a repeat element r. Furthermore, with an alphabet of size $|\mathcal{A}|$, a k-mer would be expected to occur every $|\mathcal{A}|^k$ positions at

random, assuming the composition distribution is uniform. Hence, a hash key of size k would suggest an average $\frac{n}{|\mathcal{A}|^k}$ random experts at a position. However, the expert limit parameter L restricts the maximum size of the expert panel. Therefore, the time complexity of the algorithm is $O(n(min\{\frac{n}{|\mathcal{A}|^h}, L\}) + \sum |r|)$. $\sum |r|$ is in fact smaller than n and thus the worse case complexity of the algorithm is $O(Ln)$.

The expert model is a fully adaptive compression algorithm. The model does not need to be explicitly transmitted to the decoder. Instead, identical models are maintained by both the encoder and the decoder during the compression/decompression process. The algorithm actually maintains a set of models (or sub-models) in the forms of experts. Not only does each expert adapt to fit to the data stream, but the algorithm also adaptively adjusts each expert's weight to obtain the optimal blending given the data seen so far. It has been shown that adaptive modelling is superior to non-adaptive schemes despite being simple and elegant (Cleary and Witten, 1984a). This is consistent with the superiority of the expert model over most existing biological sequence compression algorithms that are not fully adaptive.

Importantly, the expert model presents a new mechanism for combining models for prediction and compression which has not been properly considered before. Having several predictive models for compression of texts from several data sources (such as files containing both English texts and program source code) has been encountered in many works (Bell et al., 1989). To the best of the author's knowledge, there has not been a sound and proven approach to the problem so far. Blending models is presented in many text compression works, but so far only trivial or ad-hoc approaches are applied. The PPM algorithm (Cleary and Witten, 1984b), for example, maintains models of variable context lengths and, instead of blending these models for prediction of the next symbol, it attempts to find the model with the longest context in which the symbol has been seen. On the other hand, the *context tree weighting* algorithm (Willems et al., 1995) uses a simple way to blend models; models are organised into a binary tree and the weight of a node is the average weight of its children. These two methods can only be applied to variable context Markov models. They are not easily generalised to different types of models and to the problem of multiple modal sources. An attempt to combine models is presented in *Multi-Modal Data Compression* (Williams, 1991) in which, rather than blending models, the best model over a history is selected.

The approach used in the expert model presents a theoretically sound method for combining models which is based on the well-founded Bayesian framework. The combination of Markov models and repeat models was demonstrated in this work. Moreover, the framework of the algorithm can be generalised to different kinds of models, especially models for different data sources. A similar approach to the expert model has been presented in the image compression algorithm TMW (Meyer and Tischer, 1998), which is the state of the art for grey-scale image compression.

Existing biological sequence compression algorithms often separate the sequence into repeat and non-repeat areas, and encode these areas using differing schemes. Repeat areas are often compressed by some specific scheme such as by referring to an earlier copy (Grumbach and Tahi, 1993) or by using a regressor (Korodi and Tabus, 2005). Non-repeat areas are generally encoded using the naive two bits per symbol approach or a Markov model, typically of order 2. Since biological repeats are approximate and there are a great many random matches, it is a philosophical question as to when a region is considered to be a repeat. Especially, when the composition distribution of the sequence is skewed, spurious matches tend to occur more frequently than intuition suggests. These approaches, therefore, are not robust for a wide range of sequences.

The expert model, on the other hand, does not require the explicit identification of repeats from non-repeats or a separate compression scheme to encode them. It in fact blends the encoding of non-repeats and the encoding of repeats based on the effectiveness of each encoding in the recent history. A substantial repeat would result in a repeat expert that significantly outperforms the Markov experts on the region. Therefore, a region is considered a repeat if it is compressed better by combining experts than by only the Markov experts. As described in Section 3.6, instead of finding repeats for compression, the expert model performs compression to detect repeats. This is an attractive feature of the expert model for performing various data mining tasks on biological data where statistical biases in data could mislead other methods. Chapters 4 and 5 discuss applications of the expert model in more details.

The expert model presented here is a general framework that can be extended in many directions. In particular, the hash table used to suggest repeat experts can be implemented in different ways for different types of data. Several mutation models can be used by repeat experts for symbol prediction. The extendibility of the framework provides the flexibility for the algorithm to be applied to a wide range of data.

Different models of biological sequences can be integrated in the framework. Section 3.4 demonstrates that Markov models and repeat models can be combined to give good compression of biological sequences. Any models that can make use of a history to estimate the probability of the next symbol can be used in the framework to improve the compression performance of the algorithm.

The framework also allows the incorporation of "prior knowledge" for compression. Prior knowledge can be in the forms of models or other related sequences. Most importantly, sequence compression using the expert model can be used to measure the "usefulness" of the prior knowledge to the sequence being compressed. If a model can help to better compress a sequence, the model is well suited to the sequence. Likewise, if prior knowledge from another sequence improves the compression of the sequence, the two sequences can be considered related. The relatedness of the two sequences can be measured

by the improvement in compression. This feature is desirable for many knowledge discovery tasks on biological data. Later chapters present a number of applications of using the expert model in biological data mining tasks such as sequence alignment (Chapter 4) and phylogenetics analysis (Chapter 5).

3.8 Summary

This chapter has presented the expert model, a simple and yet effective algorithm for biological sequence compression. The algorithm utilises approximate repeats and statistical properties of biological sequences for compression. It is shown to outperform all published DNA and protein compressors to date while maintaining a practical running time. the expert model is capable of compressing sequences in length of up to a billion bases. As a result, it can be applied to compress the genomes of various organisms ranging from virus to animals.

The expert model presents an adaptive compression technique that blends various predictive models for compression. Though the expert model is designed specifically for compression of biological sequences, the compression framework can be generalised for other types of data. Further research could investigate the use of the framework for compression of other types of data, especially data from different sources such as texts in different languages.

So far, only Markov models and repeat models are employed in the expert model for compression. Many parts of a genomic sequence may have specific structures and statistical distributions. For example, a gene generally starts with a *promoter* region, followed by alternating exons and introns. A specific expert, such as a *gene expert* which has some built in knowledge about such structures might be employed for better prediction in these parts.

The expert model can estimate the information content of every symbol of a sequence in different contexts. By examining the information content sequence of a genomic sequence, one can locate patterns of similarity between the sequence and the context. This is useful for many sequence analysis tasks such as repeat detection as shown in Section 3.6. Finding regions of similarity between sequences is related to the *sequence alignment* problem. Furthermore, the ability of handling long sequences allows the expert model to be used for analysing whole genomes. Chapter 4 shows an approach to genomic alignment based on the information content calculated by the expert model. The approach is shown to be very effective, especially for analysing distantly related sequences, and sequences with statistical biases.

The size of a haploid genome was once thought to be correlated to the complexity of the species (Vendrely and Vendrely, 1948). This however, is found not to be correct as

genome sizes vary greatly among species and have no relationship with the number of genes in the genomes (*C-value paradox* (Gregory, 2001)). Since a genome is the carrier of genetic information of a species, it is stipulated that the *amount of information* in a genome reflects the complexity of the species. The virtues of the expert model, i.e., good compression performance and capability to handle long sequences, allow the close estimation of the amount of information contained in a genome as shown in Subsection 3.5.3. Future research could investigate if the information content of a haploid genome can be an indication of the complexity of the organism.

Compression can also be used to estimate the mutual information content of any two genomes which can be an indication of the relatedness between the two organisms and is useful for phylogenetic analysis. Chapter 5 presents the investigation of using the expert model for phylogenetic analysis from whole genomes.

Chapter 4

Genomic Sequence Alignment by Compression

He that is good with a hammer tends to think everything is a nail.
–Abraham Maslow

4.1 Introduction

Advances in sequencing technology allow high throughput production of biological sequences in laboratories around the world. The exponential increase in genomic data extracted recently introduces a need for analysis techniques that can handle the large amount of data. This is very challenging as conventional analysis methods can be overwhelmed by volume and misled by statistical biases. It is important to develop new and novel tools for analysing such data. The tools need to be time efficient and able to cope with the diversity of the data.

One of the most important tasks in sequence analysis, if not the most important one, is sequence alignment which attempts to arrange two (or more) biological sequences to identify regions of similarity. Similarities between sequences can provide clues to the discovery of the evolutionary relationship between species, to annotate new sequences, and to compare an unknown sequence against existing sequences in a large database. There are two broad kinds of sequence alignment, namely *global alignment* and *local alignment*. Global alignment attempts to match entire sequences from end to end and thus is suitable for comparing short sequences that are expected to have similar structures and functions such as proteins or genes. On the other hand, local alignment searches for conserved regions, possibly *reordered*, between two sequences. Local alignment is more suitable for analysing long sequences, such as chromosomes or genomes, especially from distantly

related species where significant insertions, deletions and rearrangements may have occurred.

Most alignment methods are based on the dynamic programming algorithm. The Needleman-Wunsch (Needleman and Wunsch, 1970) and Smith-Waterman (Smith and Waterman, 1981) algorithms are two early examples for global alignment and local alignment respectively. The quadratic time complexity of dynamic programming is acceptable for short sequences – as most sequences were in the early days of bioinformatics. However, it is infeasible to use dynamic programming algorithms on large chromosomes and genomes that are millions or even billions of bases long. Recent algorithms achieve faster speed by using heuristics. They normally first search for short matches, called *seeds*, using some indexing techniques such as hash tables or suffix arrays and then extend the seeds using dynamic programming. They are often insensitive, especially when aligning distantly related sequences.

Most traditional alignment methods rely heavily on a scoring scheme that is based on a substitution matrix, to describe the mutation rates between nucleotides or amino acids, and on other parameters such as gap penalties. However, these methods lack a well-principled objective function to measure the performance of a set of parameters: There is considerable disagreement among biologists about the "right" choice of parameters (Gusfield et al., 1992). Using a generic substitution matrix may be suitable for protein alignment as the rates of substitution in protein largely depend on the similarities between amino acid properties which are well understood. However, this is not the case in nucleotides; more than one codon can code for an amino acid and different strains show different codon preferences for a given amino acid (Comeron and Aguade, 1998). It is therefore sometimes very hard to find a suitable scoring scheme for alignment of genomes, especially when little is known about the sequences. The selection of a scoring scheme would be managed easily with a reasonable objective function.

The presence of low information content regions such as repetitive and statistically biased DNA in genomes is problematic for genomic alignment. It is estimated that the human genome contains more than 50% of repeat DNA and about 30000 CpG islands which are genomic regions containing a high frequency of C and G (Lander et al., 2001). Some genomes are extremely statistically biased. The genome of the malaria parasite *Plasmodium falciparum*, for example, contains 80% of A and T, and even more in introns and intergenic regions. Such sequences of biased composition and low information can give rise to both false-positives and false-negatives from alignment algorithms (Powell et al., 2004).

Existing alignment algorithms consider sequence alignment as a variation on the edit distance problem, and perform alignment by matching characters of the two sequences. As a result, they are unable to deal with regions of low information content

such as repetitive and statistically biased DNA. Such regions are often "masked out" before alignment (Wootton and Federhen, 1993; Wootton, 1997). Since genomic sequences are meant to convey genetic *information*, a new alignment methodology that performs alignment at the *information level* is proposed here. The methodology is based on information theory (Shannon, 1948) and the *Minimum Message Length* principle (Wallace and Boulton, 1968; Wallace, 2005). This approach considers regions that convey similar information as potential homologues. The similarity of regions can be measured by their mutual information content.

This chapter presents XMAligner, a novel method for genomic local alignment based on information theory. XMAligner makes use of the expert model compression algorithm (Cao et al., 2007) (presented in Chapter 3) to estimate the information content, and mutual information content, of the two sequences to be aligned. Since XMAligner performs alignment at the information level, it does not require masking out of repetitive and low information regions. It has an objective function to help in selecting parameters for a good alignment. The method is shown to be practical and can handle sequences of hundreds of millions of bases.

The chapter is organised as follows. Section 4.2 presents a review of existing genomic alignment methods. Section 4.3 presents the XMAligner approach. Experiments for comparison of XMAligner with several existing alignment methods are described in Section 4.4. The visualisation of alignment generated by XMAligner is shown in Section 4.5. Section 4.6 presents a side application of XMAligner to compute a substitution matrix between two genomes. Section 4.7 presents a discussion of XMAligner, and Section 4.8 concludes the chapter.

4.2 Related Works

Since alignment is the most basic tool for sequence analysis, much research has been done in this field. Most alignment methods are inspired by dynamic programming (Needleman and Wunsch, 1970; Smith and Waterman, 1981). These methods attempt to examine all possible pairings of the two sequences and choose the optimal alignment which has the best matching score. They have been used extensively, primarily for comparing protein sequences or short DNA sequences.

These alignment approaches have time and space complexity of $O(mn)$ when aligning two sequences of lengths m and n, and hence are unattractive for aligning long sequences. They became infeasible for many applications in the early 1990s when the lengths of sequences, and the sizes of databases, increased. To trade sensitivity for running time, heuristic search methods are used. For example, instead of comparing every single base of the two sequences, FASTA (Pearson and Lipman, 1988) and BLAST

(Altschul et al., 1990), the two most popular database search tools, search for seeds of k consecutive exact matches. Seeds are then extended, by limited dynamic programming, to allow for mutations and gaps.

Since 1995 when the first genome of a free-living organism was sequenced (Fleischmann et al., 1995), there has been much research on tools that are capable of comparing genomes. Most alignment methods rely on the ideas of FASTA and BLAST; they use different techniques for finding seeds and for extending seeds to identify conserved regions. A hash table is used in SSAHA (Ning et al., 2001) for locating seeds which are matched k-tuples; the seeds are then sorted and linked together. Gapped BLAST (Altschul et al., 1997) and BLASTZ (Schwartz et al., 2003) find seeds of short, near exact matches. These methods first extend seeds without gaps. Each gap-free match that is longer than a certain threshold is then extended by dynamic programming. The BLAST Like Alignment Tool, BLAT, (Kent, 2002) works in a similar way to BLAST to find seeds and then clumps similar regions together to form larger regions. Similarly, the CHAOS algorithm (Brudno et al., 2003) chains together sufficiently near seeds, and reports statistically significant chains as homologues.

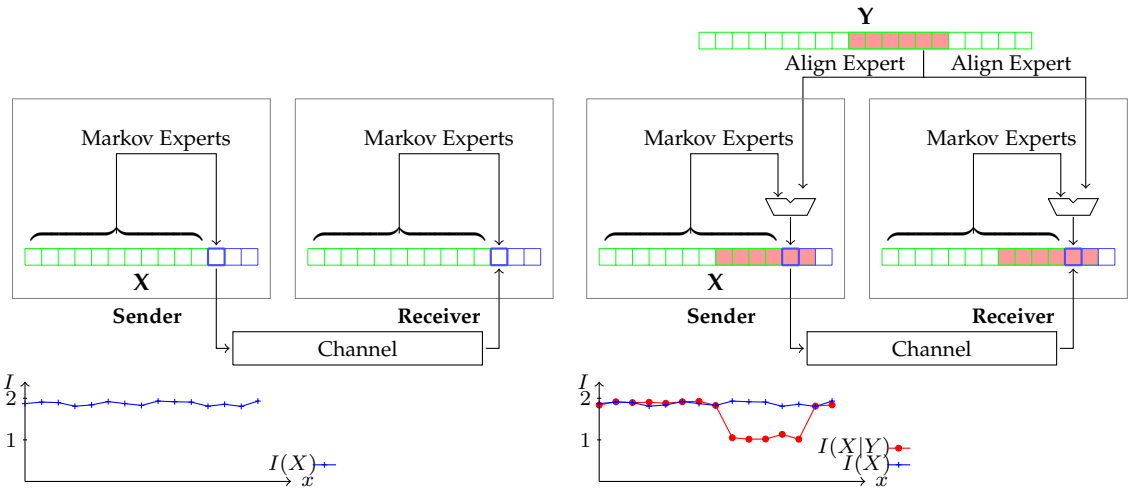
A number of genome alignment tools make use of suffix trees (Weiner, 1973; McCreight, 1976) and suffix arrays (Manber and Myers, 1991) to find seeds, rather than a hash table – which can only find fixed-length k-mer matches. MUMmer (Delcher et al., 2002; Kurtz et al., 2004) uses suffix trees to represent the sequences and bases on the suffix trees to find the maximum unique matches (MUMs) of the sequences. The MUMs are then clustered based on their sizes, gaps and distances. The gaps between clusters are closed using a modified Smith-Waterman algorithm. Similarly, AVID (Bray et al., 2003) locates exact matches as seeds using a suffix tree. Short matches are considered not biologically significant while longer matches are used as non-overlapping, non-crossing anchors. AVID recursively aligns regions between anchors by dynamic programming to perform global alignment. The Multiple Genome Aligner, MGA (Höhl et al., 2002), extends the idea to perform multiple alignment by detecting all *maximal multiple exact matches* (multiMEM) that are longer than a threshold.

One problem in genomic alignment research is that genomes contain many low information regions such as repetitive and skewed-composition areas. Alignment tools based on string matching and dynamic programming are prone to false positives in these regions. The common technique to address this problem in the above tools (Ning et al., 2001; Bray et al., 2003; Delcher et al., 2002; Kurtz et al., 2004; Kent, 2002; Schwartz et al., 2003) is to *mask out* low information content areas. However, this may cause the algorithms to miss out some important matches in these areas. Some genes, for example, are copied abundantly back into the genome to maximise their inclusive fitness. Masking out low information areas also gives rise to the issue "how low is low?"

Dynamic programming alignment algorithms rely on a scoring scheme which includes a substitution matrix and gap scores. A substitution matrix is generally drawn from some assumptions about the sequences being analysed such as the rate of mutation (PAM (Dayhoff et al., 1978)) or the minimum percentage identity of the sequences (BLOSUM (Henikoff and Henikoff, 1992)). Since more than one codon can code for the same amino acid, and different species have different preferences for nucleotide composition, it is more difficult to anticipate the mutation rates in DNA sequences. Since performing alignment involves setting these parameters, it is not clear which parameter values give the best alignment. It is therefore desirable to have an objective function to decide the best performing parameters for an algorithm.

A number of analysis tools based on the *Minimum Message Length* inductive inference principle (Wallace and Boulton, 1968) and information theory (Shannon, 1948) have been developed for comparing biological sequences. In the MML encoding method to compare sequences (Allison and Yee, 1990), two sequences are related if compressing the two together results in a shorter code than the total code of compressing them separately. A extension of this information theoretic approach to alignment is Modelling-Alignment (M-Align) (Powell et al., 2004).This method incorporates *population models* into the alignment process and can thus estimate the information content of each symbol in context, and can change the matching, insertion and deletion scores accordingly. The method has been shown to significantly reduce false positives without introducing false negatives when applied to statistically biased data. However, the quadratic complexity of M-Align prohibits applying it to very long sequences.

This thesis takes an information theory(Shannon, 1948) approach to sequence analysis. As in (Allison et al., 1992; Powell et al., 2004), it is based on the premise that if two sequences are related, one sequence must tell something new and useful about the other: A predictive model can predict a sequence better if the other sequence is known. The information content of a sequence can be measured by lossless compression. By examining information content sequences (Dix et al., 2007) produced both with and without a background sequence, similar regions of the two sequences can be identified. The expert model presented in Chapter 3 provides attractive features for genome comparison. It performs among the best biological sequence compression algorithms in the literature, with practical time and space complexity. More importantly, it can measure the information content of each individual symbol in a sequence.



(a) The transmission of sequence X

(b) The transmission of sequence X with a reference sequence Y

Figure 4.1: Transmission with and without a reference sequence.

4.3 Sequence Alignment Using Compression

Information theory (Shannon, 1948) directly relates entropy to the transmission of a sequence under a statistical model of compression. Suppose a sequence X is to be transmitted over a reliable channel where the objective is to minimise the transmitted message length. To minimise the length of the message, the sender first compresses X using a compression model and then transmits the encoded message to the receiver, which decodes the compressed stream, using the same model, to recover the original message. The compression is performed by the best possible compression model. The amount of information contained in X, or the *information content* $\mathcal{I}(X)$ of X is the amount of information actually transmitted across the channel, that is the length of the *compressed* message.

The transmission of X is illustrated in Figure 4.1a. The sender can use a predictive model, which compresses each symbol of X by estimating the probability of the symbol based on the information from all preceding symbols. A good prediction results in a short code word for the symbol. The estimated information content of every symbol makes up the information content sequence of X, which is shown in the plot below the diagram.

Suppose a reference sequence Y *related* to X is available to both parties. The sender can further reduce the transmitted message length by transmitting only the information in X that is not contained in Y with the addition of references to the shared information contained in Y. The receiver can recover X correctly because it also knows Y. Since the sender aims to send the shortest possible, recoverable message, the amount of information transmitted in this case should be no more, and probably less, than the amount of

information without the reference sequence [1]. The amount of information transmitted in the presence of the reference sequence Y is called the *conditional information content* of X given Y, denoted as $\mathcal{I}(X|Y)$. The sender is said to perform *compression of X on the background of Y*. The reduction in compressed message length caused by the presence of the reference sequence is due to shared information between the two sequences, and hence indicates the amount of *mutual information* of the two sequences. The mutual information of X and Y is denoted as $\mathcal{I}(X;Y) = \mathcal{I}(X) - \mathcal{I}(X|Y)$.

The transmission in the context of the presence a reference sequence is illustrated in Figure 4.1b. The predictive compression model now can combine the information from all preceding symbols and the information from the reference sequence to estimate the probability of a symbol. If the information from the reference sequence is *related* to the symbol, the compression model can make a better prediction of the symbol. The conditional information content of symbols given the reference sequence, therefore will be lower than the information content of the symbol without the reference sequence. The plot in the figure shows the sequence of information content of X and the sequence of conditional information content of X given the reference sequence Y. One can notice a region in X, which has a related region in Y, having significantly lower conditional information content given Y.

A local alignment of two sequences shows the mapping of similar regions in the two sequences and hence reveals the references to shared information contained in each sequence. The local alignment thus allows a reduction in transmission of a sequence in the presence of the other sequence as the reference sequence. This observation leads to the proposition that optimal alignment of two sequences leads to the best compression of one sequence on the background of the other. An alignment algorithm is proposed based on the proposition. It uses a compression model, which makes use of a local alignment, to compress a sequence on the background of a reference sequence, and suggests the alignment that gives the best compression. The quality of an alignment can be measured by the compression.

The alignment algorithm presented here is largely based on the compression model, the expert model (XM) (Cao et al., 2007, 2010a), presented in Chapter 3. The expert model was seen to be superior to other existing compression models thus giving the best estimate of the information content of sequences. In addition, its speed makes it possible to be applied to long sequences. Importantly, the expert model allows the compression of a sequence given the background of another, and can show references to the areas where better compression is achieved. These references make up the local alignment of the two sequences.

[1] The presence of an *unrelated* reference sequence might lead to a slight increase in the message length due to the possibility of referring to the reference sequence.

4.3.1 The Expert Model Compression Revisited

Recall that the expert model is a predictive model which can be used to compress as well as to measure the information content of a sequence. The model maintains a set of *experts* to estimate the probability distribution of each symbol in the sequence. Each expert assumes a model to predict the symbol based on knowledge learned from observing all preceding symbols. Since both the sender and the receiver know these symbols, they can maintain identical sets of experts. With the availability of a reference sequence, the sender and the receiver can also recruit (identical) experts from the reference sequence. The improvement in compression shows the amount of shared information between the two sequences.

The expert model employs a *global Markov expert* and a *local Markov expert* which assume a Markov model learnt from the statistics of all previous symbols and the statistics of a local context, respectively. To compress a sequence X on the background knowledge of some sequence Y, the model uses *align experts* each of which considers the symbol x_i in X to be aligned to some symbol y_j in the reference sequence Y. An align expert uses an adaptive code (Boulton and Wallace, 1969) learned from it correct predictions and its mistakes to predict x_i. Two techniques are available for an align expert to learn its probability distribution for prediction. First, in the *counting* technique, each align expert keeps track of the number of correct and incorrect predictions, and gives the following probability to the letter at y_j:

$$Pr(x_i = y_j) = p = \frac{r+1}{w+2} \tag{4.1}$$

where w is the window size over which the expert reviews its performance and r is the number of correct predictions the expert has made; the remaining probability, $1 - p$, is distributed evenly to the other letters of the alphabet. Second, in the *substituting* technique, align experts maintain a substitution matrix and give predictions according to the matrix.

To recruit potential align experts, the expert model uses a hash table which associates every position in the reference sequence with the hash key composed of k symbols preceding the position. It proposes experts for the panel that suggest that the current symbol (x_i) is copied from a position y_j which has the same hash key as k symbols preceding x_i. In order to keep the expert panel to a manageable size, the expert model has a parameter L which limits the number of experts at one time. The choice of hash key size k and the expert limit L is a trade-off between running time and compressibility, and hence the alignment quality. Generally, a small k and a large L allow the model to search for repeats more exhaustively and thus give better compression at the cost of more time.

As was seen previously, there are two groups of nucleotides – purine (C and T) and pyrimidine (A and G). The biological properties of two nucleotides in a group are

more similar than those from different groups. Therefore, substitutions changing nucleotides within a group (transitions) are more common than those that change the group (transversions). In order to permit mismatches in seeds, XMAligner provides an option to use a hash table on the alphabet {purine, pyrimidine}.

Not only do experts adapt themselves based on the context of symbols they have seen, the model also adaptively adjusts each expert's weight to reflect its prediction accuracy in the given context. Good experts are assigned high weights. Despite being nominated by the hash table, some align experts are just random matches and thus their predictions are not significantly better than the Markov experts. The algorithm excludes the by-random align experts to reduce noise and to be more time efficient. Furthermore, a "genuine" align expert performs well only within a homologous region. Beyond this, it makes random predictions and thus is excluded as well. It is important that the algorithm evaluates the goodness of each expert to assign a weight accordingly and to exclude the expert when necessary.

The reliability of each expert is continually evaluated. A reliable expert is given a high weight for combination while an unreliable one has little influence on the final prediction or may be even ignored. An expert's weight is determined by the performance of the expert over a recent history. More specifically, the weight is proportional to the negative exponential of the length of the encoding message of symbols in the history window. A detailed description of the mathematics is presented in Section 3.4 and in (Cao et al., 2007).

4.3.2 Identifying Similar Regions

The main idea behind XMAligner is that if two sequences are related, one will tell something new and useful about the other, that would not be known otherwise. If a region R_x in the query sequence X has some relationship with some region R_y in the reference sequence Y, the shared information between R_x and R_y should be better than random. The align expert that is based on R_y should perform better on R_x than the Markov experts whose predictions are based purely on the general statistics of sequence X. A region is therefore considered conserved if there is an align expert that predicts significantly better than the Markov experts. The align expert is proposed by the hash table at some point in the query sequence during the compression process and takes part in the compression until being discarded from the expert panel. It assumes that the region it predicts is related to a region in the reference sequence, and bases its prediction on the assumption. Such a pair of regions is called a *High-scoring Segment Pair* (HSP). The score of the HSP is determined by the difference between the performance of the align expert and the Markov experts.

This subsection shows that the alignment score of an HSP (Altschul, 1991) is in fact the mutual information content of the pair. Consider an align expert that aligns nucleotide x_i in X to nucleotide y_j in Y. The alignment score is specified by the logarithm of the odds ratio of a model H which assumes the two nucleotides are homologous, and a model R assuming they are random:

$$S(x_i, y_j) = log_2 \frac{Pr(x_i, y_j|H)}{Pr(x_i, y_j|R)} \tag{4.2}$$

Since model R assumes that the occurrence of x_i in X and y_j in Y are independent, the denominator of the right hand side can be expressed as $Pr(x_i, y_j|R) = Pr(x_i)Pr(y_j)$. On the other hand, model H considers symbol x_i to be related to symbol y_j and hence $Pr(x_i, y_j|H) = Pr(x_i|y_j, H)Pr(y_j)$ by Bayes's theorem. Therefore,

$$\begin{aligned} S(x_i, y_j) &= log_2 \frac{Pr(x_i|y_j, H)Pr(y_j)}{Pr(x_i)Pr(y_j)} \\ &= log_2 Pr(x_i|y_j, H) - log_2 Pr(x_i) \end{aligned} \tag{4.3}$$

$Pr(x_i|y_j, H)$ is the probability of symbol x_i estimated by the align expert upon observing y_j while $Pr(x_i)$ is the probability of x_i estimated by the Markov experts. $S(x_i, y_j)$ thus, is the mutual information of the two symbols. The alignment score of an HSP is the sum of alignment scores of all symbols in the regions. If the HSP is from two regions starting at x_m and y_n respectively and is l symbols long, its alignment score is

$$\begin{aligned} &S(x_m, y_n, l) = \\ &\sum_{i=0}^{l-1} -log_2 Pr(x_{m+i}) - \sum_{i=0}^{l-1} -log_2 Pr(x_{m+i}|y_{n+i}, H) \end{aligned} \tag{4.4}$$

The two terms are the lengths of the compressed messages of the region $x_{m,l}$ by the Markov experts, and by the align expert, respectively. In other words, the alignment score of an HSP is the mutual information content of the two regions.

An HSP is considered a homologue if its alignment score is greater than a fraction of the information content of the region from the sequence being compressed. Specifically, XMAligner has a parameter *homology ratio threshold* r, and selects HSPs having alignment scores

$$S(x_n, y_m, l) > r \sum_{i=0}^{l-1} -log_2 Pr(x_{n+i}) \tag{4.5}$$

as the local alignment.

Once all the HSPs have been selected, overlapping HSPs and HSPs having distances less than a certain threshold are chained together to form larger regions. More specifically, two HSPs (x_{m_1}, y_{n_1}, l_1) and (x_{m_2}, y_{n_2}, l_2) where $m_1 < m_2$ are considered close if the

distances between the end of HSP (x_{m_1}, y_{n_1}, l_1) and the beginning of HSP (x_{m_2}, y_{n_2}, l_2) in both sequences are less than a predefined gap. The alignment score of a chain is the sum of the alignment scores of all HSPs involved. The alignment algorithm is formally described in Algorithm 2.

Algorithm 2 XMAligner Algorithm

XMAligner(Sequence X, Y)
param L: limit on size of the expert panel E
param k: size of the hash key
param r: the ratio threshold to determine statistically significant HSPs.
param h: size of the window to evaluate experts
param T: threshold to discard align experts
Use the hash table to index every position of the reference sequence
$E \leftarrow$ empty set
for $n \leftarrow 1$ to $|X|$ **do**
  while $|E| < L$ **do**
    if expert θ_e which matches y_m to x_n is proposed **then**
      add θ_e into E
      set $Start_X(\theta_e) \leftarrow n$ {The starting point of expert θ_e in query sequence X}
      set $Start_Y(\theta_e) \leftarrow m$ {The starting point of expert θ_e in reference sequence Y}
    else
      break {No expert is proposed}
    end if
  end while
  set $Pr(x_n) \leftarrow \sum_{\theta_e \in E} w_{\theta_e} Pr(x_i|\theta_e)$ where $w_{\theta_e} = 2^{-msgLen(x_{n-h+1..n}|\theta_e)}$
  $msgLen(x_n) \leftarrow -log_2 Pr(x_n)$
  for all $\theta_e \in E$ **do**
    $msgLen(x_n|\theta_e) = -log_2 Pr(x_n|\theta_e))$
    update θ_e
    if $msgLen(x_{n-h}..x_n|\theta_e) > msgLen(x_{n-h}..x_n|\theta_{Markov}) - T$ **then**
      remove θ_e from E
      set $l \leftarrow n - Start_X(\theta_e)$
      form an HSP that matches $x_{Start_X(\theta_e),l}$ with $y_{Start_Y(\theta_e),l}$.
      set score $S(H) \leftarrow \sum_{i=0}^{l-1} -log_2 Pr(x_{n-i}|\theta_{Markov}) - \sum_{i=0}^{l-1} -log_2 Pr(x_{n-i}|\theta_e)$
      if $S(H) > r \sum_{i=0}^{l-1} -log_2 Pr(x_{n-i}|\theta_{Markov})$ **then**
        Add the HSP to a list
      end if
    end if
  end for
end for
chain sufficiently close HSPs together

4.4 Experimental Results

This section describes experiments comparing the performance of XMAligner to several common genomic alignment algorithms. The criteria for selecting these algorithms are that (i) they can align long sequences, and (ii) they are available for installation on a workstation. The alignment algorithms selected for comparison are DIALIGN (Morgenstern, 1999), CHAOS (Brudno et al., 2003), BLAT (Kent, 2002), SIM4 (Florea et al., 1998), and Nucmer and Promer in the MUMmer package (Kurtz et al., 2004). Experiments were run on a workstation equipped with an Intel dual core 2.33 Ghz CPU with 8 GB of memory. The machine ran Linux Ubuntu 8.04.

The performance of each algorithm is evaluated based on its ability to find homologues. The total of true positives (TP) is considered to be the number of homologous nucleotides that are correctly predicted as homologous, true negatives (TN) to be the number of non-homologous nucleotides that are correctly predicted as non-homologous, false positives (FP) as the number of non-homologous nucleotides that are predicted to be homologous, and false negatives (NF) as the number of homologous nucleotides that are incorrectly predicted to be non-homologous. The performance of the alignment is measured by sensitivity (Sn) and specificity (Sp) as defined by (Burset and Guigó, 1996):

$$Sn = \frac{TP}{TP + FN} \tag{4.6}$$

$$Sp = \frac{TP}{TP + FP} \tag{4.7}$$

It is worth noting that the specificity defined in Equation 4.7 is not the traditional specificity which is defined as

$$Sp = \frac{TN}{TN + FP} \tag{4.8}$$

As argued by (Burset and Guigó, 1996), because the frequency of non-homologous nucleotides tends to be much higher than that of homologous nucleotides, TN is much larger than FP and thus the traditional specificity in Equation 4.8 produces very large non-informative values. Therefore, the specificity in Equation 4.7, which is usually referred to as *precision* in statistics, is more suitable in evaluation of homologue finding programs. Where possible, the receiver operator characteristics (ROC) curve plotting specificity against sensitivity of each algorithm is presented.

4.4.1 Simulated Data

One of the main purposes of performing local alignment is to find homologues. An evaluation of an alignment tool compares the homologues suggested by the tool against "true"

homologues. True homologues in genomes, however, are not always reliable as they are often located by automated tools or by subjective prediction of human experts. The first experiment here used simulated data so that true homologues are known for comparison. Another benefit of using simulated data is that, data were generated by a controlled procedure which can be varied to explore the problem space.

The simulated data set used in this experiment contains 32 pairs of sequences. Each sequence is 100 kilobases long. Artificial homologous regions make up about 15% of each sequence. The average length of a homologue is one kilobase. The generation of a pair is as follows. First, two sequences were generated independently of each other. Homologous regions were then generated and were inserted into the first sequence. They were copied to random positions in the second sequence with errors specified by a substitution matrix. The generation allowed the homologous regions to be shuffled throughout the sequences. Alignment tools were evaluated based on their ability to detect areas of homology in the two sequences.

The artificial genetic distance between a sequence pair is specified by a substitution matrix which determines the mutation rates when copying a homologue from one sequence to the other. Distances are in number of *epochs* where the substitution matrix for one epoch is:

$$S = \begin{vmatrix} 0.990 & 0.001 & 0.007 & 0.002 \\ 0.002 & 0.990 & 0.001 & 0.007 \\ 0.007 & 0.002 & 0.990 & 0.001 \\ 0.001 & 0.007 & 0.002 & 0.990 \end{vmatrix}$$

Sequences separated by n epochs are thus separated by the matrix $Q_n = S^n$. The data set in this experiment contains sequences of distances of 5, 25, 45 and 65 epochs. The corresponding substitution matrices Q_{05}, Q_{25}, Q_{45} and Q_{65} are:

$$Q_{05} = \begin{vmatrix} 0.975 & 0.005 & 0.018 & 0.007 \\ 0.007 & 0.971 & 0.005 & 0.018 \\ 0.018 & 0.007 & 0.971 & 0.005 \\ 0.005 & 0.018 & 0.007 & 0.971 \end{vmatrix}$$

$$Q_{25} = \begin{vmatrix} 0.881 & 0.020 & 0.070 & 0.029 \\ 0.029 & 0.881 & 0.020 & 0.070 \\ 0.070 & 0.029 & 0.881 & 0.020 \\ 0.020 & 0.070 & 0.029 & 0.881 \end{vmatrix}$$

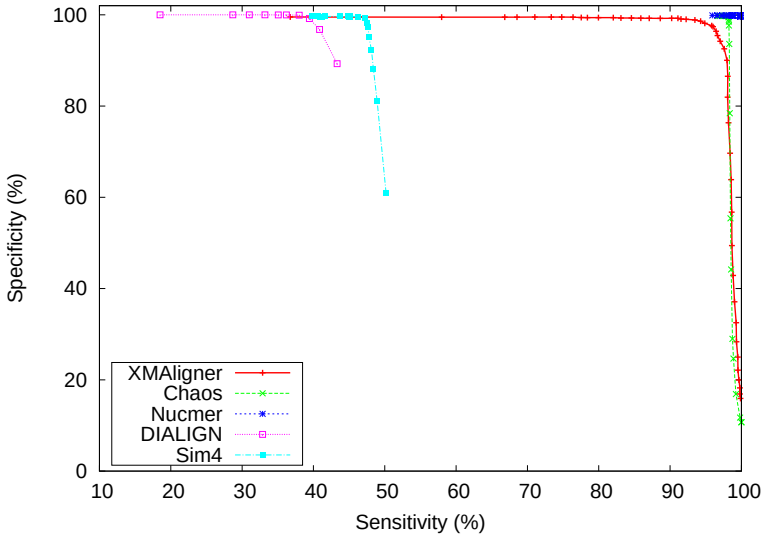
$$Q_{45} = \begin{vmatrix} 0.803 & 0.036 & 0.112 & 0.049 \\ 0.049 & 0.803 & 0.036 & 0.112 \\ 0.112 & 0.049 & 0.803 & 0.036 \\ 0.036 & 0.112 & 0.049 & 0.803 \end{vmatrix}$$

$$Q_{65} = \begin{vmatrix} 0.736 & 0.050 & 0.148 & 0.066 \\ 0.066 & 0.736 & 0.050 & 0.148 \\ 0.148 & 0.066 & 0.736 & 0.050 \\ 0.050 & 0.148 & 0.066 & 0.736 \end{vmatrix}$$

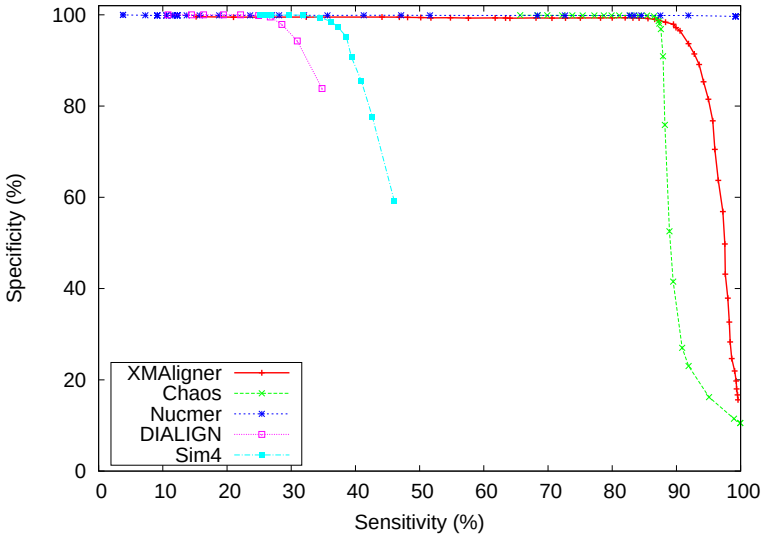
The distributions of the sequences were varied to test the robustness of the alignment algorithms. Two different distributions, a skewed distribution which generates A, C, G and T with probabilities 40%, 10%, 10% and 40% respectively, and a uniform distribution, were used to generate the data set. For a particular distance, eight pairs of sequences from the combination of these distributions for non-homologous regions in the first sequence, homologous regions and non-homologous regions in the second sequence, were generated. In total, there were 32 pairs of sequences with various distances and composition combinations, all the true positives being known. For each pair, the second sequence was aligned against the first sequence, and the regions suggested by the alignment algorithm are compared with the generated homologous regions.

The programs DIALIGN (Morgenstern, 1999), CHAOS (Brudno et al., 2003), BLAT (Kent, 2002), SIM4 (Florea et al., 1998) and Nucmer (Kurtz et al., 2004) were applied to each pair of sequences. Promer was not included because it performs alignment at the amino acid level while the data generated exhibit substitutions at the nucleotide level. Besides, the artificial homologous regions are not actual coding regions and hence cannot sensibly be translated to protein. For each algorithm used, a best guess was made at its parameters so as to get different values of sensitivity and specificity. In particular, the HSP score threshold (C) in Sim4, the score cut-off (co) in Chaos, the threshold (thr) in DIAlign, the min cluster (c) in Nucmer, and the homology ratio threshold (r) in XMAligner were varied.

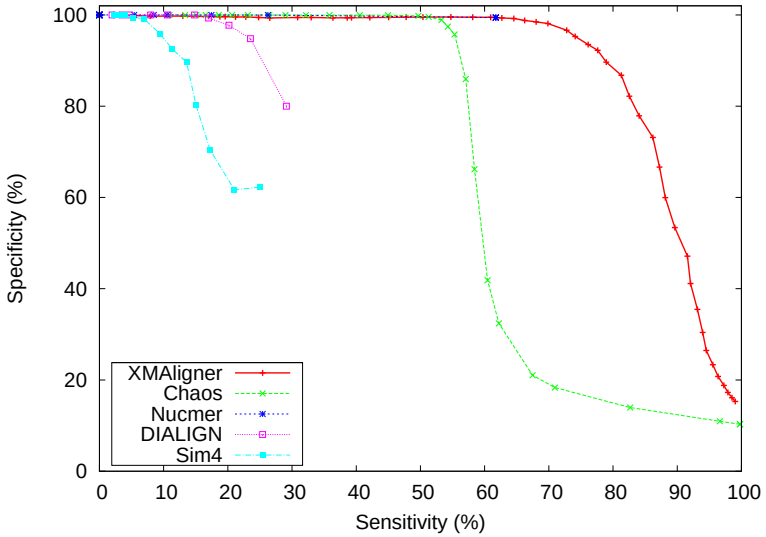
Figure 4.2 shows the performance of these alignment algorithms on the data set, grouped by distance. Generally the performance of DIAlign and Sim4 was inferior to Nucmer, CHAOS and XMAligner. On the closely related sequences (i.e., sequences at distances of small numbers of epochs), Nucmer and CHAOS performed well. They outperformed the others on the sequence set at the distance of 5 epochs. On the set of sequences separated by 25 epochs, XMAligner performance was only behind that of Nucmer. On the set in the distance of 45 epochs, XMAligner joined Nucmer as the best performers. For the more distant sequences in the 65 epoch distance set, the XMAligner clearly outperformed the other algorithms.



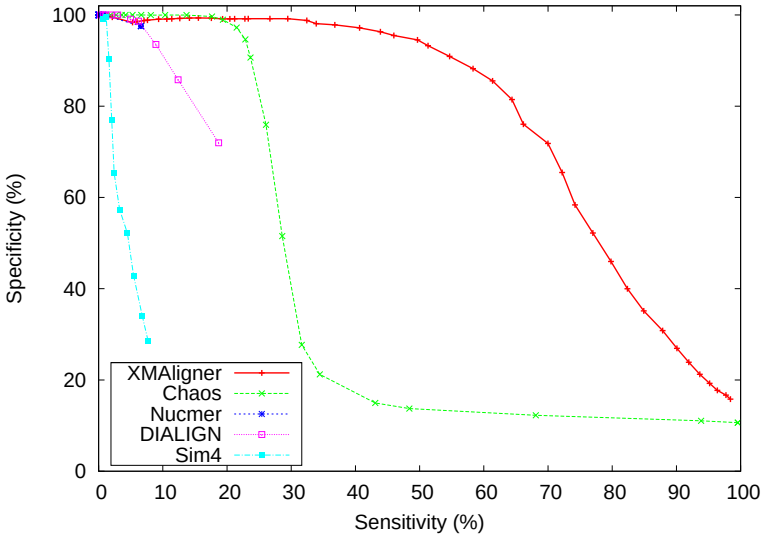
(a) Distance = 5 epochs



(b) Distance = 25 epochs

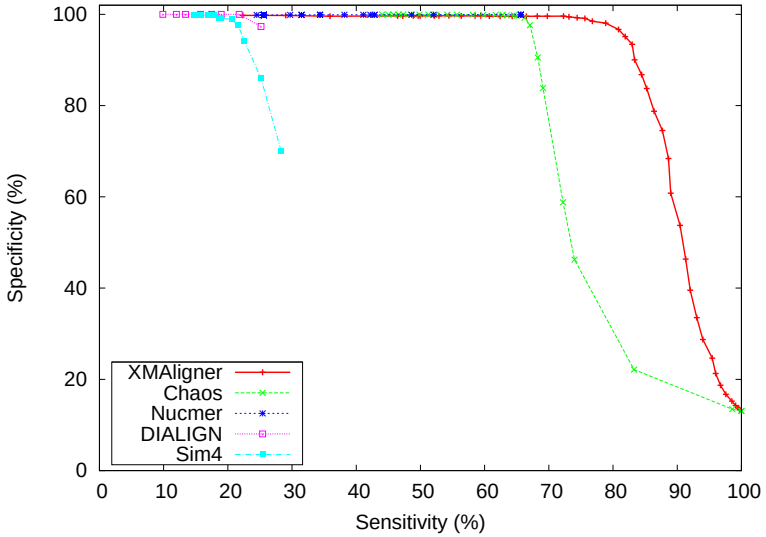


(c) Distance = 45 epochs

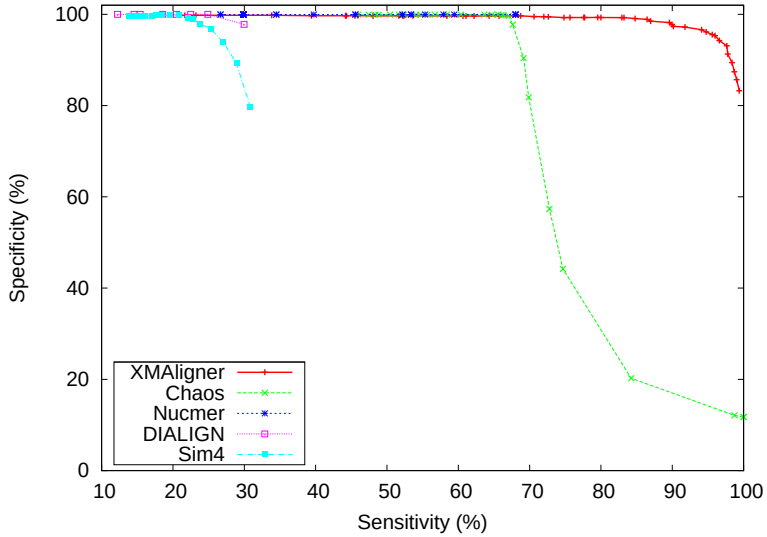


(d) Distance = 65 epochs

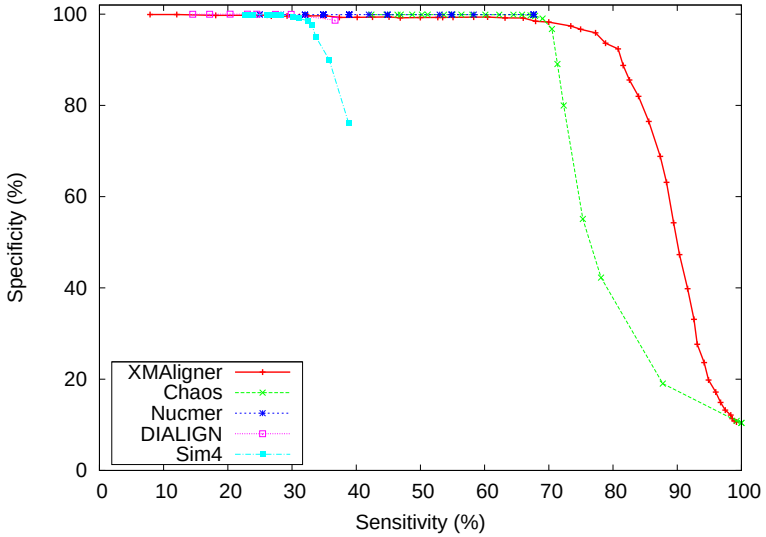
Figure 4.2: Comparative Performance on Different Distances.



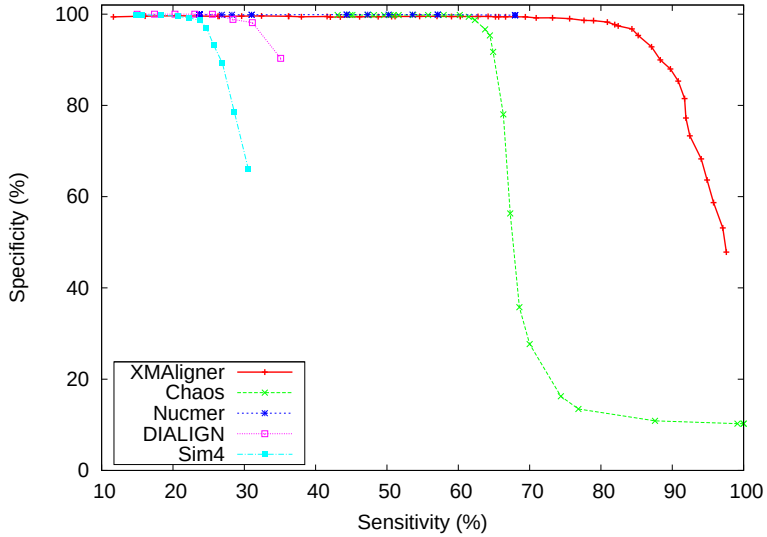
(a) UUU



(b) UUS

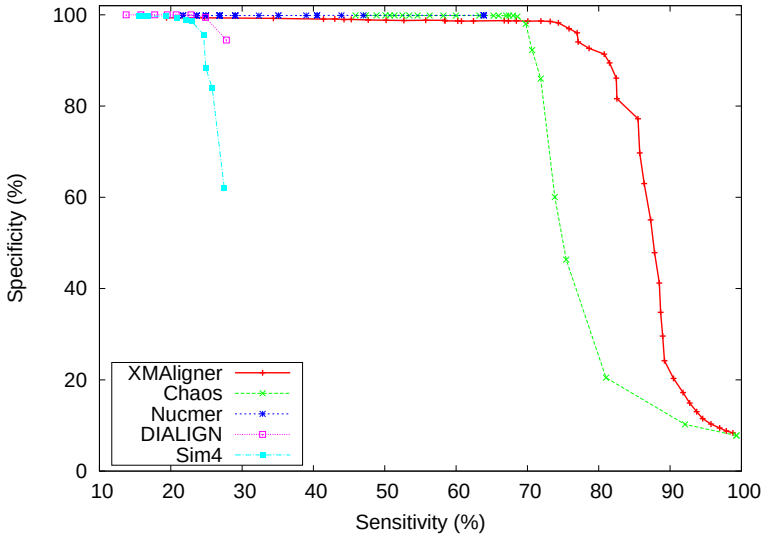


(c) USU

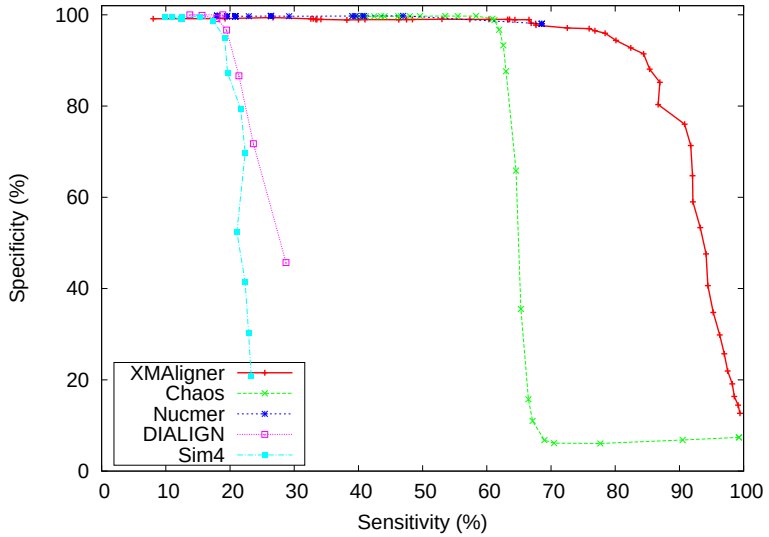


(d) USS

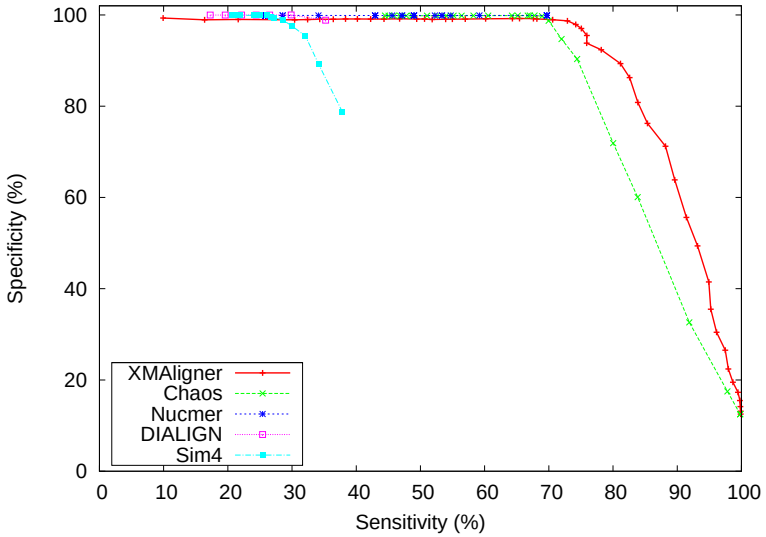
Figure 4.3: Comparison of Performance on Different Compositions.



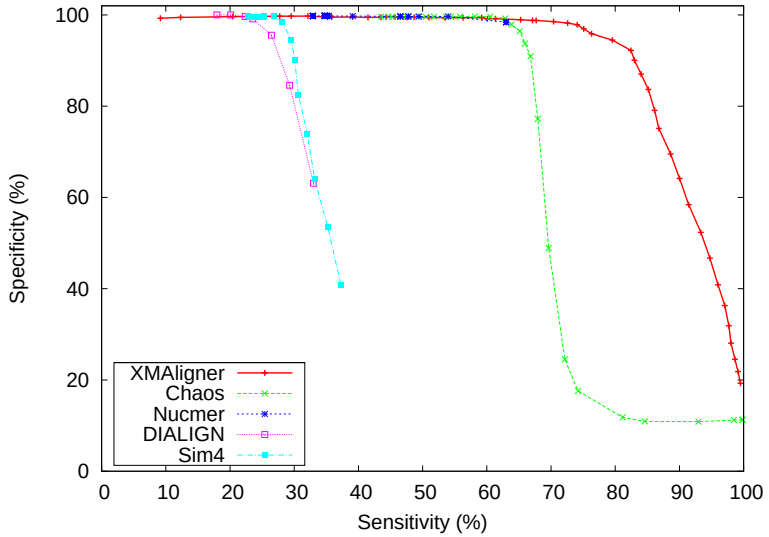
(e) SUU



(f) SUS



(g) SSU



(h) SSS

Figure 4.3: Comparison of Performance on Different Compositions (cont.).

Figure 4.3 shows the performance of these alignment algorithms on the data grouped by the composition distributions. Each group is denoted by three letters, each drawn from {U, S}, standing for the uniform distribution (U) and the skewed distribution (S). The first and last letters represent the distribution composition of the non-homologous regions in the first and the second sequences respectively. The second letter represents the distribution composition of the homologous regions. For example, the set denoted SUS contains sequence pairs that the homologous regions are uniformly distributed and the non-homologous regions in both sequences are generated by the skewed distribution.

SIM4 and DIALIGN performed poorly on all of these data groups. Nucmer was the most specific but in most cases was unable to get a sensitivity level of over 70%. XMAligner showed superiority over CHAOS on all groups of data, and by a bigger margin on statistically biased groups. CHAOS's performance deteriorated in the data group containing pairs in which the non-homologous regions in the second sequences were generated by the skewed distribution. Spurious matches occur more often in such biased data than in uniformly distributed data. CHAOS, which performs alignment at the character level, was misled by the bias of the data. On the other hand, XMAligner examines the information content of every symbol. In a low information region, the information content of a non-homologous symbol is calculated accordingly and thus spurious matches are minimised.

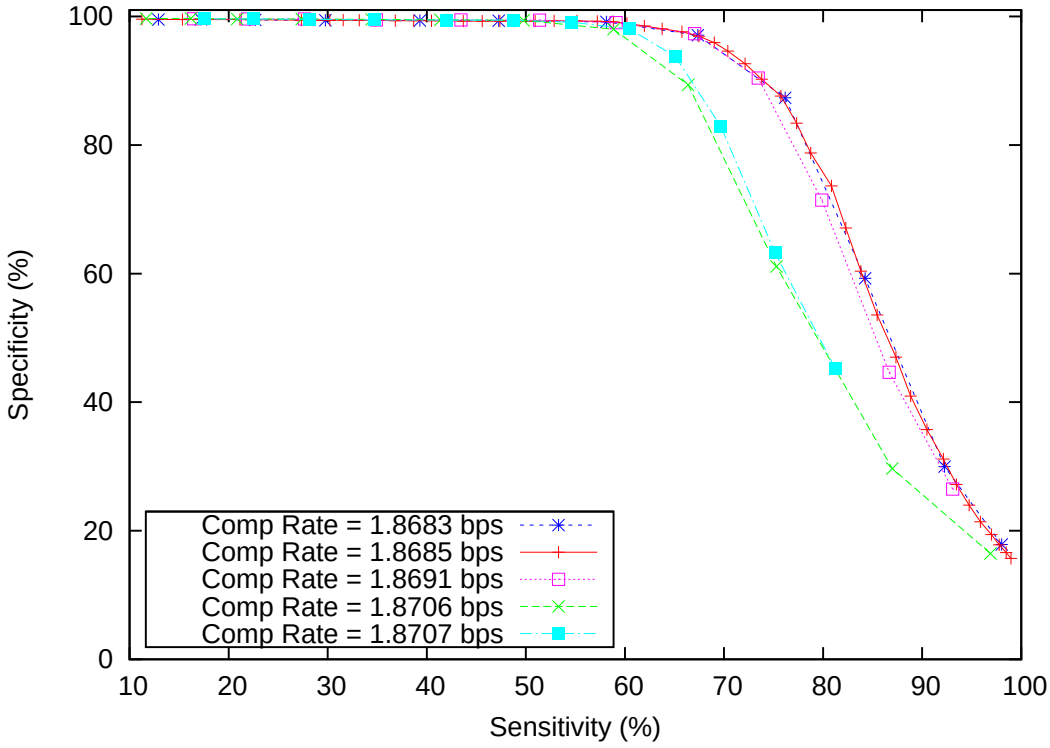


Figure 4.4: Relationship between compressibility and alignment performance.

The XMAligner algorithm is based on the premise that the best alignment of two sequences leads to the best compression of the two sequences together. As the compression of the first sequence is independent on the second sequence, the best compression of the two sequences is, here, equivalent to the best compression of the second sequence on the background of the first sequence. It is therefore proposed that the compressibility of the second sequence on the background of the first sequence be the objective function for the alignment of the two sequences.

An experiment was performed to verify the proposition that the best alignment of two sequences leads to the best compression a sequence on the background of the other. Parameters of the compression model, namely the hash key size k and the expert panel limit L, were varied so that different compression results could be obtained. The compression performance of each set of parameters is measured by the average result (in bits per symbol) of compressing the second sequence on the background of the first sequence in each pair. For each set of parameters, the homology ratio threshold was varied to obtain different sensitivity and specificity values. The ROC curve for each set of model parameters is displayed in Figure 4.4, and is labelled by the average compression result. As shown in the figure, the two configurations that produced the best compression results, 1.8683 bps and 1.8685 bps, also gave the best alignment performance. On the other hand, the configurations that produced the worst compression results (1.8706 bps and 1.8707 bps) were inferior to other configurations used in the experiment.

4.4.2 Human-Mouse Data Set

Experiments were also performed to compare alignment algorithms on real data. In an experiment, the Jareborg data set (Jareborg et al., 1999) which contains 42 annotated pairs of genomic sequences from the mouse and human genomes, was used. The sequences in the data set vary in length between 6 kilobases and 220 kilobases, with an average length of 38 kilobases. They contain 77 verified exon pairs. As exons are under stronger selective pressure, they tend to be more conserved than non-coding regions. The performance of an alignment algorithm is often evaluated based on its ability to detect exons. Indeed, the data set was used to evaluate alignment algorithms in several previous studies (Morgenstern et al., 2002; Brudno et al., 2003).

For a pair of the data set, each of the algorithms selected was applied to align the mouse sequence against the human sequence. The HSPs detected in the mouse sequence were compared with the annotated mouse exons. Each algorithm was run with its default parameters except for the best guessed threshold for varying sensitivity levels. Since exons make up about 4% of the mouse genomic sequences, a random alignment method would result in 4% specificity for any levels of sensitivity.

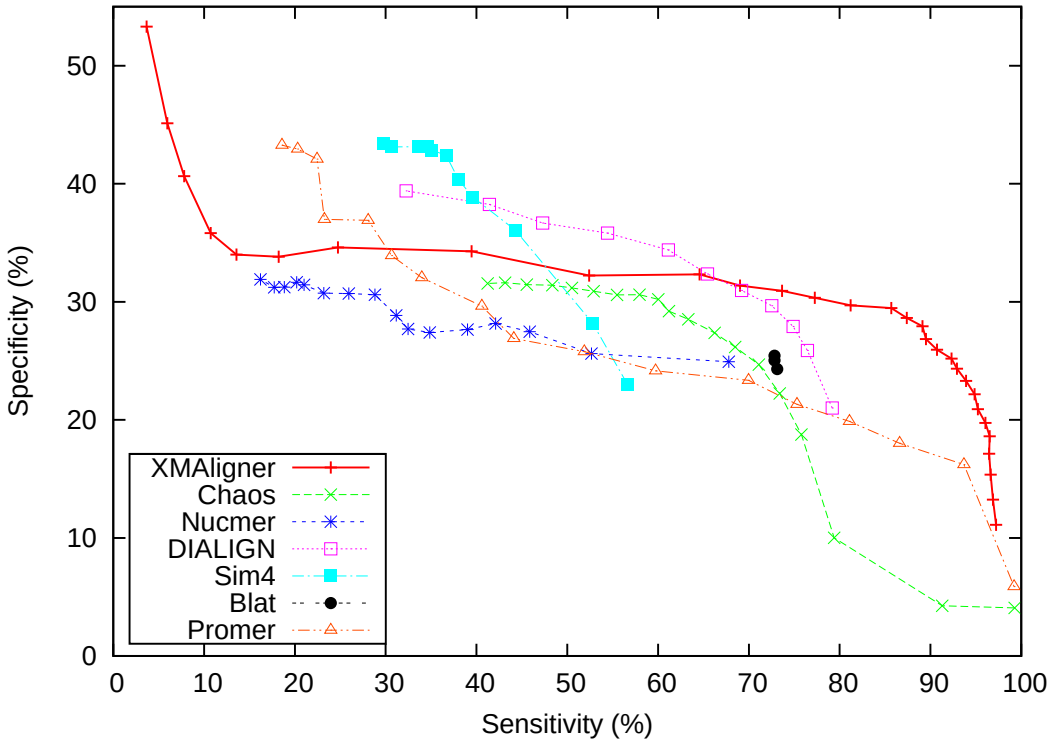


Figure 4.5: Performance Comparison on Human-Mouse data set.

The sensitivity versus specificity ROC curves for these algorithms are plotted in Figure 4.5. In general, the XMAligner was the most sensitive among the algorithms in the experiment. In particular, it outperformed CHAOS, Nucmer and BLAT which also align sequences at the DNA level. Other methods, which either translate potential exons to protein and perform alignment at the protein level (Promer and DIALIGN) or have some built-in exon boundary detection mechanism, are more specific.

4.4.3 Malaria Data Set

XMAligner was applied to align the genomes of five *Plasmodium* species, namely *P. falciparum*, *P. knowlesi*, *P. vivax*, *P. gallinaceum* and *P. yoelii*. The genome sequences were obtained from PlasmoDB release 6.2 (PlasmoDB, 2009b). Of the five species, *P. falciparum* and *P. vivax* are malaria parasites on human while *P. knowlesi* and *P. yoelii* cause malaria in monkey and rodent respectively. *P. gallinaceum* is a bird malaria parasite. The nucleotide compositions in these genomes are very different. The AT content in the genome of *P. falciparum* is as high as 80% genome-wide, and even over 90% in introns and intergenic regions while the AT content in the *P. vivax* genome and in *P. vivax* coding regions is 57.60% and 53.70% respectively. The characteristics of these genomes are presented in

Table 4.1. The five genomes have been annotated for genes but only the genomes of *P. falciparum* and *P. knowlesi* were assembled at the time of the experiment.

Table 4.1: *Plasmodium* genomes characteristics.

Species	Host	Genome Size	%(AT) in Genome	%(AT) in CDS
P. falciparum	Human	23.3 Mb	80.63%	76.22%
P. vivax	Human	27.0 Mb	57.60%	53.70%
P. knowlesi	Monkey	23.5 Mb	59.14%	69.77%
P. yoelii	Rodent	20.2 Mb	77.35%	75.22%
P. gallinaceum	Bird	16.9 Mb	79.30%	- %

The genomes of *Plasmodium* species exhibit an extremely difficult example of sequence alignment. The highly skewed distributions of genomes of species such as *P. falciparum*, especially in non-coding regions, may lead to the return of spurious matches. Furthermore, in different stages of their life-cycle, *Plasmodium* species interact with the mosquito vector and the vertebrate host and thus their genes are under strong evolutionary pressure which has led to the diversity of *Plasmodium* species. The strong evolutionary pressure from these interactions and co-evolution with hosts has resulted in different codon preferences among the genomes of *Plasmodium* species. Indeed, the AT content of coding regions of *P. falciparum* is as high as 76% while the AT content of coding regions of another human malaria parasite, *P. vivax* is only 53%, although the two species have similar metabolic pathways and their proteins share a high level of identity (Das et al., 2009).

Each of the *P. falciparum* and *P. knowlesi* genomes was aligned against each of the other four genomes and against the concatenation of these four genomes. The similar regions detected during alignment were compared with the exon annotation from PlasmoDB release 6.2 to determine the performance of each alignment algorithm. The performance of XMAligner was compared with that of Promer and Nucmer from MUMmer (Kurtz et al., 2004), which are the only two other alignment algorithms able to align such long sequences. Nucmer aligns the sequences at the nucleotide level while Promer translates potential exons to protein and aligns at the protein level. Promer is generally used when the sequences are relatively divergent, which Nucmer cannot handle.

A hash table with key size 20 was used to propose experts in XMAligner. Nucmer and Promer were run with their default parameters. The alignment of one genome against another by XMAligner took about 40 minutes. To get high sensitivity, XMAligner alignment was run in both forward and reverse directions, and the alignment results were combined. The total time for alignment of a pair of genomes was therefore about 80 minutes. The running time of Promer was shorter, about 4 to 5 minutes for alignment of one genome against another, and 20 minutes to align one genome against the four other

Table 4.2: Sensitivity and specificity of exon detection from the *P. falciparum* genome.

Method & params	*P.f*/*P.g*		*P.f*/*P.k*		*P.f*/*P.v*		*P.f*/*P.y*		*P.k*/All		Total time (Mins)
	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	
XMAligner											
r=0.05	89.16	70.07	71.35	78.40	70.46	81.43	88.76	75.21	90.76	73.57	470.55
r=0.15	76.44	75.94	57.61	83.53	55.94	86.21	75.81	81.49	80.12	79.71	451.62
r=0.25	51.83	86.50	42.03	90.81	39.60	91.65	52.22	89.45	59.73	88.63	441.40
r=0.35	35.37	93.55	31.13	94.54	28.76	94.58	36.11	93.92	44.08	93.31	439.53
Promer											
c=10	78.43	50.48	66.88	51.13	62.66	51.61	80.37	51.80	87.55	52.85	327.21
c=20	46.23	78.72	43.15	89.13	39.76	92.35	48.98	83.14	54.16	79.72	33.39
c=40	34.38	86.36	29.83	95.92	27.13	97.32	32.92	90.01	31.14	87.89	28.23
c=65	26.79	87.94	21.61	97.01	19.82	98.13	23.33	91.55	17.45	90.09	26.82
Nucmer											
c=40	12.51	65.65	2.08	26.23	1.03	17.19	10.98	64.99	15.99	66.34	17.76
c=65	6.69	90.49	0.49	51.20	0.31	38.26	5.67	91.98	8.44	91.04	7.67
c=90	3.75	94.91	0.23	42.54	0.20	40.12	3.11	92.95	4.79	93.22	6.66
c=120	1.91	98.06	0.16	56.73	0.16	73.87	1.71	97.27	2.51	94.63	6.18

Table 4.3: Sensitivity and specificity of exon detection from the *P. knowlesi* genome.

Method & params	*P.k*/*P.f*		*P.k*/*P.g*		*P.k*/*P.v*		*P.k*/*P.y*		*P.k*/All		Total time (Mins)
	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	Sen. (%)	Spe. (%)	
XMAligner											
r=0.05	98.84	49.27	98.57	49.56	99.66	49.52	98.65	49.67	99.8	48.96	478.13
r=0.15	91.73	51.62	89.04	52.50	98.23	51.48	90.74	52.68	98.77	50.57	470.40
r=0.25	61.82	63.02	52.38	64.83	93.49	57.09	59.61	66.42	93.30	57.06	450.56
r=0.35	42.12	82.86	34.05	84.74	90.01	62.06	40.78	85.89	88.64	63.87	446.37
Promer											
c=10	60.32	60.80	47.52	58.10	94.49	54.89	57.67	63.35	94.89	54.28	109.48
c=20	45.55	90.53	37.44	91.90	92.11	67.16	43.62	91.82	92.07	67.64	41.37
c=40	32.40	95.28	28.60	96.46	85.67	79.62	30.40	95.50	84.60	80.12	33.99
c=65	23.75	96.82	21.90	97.49	77.10	85.41	22.24	97.05	74.88	85.78	30.29
Nucmer											
c=40	1.89	41.00	1.77	45.33	54.74	65.48	1.96	49.45	54.81	65.09	14.05
c=65	0.33	23.45	0.28	28.40	41.46	69.63	0.38	28.94	41.40	69.66	9.96
c=90	0.08	11.68	0.09	27.48	31.74	72.08	0.10	19.21	31.66	71.90	8.26
c=120	0.01	3.72	0.03	25.94	23.74	75.11	0.03	32.68	23.67	74.91	7.43

genomes. Nucmer was even faster, it needed only one minute for pairwise alignment and four minutes for aligning one against four genomes.

The sensitivity and specificity of exon detection of the three programs on the genomes of *P. falciparum* and *P. knowlesi* are shown in Tables 4.2 and 4.3 respectively. A column with the header X/Y shows the performance of aligning the genome of X against the genome of Y and a column with header X/ALL shows the performance of aligning the genome of X against the other four genomes. The performance of each program is shown by sensitivity and specificity, presented as percentages. The parameters minimum cluster (c) of Nucmer and Promer, and the homology ratio threshold (r) of XMAligner were varied to get several different values of sensitivity.

Nucmer performed poorly in most cases, with the exceptions of aligning the *P. falciparum* genome against the *P. gallinaceum* genome, and the *P. knowlesi* genome against the *P. vivax* genome. The genomes in each of these pairs show a high similarity to each other. *P. vivax* and *P. knowlesi* are closely related. They are speculated to have split from their most recent common ancestor about only 2-3 million years ago (Carter, 2003). On the other hand, the genomes of *P. falciparum* and *P. gallinaceum* have similar AT content (80% of their genomes are A and T). The similarities between the two species even led to the belief that *P. falciparum* had arisen from the lateral transfer of parasites between the bird and human hosts (Waters et al., 1991). In the alignment of other pairs, where the two species are distantly related, Nucmer could only obtain a sensitivity of no more than 2%. Promer performed significantly better than Nucmer in these alignments, even though the matching techniques of the two algorithms are similar. The only difference between Promer and Nucmer is that Promer translates possible exons into proteins and performs alignment at protein level. Since protein tends to be more conserved than DNA, Promer was superior to Nucmer on these distantly related genomes as a result. The results suggest that existing alignment algorithms using the conventional dynamic programming approach are unable to align distantly related genomic sequences.

Although XMAligner aligns sequences at the nucleotide level (i.e., it does not take exons and protein into account), it showed a much higher level of both sensitivity and specificity than Promer in the alignment of most pairs. The only exception is the closely related pair *P. knowlesi* and *P. vivax*, where XMAligner was more sensitive but less specific. With such a close relationship, many regions other than exons also tend to conserved. While Promer translates DNA to proteins for alignment, the annotation of just codons is clearly advantageous to Promer's specificity.

4.5 Visualisation of Alignment

The alignment results from XMAligner can be integrated into the InfoV toolkit (Dix et al., 2007) for visualisation. When aligning a sequence X against a sequence Y, XMAligner outputs the sequence of estimated information content of X and the sequence of estimated conditional information content of X given Y, along with a list of HSPs. The toolkit can read these information content sequences, manipulate and display them for viewing. Annotations of sequences from other sources can also be visualised by the toolkit.

In an earlier publication (Cao et al., 2009c), an alignment experiment using the XMAligner and InfoV toolkit was performed on the *Plasmodium* genomes obtained from PlasmoDB release 5.4 (PlasmoDB, 2008). XMALigner was applied to align contig ctg6843 from the *P. vivax* genome against the genomes of *P. falciparum*. The sequence of information content of the contig and the sequence of conditional information content of the contig given the *P. falciparum* genome were generated by XMAligner. Contig ctg6843 is 589976 bases long. Compressing the contig by using only the Markov experts yielded 1.91 bits per symbol, a total information content of 1126854.16 bits. Compressing the contig on the background of the *P. falciparum* genome gave 1.85 bits per symbol, a total of 1091455.60 bits. In other words, the mutual information content of contig ctg6843 and the genome of *P. falciparum* is about $1126854.16 - 1091455.60 = 35398.56$ bits. The information content sequences were loaded into InfoV for viewing. The visualisation of the alignment by InfoV is shown in Figure 4.6. The top canvas plots the information content of contig ctg6843 and the conditional information content of the contig given the *P. falciparum* genome. The sequence of mutual information content, obtained by taking the difference of the information content and the conditional information content, is plotted in the bottom canvas.

InfoV is able to display the annotations of a sequence and the HSPs (high scoring pairs) from an alignment. The two rows of red and blue boxes near the bottom of the viewer in Figure 4.6 display the HSPs from the alignment and the exon annotation of contig ctg6843 from PlasmoDB release 5.4. When a box is clicked, a pop up windows shows the relevant information of the HSP or of the annotation. Users can zoom in and out to view particular areas of interest. Figure 4.6 shows the view from position 485000 to 510000 of the contig.

During the experiment, it was noticed that a cluster of HSPs paired regions in the contig ctg6843 to some annotated coding regions in the *P. falciparum* genome. These regions showed a high level of similarity but was not annotated in PlasmoDB release 5.4. The cluster of these regions starts at position 491038 in the contig ctg6843, and is about 15000 bases long. Its counterpart in the *P. falciparum* genome starts at position 6971447. This area in the *P. falciparum* genome is a cluster of three genes *MAL7P1.203*, *MAL7P1.320* and *MAL7P1.204*. The information for the alignment of an HSP is shown in Figure 4.6.

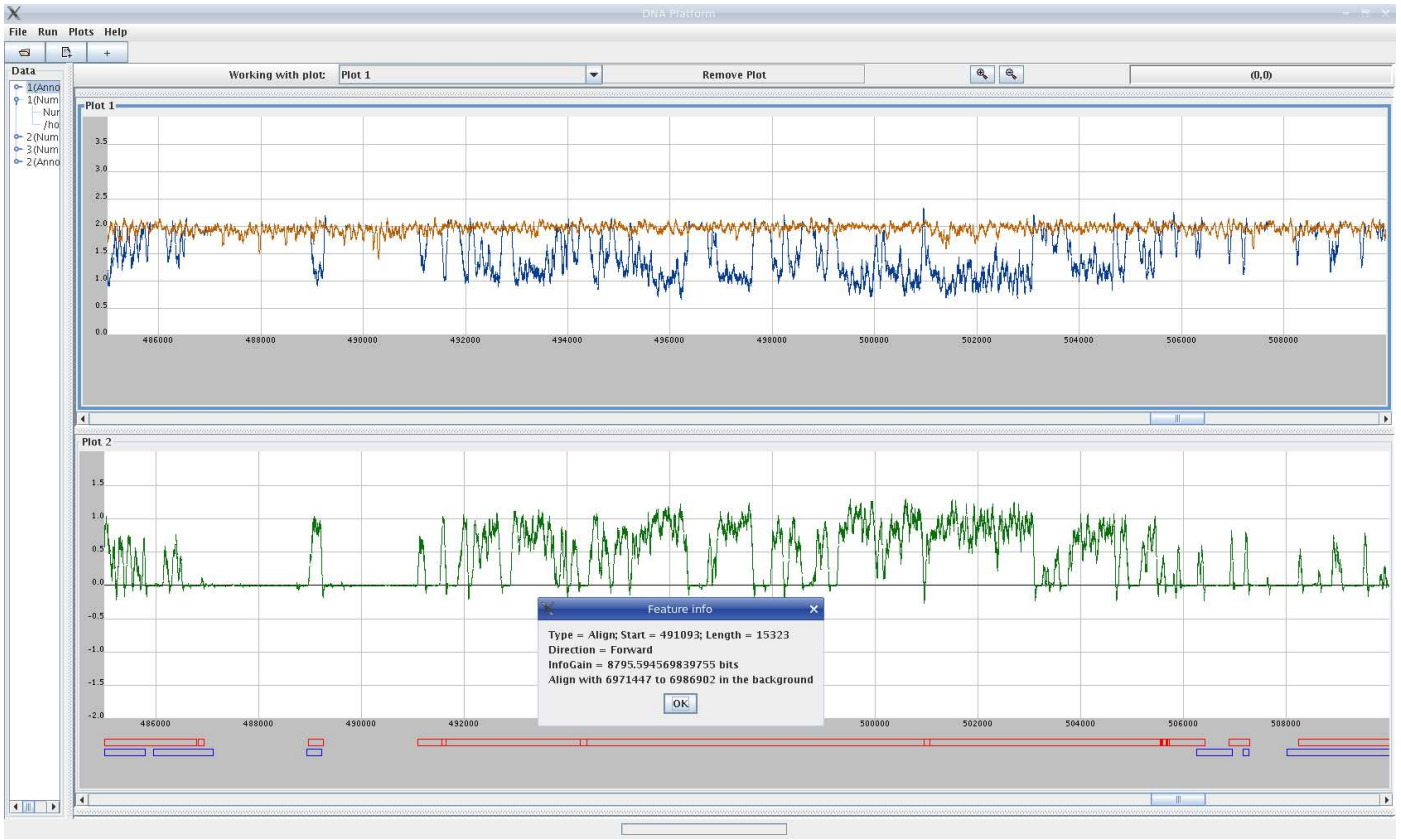


Figure 4.6: Visualisation of the alignment of the *P. vivax* contig ctg6843 against the *P. falciparum* genome.

The figure shows the view from position 485000 to 510000 of contig ctg6843 from the *P.vivax* genome. In the top canvas, the brown graph represents the estimated information content of contig ctg6843 and the blue graph represents the estimated conditional information content of the contig given the *P. falciparum* genome. In the bottom canvas, the green graph shows the estimated mutual information content of the two sequences. The blue boxes are the exons annotated obtained from PlasmoDB release 5.4 and red boxes are the HSPs detected by XMAligner. The dialogue box shows the properties of an annotation or an HSP when it is clicked. XMAligner finds a region of 15 Kb long in contig ctg6843, starting from position 491038. The region is similar to a region on the *P. falciparum* genome.

The area was thought to be a synteny region, conserved across malaria species, and containing some genes (Huestis, 2008). Later versions of PlasmoDB (PlasmoDB release 6.2 (PlasmoDB, 2009b)) verified this finding and annotated the area as gene *PVX_081792* in the *P. vivax* genome.

4.6 A Side Application: Computing Substitution Matrices

Sequence alignment generally relies on a substitution matrix which ideally reflects the probability of substitution of each symbol by another in an alignment. Most alignment algorithms based on the dynamic programming paradigm (Needleman and Wunsch, 1970; Smith and Waterman, 1981) such as the MUMmer(Kurtz et al., 2004) and Gapped BLAST (Altschul et al., 1997) attempt to find the optimal match between the sequences where matching scores are derived from a substitution matrix. It is well known that the use of an appropriate substitution matrix significantly improves the sensitivity of sequence alignment (Henikoff and Henikoff, 1992) and database search tools (Altschul et al., 1990).

Substitution matrices also provide clues to the dates of various evolutionary events and to the molecular evolution mechanism. Nucleotides mutate over time and thus genomes of distant species tend to be more divergent than closely related ones. Therefore, substitution models are often used in constructing evolutionary trees in phylogenetic analysis (Felsenstein, 1981). The extensive literature of this area is reviewed in (Lio and Goldman, 1998). Work by (Hamady et al., 2006) recently uses the nucleotide substitution rate matrix to detect horizontal gene transfers.

A substitution matrix is classically drawn from some assumptions about the sequences being analysed. The Point Accepted Mutation (PAM) (Dayhoff et al., 1978) amino acid substitution matrix is calculated by observing the differences in closely related proteins with a certain ratio of substitution residues. The PAM1 matrix estimates what rate of substitution would be expected if 1% of the amino acids had changed. On the other hand, the BLOck SUbstitution Matrix (BLOSUM) (Henikoff and Henikoff, 1992) is derived from segments in a block with a sequence identity above a certain threshold to reduce bias from closely related sequences.

While much research has been done to find substitution matrices for protein alignment, less attention has been paid to DNA substitution matrices. Since not all DNA substitutions change the encoded amino acids, information is lost when looking at amino acids instead of nucleotide bases. More than one codon can code for the same amino acid, and different strains sometimes show different preferences for a codon that encodes a given amino acid (Comeron and Aguade, 1998). Hence, it is more difficult to anticipate mutation rates in DNA sequences. For that reason, nucleotide substitution matrices are often selected empirically or by some simple assumptions. For example, the Jukes-Cantor

model (Jukes and Cantor, 1969) assumes that all the changes among four nucleotides occurs with equal probability. The Kimura model (Kimura, 1980) allows transitions and transversions to occur with different rates while the equal-input model (Felsenstein, 1981) allows the four nucleotides to have unequal frequencies at equilibrium. These models are rarely precise in practice.

Few attempts have been made to estimate DNA substitution matrices from real sequences (Goldman, 1993; Yang, 1994; Yap and Speed, 2004). These methods assume one of the substitution models mentioned above, and use the maximum likelihood approach to fit the parameters of the model. They are therefore, computationally expensive and are not relevant for analysis of long sequences such as genomes. They also depend on sequence alignment, which in turn is plausible only when a reliable substitution matrix is used.

This section presents a side application of the alignment algorithm XMAligner to compute the substitution matrix between two genomes. Instead of making a fixed assumption about the substitution models, the method finds the substitution matrix from the data being analysed. It considers the substitution matrix as a parameter to alignment and applies an expectation maximisation approach, in which the E step aligns sequences using XMAligner, and the M step estimates the parameters by maximising the expected likelihood found in the alignment process. This also derives a universal measure of the quality of substitution matrices.

4.6.1 Method

The method aims to find the substitution matrix that optimise the alignment score, and hence the compression of the two sequences together. As the substitution matrix is part of the parameter set of the alignment algorithm, the method uses an expectation maximisation approach to find the best performing parameters. It starts with a default substitution matrix. The matrix is then re-estimated from the alignment results.

Specifically, once the local alignment of the two sequences is constructed, the substitution matrix is recomputed from the set of HSPs by counting the number of mutations. Entry $P(x, y)$ ($x, y \in \{A, C, G, T\}$) of the substitution matrix is given the value:

$$P(x, y) = \frac{C_{x|y}}{C_y} \tag{4.9}$$

where $C_{x|y}$ is the number of symbol x from sequence X that are aligned with symbol y from sequence Y, and C_y is the number of symbol y in the set of HSPs.

As the sequences being analysis could be very long, the number of HSPs can be large, and many of them may be false positives. It is desirable to select those HSPs that are

likely to be drawn from "true" homologous regions. This is done by statistical hypothesis testing.

From Karlin-Altschul statistics (Karlin and Altschul, 1990), the expected frequency of occurrence of an HSP by chance (E-value) with score S or greater is:

$$E = KMNe^{-\lambda S} \tag{4.10}$$

where M and N are the lengths of the two sequences and thus MN is the size of the search space, K is the measure of the relative independence of the points in this space, and λ is a scale factor. Here $\lambda = log_e 2$ since logarithm of base 2 is used for calculating the information content. The occurrence of HSPs can be modelled as a Poisson process with characteristic parameter E. The Poisson event is that an HSP having score S or higher occurs. If the null hypothesis is that the HSP is unrelated, the probability of observing one HSP having score S or greater (p-value) is:

$$P = 1 - e^{-E} \tag{4.11}$$

Generally, the null hypothesis is rejected if the p-value is smaller than or equal to a significance level α. As E approaches 0, E and P are practically equal. Therefore, at the significance level $\alpha = 0.05$, the E-value can be used instead of p-value. In other words, an HSP is accepted if it has an E-value less than α:

$$E = KMNe^{-\lambda S} \leqslant \alpha \tag{4.12}$$

Solving Equation 4.12 gives

$$S \geqslant -log\frac{\alpha}{KMN} \tag{4.13}$$

Once the substitution matrix is computed from the accepted HSPs, it is used for alignment in the next iteration. The algorithm iterates until the matrix converges to an optimal one. Through expectation maximisation, convergence is guaranteed.

4.6.2 Experimental Results

Experiments were performed to test the performance of the method. In the experiments, the initial substitution matrix was set to

$$P_{initial} = \begin{vmatrix} .70 & .10 & .10 & .10 \\ .10 & .70 & .10 & .10 \\ .10 & .10 & .70 & .10 \\ .10 & .10 & .10 & .70 \end{vmatrix} \tag{4.14}$$

It is hard to verify substitution matrices derived from real data. Therefore a set of simulated data was used so that the substitution matrix computed can be compared with the matrix used to generate the data. A set of real data was also used to demonstrate the ability of the method.

Simulated data were used to validate the correct inference of substitution matrices. The benefit of using simulated data is that the data can be generated with added noise from a known substitution matrix, and thus it is easy to compare the generated matrix with the target one. Firstly, two "model genomes", each of one million bases, were generated. About 10% of the first genome was "coding regions" which were copied with errors and inserted into the second genome. The mutation rates were specified by a matrix P_{target}. The "non-coding regions" of the two genomes were independent on each other.

Table 4.4: The target and computed substitution matrices in the experiment with simulated data.

$$P_{target} = \begin{vmatrix} .600 & .050 & .300 & .050 \\ .030 & .650 & .070 & .250 \\ .300 & .040 & .600 & .060 \\ .050 & .300 & .050 & .600 \end{vmatrix} \quad P_{computed} = \begin{vmatrix} .596 & .051 & .300 & .052 \\ .029 & .652 & .009 & .250 \\ .299 & .041 & .599 & .061 \\ .052 & .299 & .050 & .598 \end{vmatrix}$$

The substitution matrix was reconstructed from the data by aligning the second genome against the first one. This is similar to compressing the second genome over the background knowledge of the first genome. The method was run for ten iterations, and after the fifth iteration the changes to the matrix in two consecutive iterations were seen to be negligible. In other words, the matrix converged after 5 iterations. The running time for each iteration was roughly 2 minutes. The target matrix P_{target} and the computed matrix $P_{computed}$ are presented in Table 4.4. Rows and columns in each matrix are in A, C, G, T order. Matrix $P_{computed}$ is clearly very similar to the target matrix used for generating data.

The method was also used to compute the substitution matrices between any two genomes in the five *Plasmodium* genomes described in Table 4.1. This analysis is challenging because conventional techniques can be overwhelmed by the high volume and misled by the imbalance in composition. As was seen in Subsection 4.4.3, two of the best performing genome alignment programs in the literature, Nucmer and Promer (Kurtz et al., 2004), failed to produce reasonable alignments of them. However alignments by XMAligner showed a high consistency with the manual annotations of these genomes (Cao et al., 2009b). This suggests that the substitution matrices derived from the alignments are significant. Generally, about 4 or 5 iterations were required for convergence in each run. The substitution matrices for these genomes are presented in Table 4.5; the matrix of substitution of nucleotides in genome Y to genome X is denoted P_{Y-X}.

Table 4.5: The substitution matrices of different malaria genomes.

$$P_{Pf-Pk} = \begin{vmatrix} .701 & .074 & .144 & .081 \\ .107 & .707 & .054 & .131 \\ .184 & .066 & .642 & .108 \\ .089 & .156 & .075 & .680 \end{vmatrix} \quad P_{Pk-Pf} = \begin{vmatrix} .779 & .040 & .081 & .100 \\ .137 & .372 & .057 & .436 \\ .419 & .060 & .381 & .140 \\ .103 & .083 & .040 & .774 \end{vmatrix}$$

$$P_{Pf-Pv} = \begin{vmatrix} .613 & .086 & .227 & .074 \\ .085 & .705 & .077 & .133 \\ .146 & .084 & .687 & .083 \\ .073 & .233 & .086 & .608 \end{vmatrix} \quad P_{Pv-Pf} = \begin{vmatrix} .797 & .039 & .069 & .095 \\ .136 & .386 & .049 & .429 \\ .428 & .053 & .378 & .141 \\ .095 & .072 & .037 & .796 \end{vmatrix}$$

$$P_{Pf-Py} = \begin{vmatrix} .762 & .041 & .084 & .113 \\ .112 & .613 & .059 & .216 \\ .226 & .059 & .603 & .112 \\ .115 & .082 & .040 & .763 \end{vmatrix} \quad P_{Py-Pf} = \begin{vmatrix} .765 & .041 & .082 & .112 \\ .114 & .567 & .059 & .260 \\ .236 & .057 & .593 & .113 \\ .112 & .080 & .043 & .765 \end{vmatrix}$$

$$P_{Pk-Pv} = \begin{vmatrix} .741 & .063 & .145 & .051 \\ .060 & .754 & .076 & .110 \\ .101 & .072 & .757 & .060 \\ .052 & .142 & .065 & .741 \end{vmatrix} \quad P_{Pv-Pk} = \begin{vmatrix} .808 & .050 & .083 & .059 \\ .091 & .677 & .061 & .171 \\ .200 & .067 & .641 & .092 \\ .063 & .084 & .050 & .803 \end{vmatrix}$$

$$P_{Pk-Py} = \begin{vmatrix} .796 & .036 & .068 & .100 \\ .140 & .451 & .051 & .3578 \\ .357 & .051 & .450 & .142 \\ .101 & .068 & .036 & .795 \end{vmatrix} \quad P_{Py-Pk} = \begin{vmatrix} .687 & .075 & .146 & .092 \\ .107 & .577 & .066 & .250 \\ .121 & .048 & .726 & .105 \\ .073 & .124 & .074 & .729 \end{vmatrix}$$

$$P_{Py-Pv} = \begin{vmatrix} .630 & .086 & .212 & .072 \\ .081 & .696 & .077 & .146 \\ .134 & .069 & .715 & .082 \\ .071 & .208 & .085 & .636 \end{vmatrix} \quad P_{Pv-Py} = \begin{vmatrix} .822 & .034 & .056 & .088 \\ .146 & .444 & .046 & .364 \\ .363 & .047 & .442 & .148 \\ .088 & .057 & .033 & .822 \end{vmatrix}$$

4.7 Discussion

Equation 4.5 shows that the mutual information of an HSP is in fact the traditional alignment score of the HSP which is also measured by the logarithm of the odds ratio of the probability that two symbols are related and the probability that they are independent. However, the XMAligner adaptively estimates probabilities based on the context of the pair of symbols. For example, in a low information region, the information content of a more frequent symbol is lower and its alignment score is computed accordingly. As a result, the new methodology performs better than traditional methods on statistically biased data, as demonstrated in Subsection 4.4.

The matching scores in the traditional dynamic programming approach are also calculated based on the information theory perspective. Indeed, an entry in the common substitution matrices such as PAM (Dayhoff et al., 1978) and BLOSUM (Henikoff and

Henikoff, 1992) represents the logarithm of the ratio of the probabilities of two hypotheses: the pair are homologous and the pair are random. These probabilities are calculated based on some pre-aligned data or under some evolutionary assumptions. However, it is desirable to estimate these probabilities from the sequences at hand. This calculation better reflects the information content of each symbol of the sequences to be aligned. These scores can even be estimated if the sequences are sufficiently long (Cao et al., 2009b) as shown in Section 4.6.

With reference to the traditional dynamic programming approach, an align expert proceeds diagonally and thus can only find gap-free similar regions. However, there can be more than one align expert employed at any time. If there are gaps in the conserved regions, some neighbouring expert(s) would be proposed so the XMAligner can handle gaps implicitly without any assumptions about gap scores.

Most existing alignment algorithms lack an objective function to indicate which parameters are the most suitable for the data. Objective functions are very important for applications like sequence alignment because biological data are so diverse. It is very hard to anticipate which parameter values capture the essence of the data and will give the best results, especially for data that are not well studied. The objective function provided by XMAligner naturally guides parameter estimation and improves alignment quality.

XMAligner presents a new methodology for extending seeds and thus it can make use of any technique for locating seeds from conventional alignment algorithms. Any method for locating seeds such as gapped hash table (Keich et al., 2004), suffix trees and arrays can be applied. Indeed, the XMAligner has an option to use suffix arrays and suffix trees to locate seeds. Other techniques will be implemented in the future. The suitability of each seeding technique is measured by the compression objective function.

4.8 Summary

This chapter has presented XMAligner, a new sequence alignment approach that matches long sequences at the information level. Unlike traditional alignment algorithms which perform matching at the character level, XMAligner reports aligned regions of two sequences if there is some shared information between the two regions. The approach is shown to outperform conventional character-matching approaches, especially for distantly related sequences and sequences with statistically biased composition. The method is able to align eukaryotic genomes with modest hardware requirements. The output from the XMAligner can be integrated into a visualisation tool to aid the analysis of sequences.

This work argues that, as genomic sequences are meant to carry information, aligning at the information content level is a better approach for genomic sequence alignment.

Each symbol of the sequences should be examined within the context in which it occurs in the sequence, and the information content of the symbol should be measured accordingly. The approach therefore, is better suited than conventional approaches which measure the alignment score of matching symbols entirely based on a fixed scoring scheme.

The method is based on the sound theoretical foundations of information theory. It is shown that the method successfully regains the substitution matrix from simulated data derived from a known matrix with introduced noise. The method has also been applied to real data with differing phylogenetic distances and nucleotide composition which could mislead popular current methods. Unlike traditional methods, the method does not rely on being given a predetermined substitution matrix. It incorporates the alignment of sequences and the substitution matrix computed in an expectation maximisation process. Furthermore, it can handle very long sequences in practical running time. The method therefore, can facilitate knowledge discovery in large and statistically biased databases. Examples of future applications of the method include multiple alignment of genomic sequences and analysis of genome rearrangement.

Chapter 5

Phylogenetic Tree Construction

Nothing in biology makes sense except in the light of evolution
–Theodosius Dobzhansky

Essentially, all models are wrong, but some are useful.
–George E. P. Box

5.1 Introduction

The elucidation of the evolutionary relationship of all species on earth has been a major scientific quest since Darwin's time (Darwin, 1859). The evolutionary relationship is often represented by a hierarchical structure, called the *phylogenetic tree* of the species, which shows speciation events – the splitting of lineages. Not only does the evolutionary relationship facilitate the classification of species and the understanding of the history of life, but it is also useful for the study of many other aspects of biology. Because "nothing in biology makes sense except in the light of evolution" (Dobzhansky, 1973), inferring phylogenetic trees has been one of the central activities in biological research.

Recent sequencing technologies produce large volumes of molecular sequences that provide a more complete data source for phylogenetic analyses than the traditional morphological data. Phylogenetic analyses using molecular data have led to many interesting observations about the relationships among species such as those between fungi and animals relative to plants (Baldauf and Palmer, 1993), and between rodents and other mammal species (Cao et al., 1994b; Robinson-Rechavi et al., 2000). However, phylogenetic analyses face many challenges despite the existence of a great many methods developed in the last two decades. These methods often use simple mathematical models and make assumptions that are rarely true in reality. As remarked by Nei and Kumar (2000), none

of the phylogenetic analysis methods is perfect. Every method has its own strengths and weaknesses, and is only good for some types of data. It is therefore necessary to develop new techniques so that tools are available to deal with a wider range of data.

Existing molecular phylogenetic methods generally require a multiple alignment of the sequences. However, obtaining a good multiple alignment is a difficult problem; it often requires manual alignment by experts and is time consuming (Yang, 1994; Cao et al., 1998). The complexity of the multiple alignment problem limits practical algorithms to a "few" "short" sequences unless heuristics are used. This limits these methods to short, homologous sequences such as a gene or a ribosomal RNA.

Phylogenetic analyses based on different genes or rRNA can result in inconsistent phylogenetic trees since they may have different rates of evolution (Leclerc et al., 2004). The evolutionary history of the species may not even be the same as the evolutionary history of every one of their genes because some genes may have arisen by means other than inheritance such as horizontal transfer (Lerat et al., 2003; Gogarten and Townsend, 2005). It can also be difficult to find genes that are sufficiently conserved across the species of interest and, at the same time, are sufficiently diverged to be of use for analysis. It is suggested that, analyses based on whole genomes can give more reliable phylogenetic results as each organism is essentially defined by the information contained in its genome. Phylogenetic analysis methods that can handle such long sequences are therefore necessary.

Most existing methods of phylogenetic analysis involve examining the mutation patterns of characters in the given sequences. Since characters in sequences are means to store genetic information, and to pass on the information to the next generation, mutations of characters cause changes in the information in the sequences. It is therefore, proposed that phylogenetic trees can be inferred by examining the sequences *at the information level*.

This chapter investigates an information theoretic approach to the phylogenetic study of DNA molecules. Inspired by the expert model (XM) compression algorithm (presented in Chapter 3), it proposes a new measure of genetic distances, named *XMDistance*, based on the information content of sequences. The chapter presents two practical compression techniques to measure the information content of sequences, and the mutual information content between sequences. If the sequences have been aligned, a simple Markov model is used for compression. In cases where the sequences are not aligned or cannot possibly be aligned, the expert model is used. The proposed distance measure does not require any assumptions about the evolutionary model. XMDistance is shown to perform comparably with existing standard phylogenetic analysis methods while having a much better time complexity. Furthermore, the method can be extended for phylogenetic analyses from whole genomes and from sequences that cannot be reliably aligned.

The chapter is organised as follows: Section 5.2 presents the background to phylogenetic studies. The most common phylogenetic analysis methods are reviewed in Section 5.3. Section 5.4 gives a theoretical analysis of the proposed method and describes the computation of the distances when the sequences have been aligned. Experiments on XMDistance are presented in Section 5.5. In Section 5.6, the approach is applied to phylogenetic analyses of whole genomes where the sequences cannot be reliably aligned. Section 5.7 concludes this chapter.

5.2 Phylogenetic Analysis Background

A *phylogenetic tree*, or a *phylogeny* is a hierarchical structure showing the genealogical relationship of a group of species, a group of genes, or a group of populations. The branching pattern of a phylogenetic tree is called the *topology* of the tree. Certain trees have each of their branches labelled with a weight and thus are called *weighted* trees. Each exterior node (or leaf) of a tree represents a species or a gene and is called a *taxonomic unit* (or a *taxon*). An interior node represents the most recent ancestor of the branches derived from it. Generally, a species splits into two descendant species at the time of speciation, and thus the corresponding node in a phylogenetic tree has two daughter nodes. Such a tree is called a *bifurcating* tree. Several speciations may have happened within a short time period, and thus more than two species may be considered to have arisen at essentially the same time. A phylogenetic tree which is allowed to have nodes with more than two daughter nodes, is called a *multifurcating tree*. The material presented in this chapter is mainly for bifurcating trees.

A phylogeny can have a root (a *rooted tree*) or have no root (an *unrooted tree*). A rooted tree shows the directions of inheritance among species, such as a node is the *parent* of another node. Some phylogenetic analyses result in an unrooted tree. Several *rooting* techniques can be used to find the root of the tree, and the ancestry relationships among taxa. Under the *molecular clock* assumption, which is for a constant evolutionary rate across lineages, the root can be identified by locating the node that is nearly equi-distant from all exterior nodes. This technique is called *molecular clock rooting*. However, since the molecular clock assumption is often violated, molecular clock rooting is not practically applied. Instead, the *outgroup rooting* strategy is often used for identifying the tree root. In this method, one or more distantly related species, called *outgroups*, are included in the reconstruction of the phylogeny. The root of the tree is the node that connects the outgroups to the species being studied, the *ingroups*.

For a group of species, the number of possible topologies increases rapidly with increasing number of taxa. For trees that have specific values assigned to their leaves (*labelled trees*), there are $(2n-5)!! = 3 \times 5 \times 7 \times ... \times (2n-5)$ possible unrooted trees, and

$(2n-3)!! = 3 \times 5 \times 7 \times ... \times (2n-3)$ possible rooted trees, for n species. Table 5.1 shows the numbers of trees of each kind for various numbers of species.

Table 5.1: Numbers of possible topologies for a group of species.

Taxa	Rooted Trees	Unrooted Trees
3	3	1
4	15	3
5	105	15
6	945	105
7	10,395	945
8	135,135	10,395
9	2,027,025	135,135
10	34,459,425	2,027,025

A phylogenetic tree that presents the evolutionary history of a group of species is called a *species tree*. There are trees constructed from a gene presenting in these species. Such a tree is called a *gene tree*. A gene tree does not necessarily agree with the species tree. The evolutionary rate of a certain gene may be different across species. Some genes are even transferred to a species by means other than reproduction such as by horizontal transfer between species. Therefore, phylogenetic trees inferred from different genes may be different to each other, and different to the species tree.

Historically, reconstruction of phylogenetic trees often uses morphological data and the fossil records. However, these data are often incomplete and lack details of evolution. The evolutionary changes in morphological and physiological data are extremely complicated and difficult to model. Although phylogenetic analyses using these data are able to infer major aspects of the evolutionary history of organism, the details have always been controversial (Nei and Kumar, 2000).

Recent advances in molecular biology allow the generation of molecular genetic sequences of various species. Since every organism is essentially defined by the genetic information which is passed down from its parents, it is sensible to infer the evolutionary relationship of species from their genomic data. Furthermore, the primary cause of evolution is mutational changes in genes. Morphological and physiological features of organisms are ultimately controlled by the genetic information contained in these molecular sequences. Therefore, molecular data are expected to be able to resolve many phylogenetic questions that cannot be addressed with morphological data. Molecular data also provide useful information for understanding the mechanism of evolution of any characteristics of interest.

The goal of molecular phylogenetics is to assemble an evolutionary relationship for a set of species from some genetic data such as DNA, RNA or protein sequences. Typically,

each species is represented by a small molecular sequence such as a gene or a ribosomal RNA, which carries some information of evolutionary history in sequence variations. Closely related organisms generally have a high degree of agreement in these sequences, while the sequences of distantly related organisms show patterns of greater dissimilarity. Similar sequences are placed close to each other in the phylogenetic tree to show a probable ancestry of the corresponding species.

A phylogenetic analysis generally relies on assumptions about the evolutionary process that produced the observed data. The assumptions are formalised by some statistical model. A model used in phylogenetics generally presents a hypothesis about the relative rates of mutation at a site in the sequences being analysed. The importance of evolutionary models has been recognised in work by Sullivan and Joyce (2005) since selecting improper models may lead to biased results.

Traditionally, molecular phylogenetics uses protein sequences because natural selection mainly acts on proteins. They are also more conserved than DNA. Mutations between amino acids having similar biochemical properties typically occur more frequently than do mutations between dissimilar amino acids. This suggests that there are patterns of substitution among amino acids. Mutations in protein are, therefore, often modelled by a 20×20 matrix, called a *substitution matrix*, that reflects the frequencies of substituting one amino acid for another. An element m_{ij} of the matrix presents the probability that the amino acid at row i changes to the amino acid at column j during one time unit.

Examples of amino acid substitution matrices are the *point accepted mutation* (PAM) matrices proposed by Dayhoff et al. (1978). The frequencies of substitutions between amino acids in the PAM matrices are inferred empirically from some well established evolutionary trees of well understood proteins such as haemoglobin, cytochrome and fibrinopeptides. The time unit, one PAM, used in this matrix is the time during which, on average, one amino acid substitution occurs per 100 sites. Jones et al. (1992) and Gonnet et al. (1992) use the same idea to develop other amino acid substitution matrices from modern databases.

While mutations at amino acid level are subject to selection and thus amino acids are suitable for phylogenetics analysis, sequencing proteins is time consuming and highly error-prone. With the invention of rapid methods of DNA sequencing (Sanger et al., 1977) and the high throughput sequencing techniques (Schuster, 2008), DNA sequencing is more productive and more accurate. Although evolutionary change in DNA sequences is more complicated to model than that in protein sequences, recent phylogenetic analyses are more often based on DNA because of the availability and reliability of DNA data. The materials discussed in this chapter mainly focus on DNA sequences, but are largely applicable to protein sequences.

Table 5.2: Common models of nucleotide substitution.

(a) Jukes-Cantor model

	A	C	G	T
A	-	α	α	α
C	α	-	α	α
G	α	α	-	α
T	α	α	α	-

(b) Kimura model

	A	C	G	T
A	-	β	α	β
C	β	-	β	α
G	α	β	-	β
T	β	α	β	-

(c) Equal-input model

	A	C	G	T
A	-	$\alpha\pi_C$	$\alpha\pi_G$	$\alpha\pi_T$
C	$\alpha\pi_A$	-	$\alpha\pi_G$	$\alpha\pi_T$
G	$\alpha\pi_A$	$\alpha\pi_C$	-	$\alpha\pi_T$
T	$\alpha\pi_A$	$\alpha\pi_C$	$\alpha\pi_G$	-

(d) HKY (F84) model

	A	C	G	T
A	-	$\beta\pi_C$	$\alpha\pi_G$	$\beta\pi_T$
C	$\beta\pi_A$	-	$\beta\pi_G$	$\alpha\pi_T$
G	$\alpha\pi_A$	$\beta\pi_C$	-	$\beta\pi_T$
T	$\beta\pi_A$	$\alpha\pi_C$	$\beta\pi_G$	-

(e) GTR model

	A	C	G	T
A	-	$a\pi_C$	$b\pi_G$	$c\pi_T$
C	$a\pi_A$	-	$d\pi_G$	$e\pi_T$
G	$b\pi_A$	$d\pi_C$	-	$f\pi_T$
T	$c\pi_A$	$e\pi_C$	$f\pi_G$	

(f) Unrestricted model

	A	C	G	T
A	-	a_{12}	a_{13}	a_{14}
C	a_{21}	-	a_{23}	a_{24}
G	a_{31}	a_{32}	-	a_{34}
T	a_{41}	a_{42}	a_{43}	-

Note: An element S_{ij} in a matrix represents the substitution rate for the nucleotide in row i to the nucleotide in column j. π_A, π_C, π_G and π_T are the frequencies of A, C, G and T respectively.

Modelling evolutionary change of DNA is challenging. Substitution patterns of nucleotides are not the same in different regions such as protein-coding regions, introns and repetitive DNA. Even the patterns of substitution at the three positions in a codon are different. Nevertheless, practitioners often use simplified statistical models of DNA mutations. Despite the simplifications, these simple models perform as well as, or even better than, more sophisticated models in most cases. A review of some commonly used DNA substitution models follows.

The simplest DNA substitution model is the Jukes-Cantor model (Jukes and Cantor, 1969), which assumes that the probability of a nucleotide changing to a different nucleotide during a time unit is equal to a value α and that every site in a sequence evolves at a uniform rate. The model thus has only one parameter, α. In practice, the probability of transitions is often higher than that of transversions. To account for this, the Kimura model (Kimura, 1981), has one mutation rate for transitions, α, and another for transversions, β. Jukes-Cantor and Kimura models are presented in Tables 5.2(a) and 5.2(b) respectively.

Generation of data using the above substitution models results in a uniform distribution of nucleotide frequencies at equilibrium regardless of the initial nucleotide frequencies. In practice, however, biological sequences are not always uniformly distributed.

Felsenstein (1981) and Tajima and Nei (1984) propose the *equal-input* model which specifies that the mutation rates are proportional to the frequencies of the nucleotides so that the nucleotide frequencies at equilibrium are the same as the nucleotide frequencies of the observed data. The model is presented in Table 5.2(c). To account for the variation of mutation rates between transitions and transversions, the HKY model Hasegawa et al. (1985) and the F84 model (Kishino and Hasegawa, 1989; Felsenstein and Churchill, 1996) (Table 5.2(d)) differentiate between transitions and transversions and allow for unequal nucleotide composition. This is a hybrid of the equal-input model and the Kimura model. The two models are essentially the same; they differ only in the methods of calculation.

A generalisation of the above mutation models is the *General Time Reversible* (GTR) model (Tavare, 1986; Yang, 1994). The GTR model requires time reversibility; it treats a substitution and its reversal (e.g., an A→T transversion and a T→A transversion) as equivalent. The model thus has six parameters accounting for six possible substitutions among four nucleotides as described in Table 5.2(e). A more relaxed and thus more complex model, the unrestricted model (Table 5.2(f)), allows any rate of mutation between a given pair of nucleotides.

The substitution rate of a nucleotide may differ from site to site. In coding regions, some nucleotide substitutions, notably at the third position in a codon, do not cause a change to the encoded amino acid. These substitutions, called *synonymous* substitutions, thus occur more often than nonsynonymous substitutions. Furthermore, nonsynonymous substitutions vary in effect from gene to gene due to being subject to natural selection (Miyata and Yasunaga, 1980). To account for this, several models have been developed to allow substitution rates at various sites to differ. The Nei-Gojobori model (Nei and Gojobori, 1986) estimates the rates of synonymous and nonsynonymous substitutions by computing the numbers of synonymous and nonsynonymous substitutions, and the number of potential synonymous and nonsynonymous sites. Other methods, such as those by Li et al. (1985), Pamilo and Bianchi (1993) and Li (1993) extend the Nei-Gojobori model by considering transition and transversion bias.

Rather than considering nucleotide substitutions, a model by Goldman and Yang (1994) considers substitutions among *codons*. The model consists of a 61×61 matrix to represent the rates of substitutions between any two of the 61 sense codons. Similar to the HYK model, this model takes into account codon frequencies and transition/transversion bias. The idea of codon substitution is further improved in recent works, such as that by Yang and Nielsen (2008).

5.3 Review of Phylogenetic Analysis Methods

Typically, the first task in phylogenetics analysis is to construct a multiple alignment of the sequences involved. The outcome of the multiple alignment is a table. A row in the table presents a sequence, and a column specifies a site from each sequence. The symbols in a column are considered to be descended from a common position in an ancestral sequence. The table also contains gaps representing indels. An exhaustive multiple alignment is essentially intractable, especially when the number of sequences is relatively large. Practical multiple alignment methods, for example, ClustalW (Higgins et al., 1996), use heuristics such as the progressive sequence alignment technique (Feng and Doolittle, 1987). These methods start with a pairwise alignment of the two most similar sequences, and progressively add more distantly related sequences to the alignment. The order in which sequences are added is guided by a crude *guide tree*.

A tree building method is then applied to the alignment. Molecular phylogenetic tree reconstruction methods are broadly categorised into two main classes. Methods in the first class rely on a genetic distance measure. They firstly compute a matrix of pairwise distances between any two taxa, and then build a phylogenetic tree by considering the distances. These methods therefore are called *distance methods*. Methods in the second class, called *optimisation methods*, search for the tree that optimises some criterion such as minimising the number of evolutionary events (*maximum parsimony* (Camin and Sokal, 1965)), or maximising the likelihood (*maximum likelihood* (Felsenstein, 1981)). Methods in this class generally search in the tree space and are thus computationally expensive, especially for trees with a large number of taxa, say more than 10.

A distance method reconstructs a phylogenetic tree from a matrix of genetic distances between each pair of taxa. The genetic distance between two species reflects how long ago the two species split at their most recent ancestor. A distance measure is therefore required. Ideally, the distance measure should be a metric, i.e., it must satisfy the following conditions:

- non-negativity: $D(x, y) \geq 0$.
- identity: $D(x, y) = 0$ if and only if $x = y$.
- symmetry: $D(x, y) = D(y, x)$.
- triangle inequality: $D(x, y) + D(y, z) \geq D(x, z)$.

Some distance reconstruction methods require the distance measure to also satisfy the four-point condition (Buneman, 1971) which specifies that for any four species x, y, z and t, the two largest of the three quantities $D(x, y) + D(z, t)$, $D(x, t) + D(z, y)$ and $D(x, z) + D(y, t)$ must equal to each other. It is also desirable that the distance measure

is proportional to elapsed time. Not only does such a distance measure satisfy the above conditions, but it also facilitates the estimation of time which is one of the main goals of phylogenetic analysis.

Phylogenetics studies often assume a *molecular clock* which asserts that genetic sequences evolve at a constant rate. Under this assumption, the number of substitutions is proportional to the divergence time, on average. Traditional measures of genetic distances, therefore, often use the number of observed changed sites between the two sequences. A simple measure is the *p distance*, estimated by the proportion of differing sites that can be observed:

$$p = \frac{n_d}{n} \tag{5.1}$$

where n_d is the number of differing sites and n is the number of sites in the sequences. The p distance is approximately linear to the divergence time only for short distances, but it underestimates the number of actual substitutions between two distantly related sequences. The main reason for the non-linearity of the p distance with time is that multiple substitutions can occur at the same site in two lineages, and that the amino acid or the nucleotide at a site can substitute back to the original one. These parallel and backward substitutions cause the number of observed differing sites to be less than the number of true substitutions.

More sophisticated distance measures can be applied for a better estimate of the number of substitutions and the elapsed time from the number of differing sites. The expected number of differing sites of two sequences over time is computed from an underlying substitution model. The reverse function giving the time for the number of differing sites is then used to estimate the divergence time between two sequences. For example, if the number of substitutions is assumed to follow a Poisson distribution, the probability that neither of the homologous sites from two sequences, that diverged t time units ago, had undergone substitution is

$$q = e^{-2rt} \tag{5.2}$$

where r is the evolutionary rate, in number of substitutions per time unit. The number of substitutions $d = 2rt$ can then be estimated by

$$d = -ln\ q = -ln(1 - p) \tag{5.3}$$

This distance measure, called the *Poisson correction (PC) distance*, gives better estimation of the number of substitutions than the p distance does. Though it does not consider parallel and backward substitutions, PC distance reduces the effects of these substitutions, especially for small values of p (Nei and Kumar, 2000).

To take into account parallel and backward substitutions, it is necessary to use a substitution model. For example, the number of substitutions is estimated by the Jukes-Cantor model (Jukes and Cantor, 1969) as

$$d = -\frac{3}{4} ln \frac{3 - 4p}{3} \tag{5.4}$$

and by the Kimura model (Kimura, 1981) as

$$d = -\frac{1}{2} ln(1 - 2P - Q) - \frac{1}{4} ln(1 - 2Q) \tag{5.5}$$

where

$$P = (1/4)(1 - 2e^{-4(\alpha+\beta)t} + e^{-8\beta t}) \tag{5.6}$$

$$Q = (1/2)(1 - e^{-8\beta t}) \tag{5.7}$$

More complicated models results in more complex estimates.

The above distance measures do not consider variations of evolutionary rate among sites. It has been shown that the number of substitutions per site approximately follows the negative binomial distribution (Uzzell and Corbin, 1971) and thus the evolutionary rate variation can be modelled by the gamma distribution. Under this assumption, gamma distance measures are proposed (Johnson and Kotz, 1969; Kocher and Wilson, 1991). These distance measures generally are more realistic than non-gamma ones but they have larger variances and thus do not necessarily give better phylogenetic inference results (Nei and Kumar, 2000).

Once the matrix of pairwise distances between all pairs of taxa has been computed, a tree building method is then applied. The method makes use of the distances to build a tree in which every two taxa have a distance in the tree that is as close as possible to their distance as specified by the matrix.

The simplest distance tree building method is the *unweighted pair-group method using arithmetic averages (UPGMA)* proposed by Sokal and Michener (1958). The method starts with the given distance matrix and with a forest of rooted trees, one per taxon, each consisting of a single node representing that taxon. It then iterates over a number of steps, joining trees until a single tree is the result. At each step the algorithm selects the two trees, T1 and T2, whose roots, as implied by the distance matrix, are closest. A new hypothetical parent taxon, t, is made as most recent common ancestor of T1 and T2. The distance between the *roots* of any two trees, A and B, is defined as

$$d_{A,B} = \sum_{a \in A} \sum_{b \in B} \frac{1}{|A||B|} d_{ab} \tag{5.8}$$

where a and b are (real, non-hypothetical) taxa in A and B respectively. The method iterates until a single trees covers all taxa. The re-computation of distances implicitly assumes uniform mutation rates across all branches. Therefore, UPGMA often gives reasonably good phylogenetic tree topologies when the evolutionary rate is more or less constant (Nei et al., 1983; Takezaki and Nei, 1996). However, it tends to produce topological errors when the evolutionary rate is not constant and the length of each sequence is relatively short (Takezaki and Nei, 1996).

The UPGMA method assumes the molecular clock hypothesis and produces an ultrametric tree – a tree in which the distances from the root to all leaves are the same. The assumption may not hold and an ultrametric tree is not realistic in general. Furthermore, when there is a tie for the nearest pair of nodes, selecting different pairs may lead to different tree topologies. To overcome these problems, some tree construction methods seek a tree that optimises a criterion. One such optimality criterion is *least squares (LS)*. The *ordinary LS* method, proposed by Cavalli-Sforza and Edwards (1967), suggests the tree with the least sum of squared differences:

$$R_s = \sum_{i<j} (d_{ij} - e_{ij})^2 \tag{5.9}$$

where d_{ij} is the *observed distance* between two taxa i and j, and e_{ij} is the path length between i and j in the tree. The Fitch-Margoliash method (Fitch and Margoliash, 1967) applies the *weighted LS* criterion, e.i., it attempts to minimise

$$R_s = \sum_{i<j} \frac{(d_{ij} - e_{ij})^2}{d_{ij}} \tag{5.10}$$

Unfortunately some LS methods can produce unrealistic trees due to the introduction of negative branch lengths. However, the addition of the constraint disallowing negative branch lengths in work by Kuhner and Felsenstein (1994) is shown to improve the accuracy of LS methods.

The *minimum evolution (ME)* principle, proposed by Edwards and Cavalli-Sforza (1963) and Kidd and Sgaramella-Zonta (1971), advocates the tree with the smallest sum of all branch lengths, that is

$$S = \sum_{i}^{T} b_i \tag{5.11}$$

where b_i is the length of branch i. The theoretical basis of ME methods is presented in Rzhetsky and Nei (1993); the work shows that if the distance measure is statistically unbiased, the expectation of the sum of branch lengths of the true tree is the smallest among all possible trees. Similar to LS methods, ME methods often require an exhaustive search

in the tree space. Several approximation algorithms are developed to improve the running time. The stepwise algorithm proposed by Kumar (1996) utilises the beam search strategy to restrict the search space. While this algorithm is significantly faster than an exhaustive search, it is virtually guaranteed to find the optimal tree under the ME criterion. A greedy approach is proposed by Rodin and Li (2000). This method is even faster than the stepwise algorithm, with a negligible loss of accuracy.

A simplified version of the ME methods is the *neighbor joining (NJ)* method developed by Saitou and Nei (1987). This method greedily utilises the ME principle at each stage of building a tree. In this method, all taxa are initially arranged as a star topology L with no interior nodes. Two taxa are regarded as neighbours if they are connected by a single node. At each stage, the method finds the two nearest neighbours, which are the two taxa i and j with the minimum

$$D_{ij} = d_{ij} - (r_i + r_j) \tag{5.12}$$

where

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik} \tag{5.13}$$

and $|L|$ is the number of leaves in the star topology L. Taxa i and j are then removed from L and are placed under a new node k with the branch lengths

$$\begin{aligned} d_{ik} &= \frac{1}{2}(d_{ij} + r_i - r_j) \\ d_{jk} &= \frac{1}{2}(d_{ij} + r_j - r_i) \end{aligned} \tag{5.14}$$

k is then considered as a single taxon, and is added to L as a leaf. The procedure is repeated until the final tree is produced.

The NJ method can produce a near optimal ME tree in practical time. However, it is often criticised for producing only one tree so that the quality of the tree is not easily evaluated. For this reason, several authors Rzhetsky and Nei (1992); Kumar (1996) apply NJ to find the starting tree in other optimisation tree building methods such as LS and ME. The NJ tree is perturbed by various branch swapping algorithms to produce similar trees for evaluation.

Though distance methods for phylogenetic tree construction are generally fast and able to produce reasonable trees, they are not preferred by many practitioners. These methods are often criticised for the loss of information since the sequences are represented by distances between them. Branch lengths in trees produced by a distance method sometimes can have negative values and thus be biologically impossible. For that reason, a tree building method based on optimisation at the character level is often used.

Two notable examples in this class are *maximum parsimony (MP)* and *maximum likelihood (ML)* methods.

MP methods are among the earliest methods of phylogenetic analyses. They were used by Eck and Dayhoff (1966), Fitch (1971) and Hartigan (1973) for inferring phylogenies from amino acid and nucleotide sequences. The central idea of the MP methods is based on Ockham's Razor; the tree which requires the smallest number of changes (i.e., substitutions, insertions and deletions) to explain the observed data is chosen as the best tree. For a given tree topology, the smallest number of substitutions at a site over all possible ancestral states at the site is considered as the *parsimony score* of the site. The parsimony scores of all sites are summed to give the parsimony score of the tree. The tree with the least parsimony score, is called the most parsimonious tree, and is selected as the best phylogeny.

To take into account the variation of evolutionary rate among sites and substitution rate among nucleotides, the *weighted maximum parsimony* method gives different weights to sites with various informative levels (e.g. synonymous and nonsynonymous sites) (Farris, 1969) and to different kinds of substitutions – notably transitions and transversions (Swofford and Begle, 1993). These weights can be estimated by an expectation maximisation approach (Williams and Fitch, 1990). These strategies do not necessarily give the most parsimonious tree but significantly improve the probability of obtaining the correct tree (Huelsenbeck, 1995; Nei et al., 1995).

Theoretically, MP methods are expected to find the true tree if there are no parallel and backward substitutions and there are sufficient parsimonious-informative sites (Nei and Kumar, 2000). However, nucleotide sequences are subject to multiple substitutions in practice, and thus MP methods have a high probability of producing incorrect topologies, especially when the lengths of the sequences are small. Moreover, Felsenstein (1978) shows that MP methods tend to give incorrect trees if the evolutionary rates across lineages varies greatly, even if an infinite number of nucleotides are available for analysis. Another problem with MP methods is that long branches (*long branch attraction* (Hendy and Penny, 1989)) or short branches (*short branch attraction* (Nei, 1996)) tend to attract each other in the reconstructed tree. Furthermore, MP methods generally only give phylogeny topologies without branch lengths. Despite these shortcomings, MP methods are still among the most popular methods because they are generally independent of a substitution model, and they are the only methods that make use of insertions and deletions for phylogenetics analyses.

ML methods for phylogenetic inference were first introduced by Cavalli-Sforza and Edwards (1967) for gene frequency data, and later by Felsenstein (1981) for nucleotide data. These methods compute the likelihood of all possible trees, and suggest the maximum likelihood one. The likelihood of a phylogenetic tree is the probability of the observed data given the tree, and is calculated as the product of the probability at each site

given the tree:

$$L = Pr(D|T) = \prod_{i=1}^{m} Pr(D^{(i)}|T) \tag{5.15}$$

where m is the number of sites in the alignment table, and $D^{(i)}$ are the data at site i. The probability of data at each site $Pr(D^{(i)}|T)$ is computed as the sum of the probabilities of all possible assignments $x_1, x_2..x_n$ to site i at n internal nodes:

$$Pr(D^{(i)}|T) = \sum_{x1,x2,...,xn} Pr(x_1, x_2, ..., x_n|T) \tag{5.16}$$

The probability of the assignment of $x_1, x_2..x_n$ to n internal nodes is:

$$Pr(x_1, x_2, ..., x_n|T) = Pr(x_1) \prod_{k,l} Pr(x_k|x_l, d_{kl}) \tag{5.17}$$

where $Pr(x_1)$ is the prior probability of the site at the tree root (node 1 is the root) being x_1, and d_{kl} is the length of the branch (k,l) where k is a daughter node of l. If k is an external node, x_k is the observed data at this node. $Pr(x_1)$, the prior probability of amino acid or nucleotide x_1, can be estimated by the frequency of x_1 from the observed data. $Pr(x_k|x_l, d_{kl})$ is calculated by using one of the substitution models mentioned in Section 5.2.

One serious problem of ML methods is the high cost of computation. The calculation of the likelihood at one site in a tree with n external nodes involves 4^n and 20^n terms for nucleotide and amino acid data respectively. Felsenstein (1981) presents a pruning technique which significantly reduces the number of calculations. Estimation of branch lengths for a given tree topology in ML methods can be done by an expectation maximisation procedure.

Strictly speaking, MP and ML methods do not construct trees. They instead provide respective objective functions to select the best phylogeny from a set of possible ones. They therefore, rely on a search mechanism to produce trees for examination. The naive exhaustive search in the tree space is virtually impossible, especially when the number of taxa is relatively large. In this case, a heuristic search is often applied. Generally, a near optimal tree, such as that obtained from NJ method, is used as a starting provisional tree. A branch swapping algorithm, such as the *nearest neighbour interchanges*, *subtree pruning regrafting* and *tree bisection-reconnection* (Swofford and Begle, 1993), is used to generate trees that are different to the provisional tree by small changes. The *branch and bound* algorithm, developed by Hendy and Penny (1982), examines only trees that have shorter tree lengths to the previously examined trees. This method is also used in MP methods and is guaranteed to find the most parsimonious tree.

Recently, *phylogeny Bayesian inference* (BI) (Allison and Wallace, 1994; Yang and Rannala, 1997) has gained popularity in phylogenetic analyses. Similar to ML methods, BI methods are based on probability to find the most probable tree. However, unlike ML methods, which favour the tree with the maximum likelihood $Pr(D|T)$, BI methods advocate the tree with the maximum posterior probability tree given the data $Pr(T|D)$. The posterior probability of a tree is related to the likelihood and the prior probability of a tree by Bayes's theorem. Like MP and ML, BI gives a *criterion* to compare trees and requires a search method such as exhaustive or greedy search, or rather naturally, stochastic sampling, to propose candidate trees.

5.4 An Information Theory Distance Measure

Recall the transmission example presented in Figure 4.1 in Chapter 4 where a sequence X is to be efficiently transmitted over a reliable channel. X is compressed by a lossless compression model prior to being transmitted. Since X can be recovered at the destination, no information is lost during transmission. In other words, the amount of information that goes across the channel, i.e., the length of the encoded message of X, is the total information contained in X. This is the *information content* of X, denoted $\mathcal{I}(X)$. If a sequence Y *related* to X is available to both the sender and the receiver, the sender can compress X on the background knowledge of Y. The amount of information actually transmitted in this case is called the *conditional information content* of X given Y, denoted $\mathcal{I}(X|Y)$. $\mathcal{I}(X|Y)$ is expected to be shorter than $\mathcal{I}(X)$ because the sender does not need to transmit the information in X that is also contained in Y. The more related the two sequences are, the more information the two sequences share, and hence the shorter message is transmitted. The shared information of X and Y is called the *mutual information* of X and Y and can be computed as the difference between the information content and the conditional information content: $\mathcal{I}(X;Y) = \mathcal{I}(X) - \mathcal{I}(X|Y)$.

This chapter proposes the use of mutual information for measuring genetic distances (and hence elapsed time) between sequences for phylogenetic analyses. This is a departure to the standard distance measures which estimate distances using the number of observed substitutions. The information of a sequence, and the conditional information of that sequence given another, can be estimated by a compression model. The difference of the two quantities is an approximation to the mutual information of the two sequences.

The proposed distance measure is first discussed based on the simple Jukes-Cantor model, and is later generalised to more complex models. Recall that the Jukes-Cantor model assumes the probability of substituting a nucleotide for another during a time unit is equal to a value α. The matrix of substitution rates of Jukes-Cantor model after one time unit is presented in Table 5.2(a). The nucleotide substitution probabilities at a

site after t time units are described by the matrix $Q_t = S^t$. Because S is a symmetric matrix, and all of its off-diagonal elements are equal, off-diagonal elements of Q_t are also equal to a value q_t. The elements on the diagonal are $1 - 3q_t$ because the sum of a row in the matrix is 1. The matrix Q_t thus is

$$Q_t = S^t = \begin{vmatrix} 1-3q_t & q_t & q_t & q_t \\ q_t & 1-3q_t & q_t & q_t \\ q_t & q_t & 1-3q_t & q_t \\ q_t & q_t & q_t & 1-3q_t \end{vmatrix}$$

Since $Q_{t+1} = Q_t S$, the off-diagonal elements of Q_{t+1} have the value:

$$\begin{aligned} q_{t+1} &= (1-3q_t)\alpha + q_t(1-3\alpha) + 2q_t\alpha \\ &= \alpha + q_t - 4q_t\alpha \end{aligned} \tag{5.18}$$

The equation can be rewritten as:

$$q_{t+1} - q_t = \alpha - 4q_t\alpha \tag{5.19}$$

If q is considered as a continuous function on variable t, then $q_{t+1} - q_t$ is the change dq of function q over a short time period dt. This gives the differential equation:

$$\frac{dq}{dt} = \alpha - 4q\alpha \tag{5.20}$$

Solving the equation with the initial condition $q = 0$ at $t = 0$ gives:

$$q_t = \frac{1}{4} - \frac{1}{4}e^{-4\alpha t} \tag{5.21}$$

Assume that two homologous sequences X and Y diverged from their most recent ancestor sequence, S, t time units ago, and they evolved under the Jukes-Cantor model. Further assume that S is uniformly distributed, and hence so are X and Y. The entropy rate of X and Y is thus equal to

$$\mathcal{I}(X) = \mathcal{I}(Y) = 4 \times (-\frac{1}{4}\log_2\frac{1}{4}) = 2\ bits \tag{5.22}$$

The rates of substitutions between X and Y are specified by Q_{2t}. If the nucleotide at a specific site y_i in Y is, say, A, the probabilities of the corresponding site x_i in X being A, C, G and T are $1 - 3q_{2t}$, q_{2t}, q_{2t} and q_{2t} respectively. The conditional information content of x_i, if y_i is known to be A, is

$$\mathcal{I}(x_i|y_i = A) = -(1-3q_{2t})\log_2(1-3q_{2t}) - 3q_{2t}\log_2 q_{2t} \tag{5.23}$$

Let the frequencies of nucleotides A, C, G and T in Y be π_A, π_C, π_G and π_T respectively. The expected conditional information content of x_i given y_i is

$$\begin{aligned}\mathcal{I}(x_i|y_i) &= \mathcal{I}(x_i|y_i = A)\pi_A + \mathcal{I}(x_i|y_i = C)\pi_C + \mathcal{I}(x_i|y_i = G)\pi_G + \mathcal{I}(x_i|y_i = T)\pi_T \\ &= (\pi_A + \pi_C + \pi_G + \pi_T)(-(1 - 3q_{2t}) \log_2(1 - 3q_{2t}) - 3q_{2t} \log_2 q_{2t}) \\ &= -(1 - 3q_{2t}) \log_2(1 - 3q_{2t}) - 3q_{2t} \log_2 q_{2t}\end{aligned} \tag{5.24}$$

Let $p = e^{-8\alpha t}$, then $0 < p < 1$ and

$$q_{2t} = \frac{1-p}{4} \tag{5.25}$$

$$1 - 3q_{2t} = \frac{1+3p}{4} \tag{5.26}$$

so the conditional entropy rate

$$\mathcal{I}(X|Y) = -\frac{1+3p}{4} \log_2 \frac{1+3p}{4} - 3\frac{1-p}{4} \log_2 \frac{1-p}{4} \tag{5.27}$$

The conditional entropy rate $\mathcal{I}(X|Y)$ is bounded by the entropy rate $\mathcal{I}(X)$ (e.i., $\mathcal{I}(X|Y) \leq \mathcal{I}(X)$) and $\mathcal{I}(X|Y)$ approaches to $\mathcal{I}(X)$ when t approaches infinity. Therefore, the ratio $\mathcal{I}(X|Y)/\mathcal{I}(X)$ is bounded by 1. Figure 5.1 plots the function representing the ratio against t. As Equation 5.27 involves logarithm calculation and thus is too complex to manipulate, a simpler approximate function is used instead. Function fitting shows that the function $y = 1 - p^2$ where $p = e^{-8\alpha t}$ fits the ratio $\mathcal{I}(X|Y)/\mathcal{I}(X)$ closely. The plot of the function is also presented in Figure 5.1.

By the approximation,

$$\frac{\mathcal{I}(X|Y)}{\mathcal{I}(X)} \approx 1 - p^2 \tag{5.28}$$

$$= 1 - e^{-16\alpha t} \tag{5.29}$$

thus

$$e^{-16\alpha t} = \frac{\mathcal{I}(X) - \mathcal{I}(X|Y)}{\mathcal{I}(X)} \tag{5.30}$$

or

$$t = -\frac{1}{16\alpha} \ln \frac{\mathcal{I}(X) - \mathcal{I}(X|Y)}{\mathcal{I}(X)} \tag{5.31}$$

In other words, time t is proportional to

$$D = -\ln \frac{\mathcal{I}(X) - \mathcal{I}(X|Y)}{\mathcal{I}(X)} \tag{5.32}$$

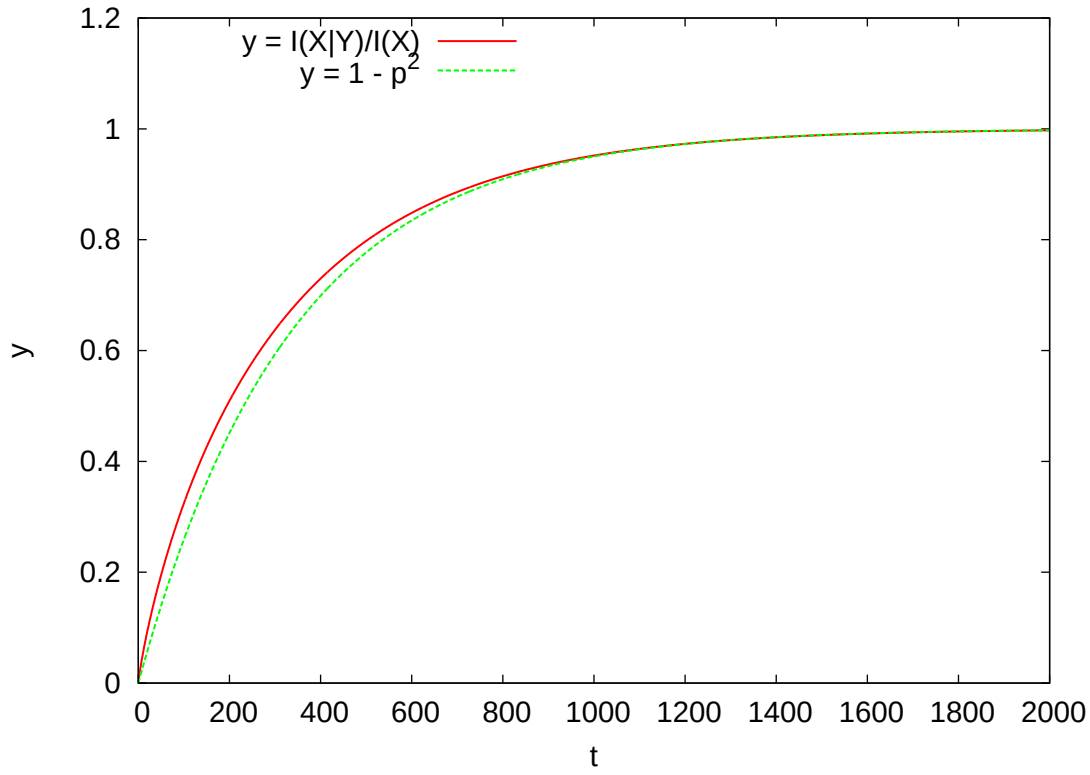


Figure 5.1: Plots of the ratio $y = \frac{\mathcal{I}(X|Y)}{\mathcal{I}(X)}$ and the function $y = 1 - p^2$ where $p = e^{-8\alpha t}$.

and thus this measure D is proposed to be used to estimate the genetic distance between any two sequences.

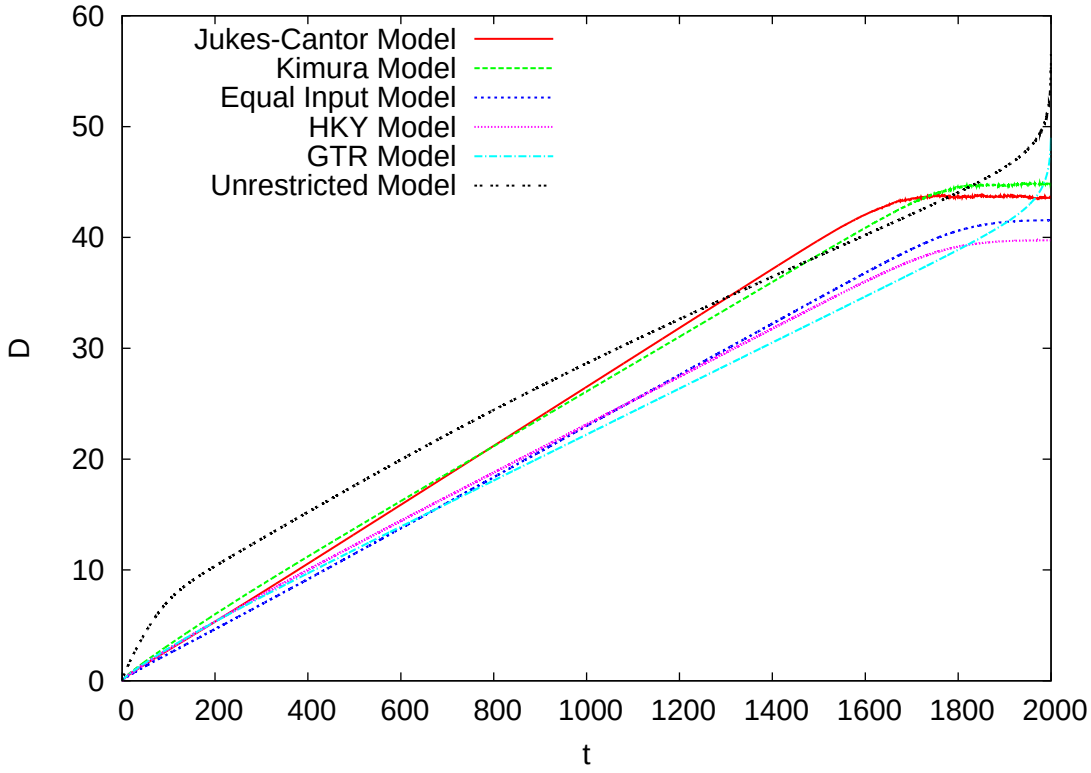
Equation 5.32 is intuitively explained by the decay of mutual entropy rate $\mathcal{I}(X;Y) = \mathcal{I}(X) - \mathcal{I}(X|Y)$ of the two sequences over time. At the time two sequences X and Y split from their most recent common ancestor, they are the same and thus the mutual information is as much as $\mathcal{I}(X)$. As time approaches infinity, the mutual information becomes zero. Assume a fraction γ of the mutual information is lost after a time unit. The mutual information of the two sequences after t time units is

$$\mathcal{I}(X,Y) = \mathcal{I}(X) - \mathcal{I}(X|Y) = \gamma^t \mathcal{I}(X) \tag{5.33}$$

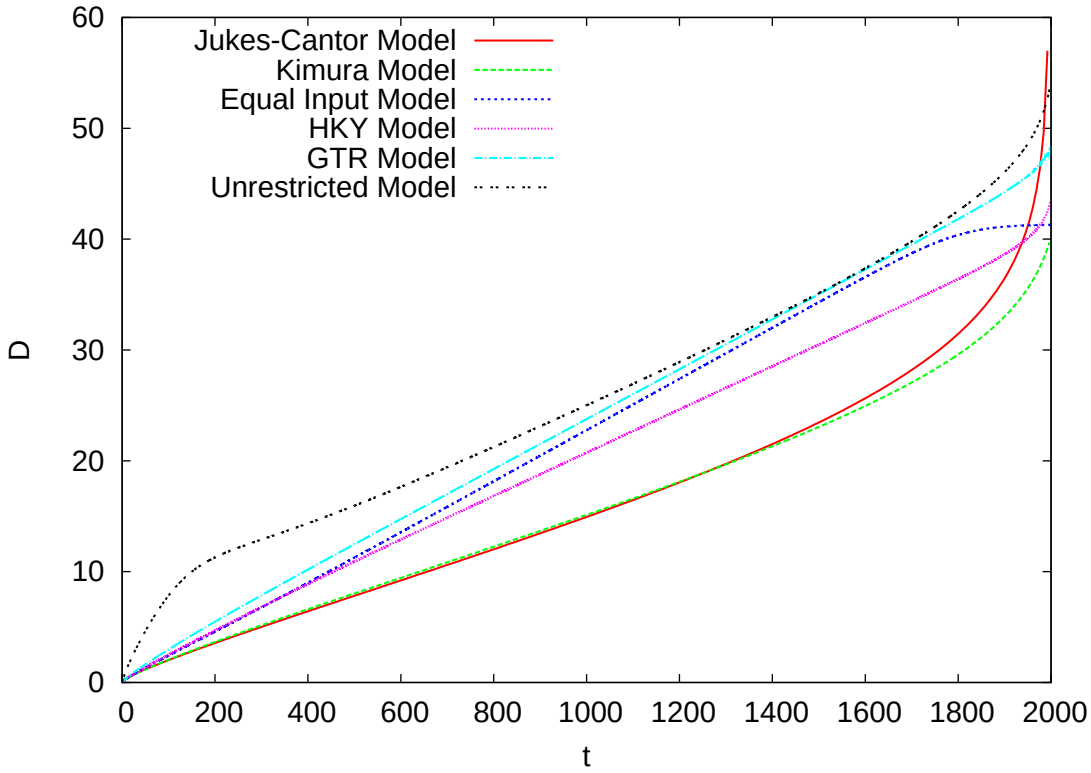
or

$$\begin{aligned} t &= \frac{1}{\ln \gamma} \ln \frac{\mathcal{I}(X) - \mathcal{I}(X|Y)}{\mathcal{I}(X)} \\ &= \frac{1}{\ln \gamma} D \propto D \end{aligned} \tag{5.34}$$

The discussion so far is for the simple Jukes-Cantor substitution model and uniformly distributed sequences for the sake of simplicity. However, the approximation is



(a) Uniform sequences



(b) Skewed sequences

Figure 5.2: The approximate linearity of the proposed distance measure D.

also applicable for other models and for non-uniformly distributed data. The theoretical mutual entropy rate of two sequences over time for each model is numerically computed and is plotted in Figure 5.2. The parameter setting for each model is chosen so that the evolution reaches equilibrium (i.e., $Q_t \simeq Q_{t+1}$) when t is close to 2000. Figures 5.2a and 5.2b respectively show plots of the proposed distance D against time t for a set of uniformly distributed data, and a set of statistically skewed data where the frequencies of A, C, G and T are 0.1, 0.4, 0.4 and 0.1 respectively. The six substitution models presented in Table 5.2 are considered.

In most models, except for the unrestricted model, the measure D is approximately proportional to time t up to the point where the evolution approaches equilibrium. After this point, the relationship between two sequences is random and the mutual information between two sequences becomes zero. The distance between the two sequences therefore no longer makes sense. For the unrestricted model, except for the extremely closely related and extremely distantly related sequences, the plot for the proposed distance against time is nearly a straight line. The observation is the same for both uniformly distributed and skewed data cases.

In practice, calculating the entropy and conditional entropy of sequences is not possible because the statistical nature of biological sequences is not fully known. However, their values can be estimated by lossless compression. The entropy, $\mathcal{I}(X)$, of a sequence X is approximately equal to the compressed size of X. Similarly, the conditional entropy $\mathcal{I}(X|Y)$ is estimated by the compression of X on the background knowledge of Y. These estimates are denoted as $\hat{\mathcal{I}}(X)$ and $\hat{\mathcal{I}}(X|Y)$.

The closest approximation of the entropy is given by the best possible compression. To compress a short sequence such as a single gene, a simple adaptive Markov model appears to be among the best performers. The Markov model compresses the sequence by scanning through the sequence, and at any position of the sequence, it simply counts the number of each type of nucleotides seen previously and accordingly, builds a distribution based on the nucleotide frequencies. The distribution is used to compress the next symbol and hence to estimate the entropy of the sequence The entropy rate of the sequence estimated by the Markov model is approximately

$$\hat{\mathcal{I}}(X) = -\sum_{i=A,C,G,T} \hat{\pi}_i^x \log_2(\hat{\pi}_i^x) \tag{5.35}$$

where $\hat{\pi}_i^x$ is the estimated frequency of each nucleotide i (A, C, G or T) in X.

Suppose a sequence Y, *homologous* to X, is available, and the two sequences have been reliably aligned. The conditional compression of X on the background knowledge of Y is also performed by a procedure similar to the above. The only difference is that the compressor gathers statistics from the number of substitutions and builds an estimated

substitution matrix $\hat{Q}$ in which each element $\hat{q_{ij}}$ is the frequency of nucleotide y_i in Y matches with nucleotide x_j in X.

$$\hat{\mathcal{I}}(X|Y) = -\sum_{i=A,C,G,T} \hat{\pi}_i^y \sum_{j=A,C,G,T} \hat{q}_{ij} \log_2 \hat{q}_{ij} \tag{5.36}$$

In principle, $\hat{\mathcal{I}}(X) - \hat{\mathcal{I}}(X|Y)$ and $\hat{\mathcal{I}}(Y) - \hat{\mathcal{I}}(Y|X)$ should be equal because they both give the mutual entropy rate of X and Y. However, the computation of entropy using compression is approximate. The evolution of sequences also involves many random events. Therefore, the two values are not necessarily identical. A better estimate of the distance D is

$$\hat{D} = -\ln \frac{\hat{\mathcal{I}}(X) - \hat{\mathcal{I}}(X|Y) + \hat{\mathcal{I}}(Y) - \hat{\mathcal{I}}(Y|X)}{\hat{\mathcal{I}}(X) + \hat{\mathcal{I}}(Y)} \tag{5.37}$$

This is the proposed distance measure (XMDistance) in this work.

5.5 Experimental Results

This section describes experiments on the XMDistance distance measure. Subsection 5.5.1 presents the comparison of XMDistance to several standard phylogenetic analysis methods on a simulated data set. Real data were used in other experiments, as described in Subsection 5.5.2.

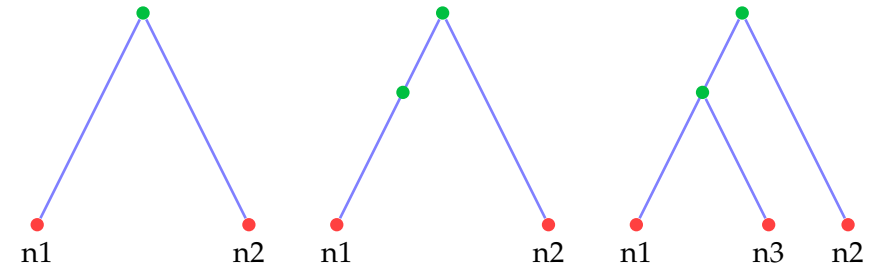
5.5.1 Simulated Data

Since the true phylogenies of real species are unknown, a set of simulated data was used for this experiment. The benefits of simulated data are that the true phylogenies are known, and the sequences are reliably aligned from the generation. Furthermore, computer simulation allows the data to be generated by a controlled process. The data set in this experiment contains 2000 phylogenies of varying sizes and levels of divergence. Each taxon in a phylogeny is represented by a sequence of length 10 kilobases.

The level of divergence of a phylogeny is defined as the date of the most recent common ancestor of all species in the phylogeny. The higher the divergence level of a phylogeny is, the greater the elapsed time since the species in the phylogeny split and the more distantly they are related. The sequences representing these "species", hence, show a higher level of mutation. All species in the phylogeny are assumed to have evolved at a constant rate, and thus the distance from a taxa in the phylogeny to the root is equated to the divergence level of the tree.

A phylogeny with a certain divergence level and topology was created as follows. The tree topology was initialised to two taxa equidistant from the root of the tree. At

each subsequent stage, a new taxon was added to the tree. One stage of the process is illustrated in Figure 5.3. A point was selected randomly on one of the branches of the tree in step 5.3b. A new taxon was then created and attached to the tree at the selected point. The length of the new branch was derived so that the distance from the taxon to the root of the tree is equal to the divergence level of the tree. Steps 5.3b and 5.3c were repeated until the tree reached the predefined number of taxa.



(a) Initialise the phylogeny (b) Pick a random inner node (c) Add the new taxon

Figure 5.3: The generation of a random phylogeny.

Once the phylogeny was created, the ROSE package (Stoye et al., 1998) was used to generate sequences representing the species of the tree. The HKY model, with the mean substitution of 1 percent mutations per site per time unit, was used for the generation of data. As sequences in this experiment were assumed to be aligned and without gaps, the generation of sequences did not include deletions and insertions.

The data set contains phylogeny trees with various properties. Phylogenies in the set are in 10 sizes – 5, 10, 15, ... and 50 taxa – and are in 10 divergence levels – 10, 20, ... and 100 time units. For a particular size and a particular divergence level, 20 trees were generated. In total, the data set contains 2000 trees.

The proposed genetic distance measure was compared with several standard tree building methods on the data set. Specifically, the methods selected for comparison were from the PHYLIP package (Felsenstein, 2005) including *dnaml* (maximum likelihood), *dnapars* (maximum parsimony) and *dnadist* (distance). Both dnaml and dnapars reconstructed a phylogeny from the sequences directly, while dnadist and XMDistance generated a matrix of pairwise genetic distances between any two taxa. The neighbour joining method (Saitou and Nei, 1987) was used to generate phylogenies from the distance matrices computed by XMDistance and dnadist. All programs were run with their default parameters. The substitution model used in dnadist was the F84 model (Kishino and Hasegawa, 1989; Felsenstein and Churchill, 1996), which is similar to the HYK model used to generate the data. This gave dnadist an advantage over XMDistance in estimating genetic distances.

The performance of each tree building method was evaluated based on how the topology and branch lengths of each reconstructed tree compared with the correct tree. To assess the reconstruction of correct topologies, two tree distance criteria were used. The first criterion, denoted *topology difference*, simply gives a score of 0 if the reconstructed tree has the same topology as the true tree, and a score of 1 otherwise. Essentially, this measure gives the number of phylogenies where the tree building method gives an incorrect topology. The second criterion is the *symmetric difference* (Robinson and Foulds, 1981). It considers all possible partitions of the taxa created by removing one branch from a tree. It counts the number of partitions that are given by one of the trees but not both of them.

Two other criteria took branch *lengths* into account. The first criterion, *SSD*, is the sum of the squared differences of the distances between every corresponding pair of leaves in the trees. The second criterion, *branch score difference* (Kuhner and Felsenstein, 1994) considers all possible partitions of a tree, as in the symmetric difference distance. This criterion however, gives the sum of the squared differences of the lengths of the branches, the removal of which creates identical partitions on the two trees; if a particular partition does not present on a tree, a branch of length 0 is considered to have created the partition for that tree. As different methods may have different approaches to estimate branch lengths, phylogenies produced by different methods may have different branch length scales. In this experiment therefore, each phylogeny was normalised so that the sum of all branch lengths is equal to 1 before being evaluated by the SSD and the branch score difference measures.

The experiment was performed on a desktop computer equipped with a 2.33 Ghz Pentium Core 2 Duo CPU and 8 GB of memory. The machine ran Linux Ubuntu 8.04. The performance results of the selected tree building methods on the data set are presented in Table 5.3 and Table 5.4. Table 5.3 shows the performance on the data set grouped by tree sizes, and Table 5.4 shows that on the data set grouped by divergence levels. The performance criteria were applied to each method which was run on 200 example trees in each group. The tables show the sum of each criterion for 200 trees in each group and their totals. The total running times in seconds of the four tree building methods for each group are also presented in Table 5.3.

Generally, the maximum likelihood method performed the best among the four methods on the data set. XMDistance outperformed the maximum parsimony and the standard distance methods, in both producing correct topologies and estimating branch lengths. The maximum parsimony method was inferior to the other three methods. In particular, out of the 2000 phylogenies, dnaml method reconstructed 81 incorrect topologies while XMDistance produced 146 incorrect topologies. The number of incorrect topologies produced by the standard distance measure and the maximum parsimony methods were 167 and 680 respectively. The symmetric difference scores of trees produced by dnaml, XMDistance, dnadist and dnapars were 190, 324, 378 and 2470 respectively. The

Table 5.3: Performance evaluation of tree building methods on the data set, grouped by tree sizes.

	S=5	S=10	S=15	S=20	S=25	S=30	S=35	S=40	S=45	S=50	Total
Topology Difference Criterion											
dnaml	0	0	0	0	2	6	6	16	24	27	81
dnapars	0	0	3	26	44	81	104	127	140	155	680
dnadist	0	0	0	2	4	12	21	31	45	52	167
XMDistance	0	0	0	0	5	8	16	28	39	50	146
Symmetric Difference Criterion											
dnaml	0	0	0	0	4	12	14	38	58	64	190
dnapars	0	0	6	52	114	210	310	502	550	726	2470
dnadist	0	0	0	4	8	26	44	72	100	124	378
XMDistance	0	0	0	0	10	18	34	64	88	110	324
SSD Criterion											
dnaml	0.25	0.39	0.47	0.50	0.54	0.61	0.63	0.68	0.73	0.71	5.52
dnapars	1.76	3.75	7.10	9.89	12.11	13.87	14.28	16.14	16.40	16.93	112.21
dnadist	0.92	2.37	3.52	4.14	4.50	4.94	5.43	5.82	5.94	6.66	44.26
XMDistance	0.68	1.69	2.43	2.76	3.22	3.41	3.64	3.82	4.09	4.18	29.92
Branch Score Difference Criterion											
dnaml	4.70	3.02	2.57	2.28	2.09	1.90	1.79	1.69	1.57	1.51	23.12
dnapars	9.29	6.43	5.40	4.62	4.16	3.78	3.55	3.31	3.16	3.72	47.42
dnadist	9.36	6.30	4.88	4.04	3.62	3.21	2.96	2.79	2.56	2.45	42.19
XMDistance	7.59	5.12	4.20	3.49	3.13	2.90	2.72	2.54	2.37	2.24	36.31
Running Time (in seconds)											
dnaml	114	948	3085	7021	12715	20249	30053	41431	55076	70289	240981
dnapars	6	209	2299	7643	14893	23432	31966	42424	52232	63544	238646
dnadist	3	16	51	118	215	342	507	696	921	1175	4044
XMDistance	60	72	87	110	140	178	221	270	317	371	1826

Table 5.4: Performance evaluation of tree building methods on the data set, grouped by diversity levels.

	T=10	T=20	T=30	T=40	T=50	T=60	T=70	T=80	T=90	T=100	Total
Topology Difference Criterion											
dnaml	5	3	1	3	8	8	15	10	12	16	81
dnapars	15	21	43	60	72	71	101	94	101	102	680
dnadist	21	17	11	11	15	15	24	14	19	20	167
XMDistance	21	11	9	11	12	13	22	15	16	16	146
Symmetric Difference Criterion											
dnaml	10	6	2	6	16	16	30	26	38	40	190
dnapars	12	44	108	134	224	262	394	350	466	476	2470
dnadist	48	40	22	26	32	32	52	36	46	44	378
XMDistance	50	28	18	22	24	28	46	32	42	34	324
SSD Criterion											
dnaml	1.02	0.54	0.43	0.39	0.38	0.40	0.46	0.52	0.62	0.76	5.52
dnapars	1.55	2.34	3.98	6.12	8.39	11.89	14.05	17.54	20.94	25.41	112.21
dnadist	1.16	0.85	1.07	1.54	2.28	3.25	4.62	6.95	9.62	12.91	44.26
XMDistance	4.45	3.93	3.69	3.43	3.00	2.63	2.45	2.18	2.09	2.08	29.92
Branch Score Difference Criterion											
dnaml	3.25	2.37	2.13	1.96	2.02	1.98	2.08	2.23	2.52	2.59	23.12
dnapars	3.96	2.74	3.02	3.52	4.08	4.76	5.48	5.98	6.83	7.05	47.42
dnadist	3.38	2.68	2.71	2.94	3.38	3.79	4.52	5.31	6.15	7.33	42.19
XMDistance	4.43	4.01	3.89	3.63	3.64	3.48	3.48	3.37	3.18	3.19	36.31

order of performance of these methods was the same for estimating branch lengths. Both SSD and branch score difference criteria of the maximum likelihood method were better than that of XMDistance method, which in turn performed better than dnadist and dnapars programs in estimating branch lengths.

The running times of the four methods are also reported in Table 5.3. For the dnadist and XMDistance methods, the running time for inferring a phylogeny includes the time to compute the distance matrix and the time to infer the tree by the neighbour joining method. Among the four tree building methods, XMDistance was the fastest method, especially for large phylogenies. It needed just over 3 minutes to reconstruct all 2000 phylogenies, and was twice as fast as dnadist, the second fastest method. In contrast, the maximum likelihood method took 240981 seconds, or about 67 hours to infer the 2000 phylogenies. The maximum parsimony method was slightly faster than the maximum likelihood method.

The standard distance and maximum parsimony methods were faster than XMDistance on small trees, such as trees with less than 15 taxa. However, with increasing phylogeny sizes, XMDistance was the fastest. Since XMDistance was implemented in Java and it relied on BioJava (Holland et al., 2008) for input and output operations, time for I/O was much longer than the Phylip package which was implemented in C. It was noted that for inferring small phylogenies (e.g., phylogenies with five taxa), XMDistance program spent 50% of its running time reading the sequences.

5.5.2 Real Data

The proposed distance measure, XMDistance, was applied to a real mtDNA data set. The data set, prepared by Cao et al. (1998), contains the complete mitochondrial genomes of 20 mammal species: cow, fin whale, blue whale, harbour seal, gray seal, horse, cat, rhinoceros, mouse, rat, human, chimpanzee, bonobo, gorilla, Bornean orangutan, Samatran orangutan, gibbon, opossum, wallaroo and platypus. These sequences were manually aligned and the alignments were carefully checked by eye. All gaps and ambiguous alignment sites were excluded. The alignment process resulted in 20 sequences which were 9993 bases long. XMDistance was used to compute the matrix of pairwise distances and the neighbour joining method (Saitou and Nei, 1987) implemented in the PHYLIP package (Felsenstein, 2005) was used to generate the phylogeny. Platypus was used as the outgroup to determine the tree root.

The phylogeny for the 20 mammal species inferred by XMDistance is presented in Figure 5.4. The phylogeny is in complete agreement with the analysis by Cao et al. (1998). In general, the XMDistance tree correctly placed respective species into the correct groups, i.e., marsupials (opossum and wallaroo), rodents (mouse and rat), primates (human, chimpanzee, bonobo, gorilla, orangutan and gibbon), and ferungulates (the rest

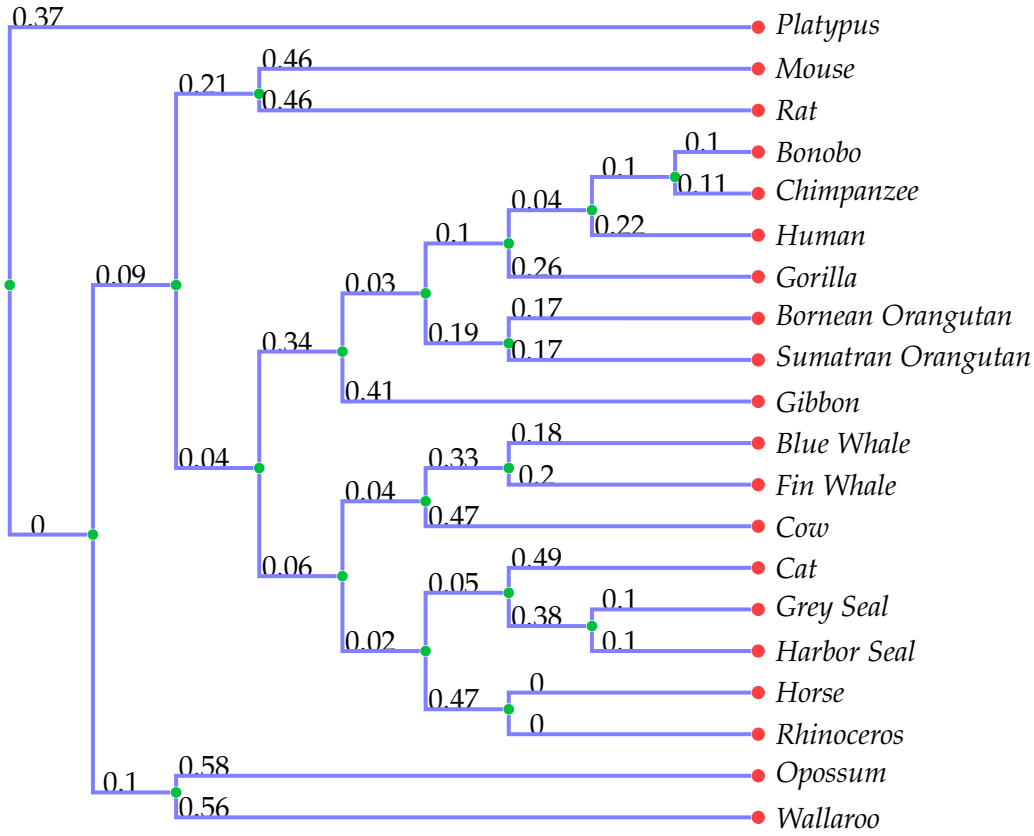


Figure 5.4: The phylogeny of mammals inferred by XMDistance. The number in a branch represents the relative length of the branch.

of the species except the outgroup platypus). Within the ferungulates group, the phylogenetic analysis placed the odd-toed ungulates clade (horse and rhinoceros) as a sister of the carnivores clade (cat and seal), and aligned the even-toed ungulates clade (cow) with the cetacea clade (fin whale and blue whale). This finding is supported by some of recent analyses such as by (Nishihara et al., 2006) and (Murphy et al., 2001b). The relationship within the primates group is also consistent with most of the morphological and molecular analyses (Cao et al., 1998).

The relationship of the three placental groups presented in this study, namely primates, rodents and ferungulates groups is still a controversial issue in phylogenetics. Early analyses on several protein coding genes in the nuclear genomes (Bulmer et al., 1991; Easteal, 1990) and on the mitochondrial genomes (Janke et al., 1994; Kuma and Miyata, 1994; Cao et al., 1994a) suggest that, rodents diverged earlier than the split between primates and ferungulates. Some recent analyses (Murphy et al., 2001a; Scally et al., 2001), however, place rodents group as a sister to primates group. Though this hypothesis is today commonly considered as the consensus (Springer et al., 2004), it is often challenged

by studies, specially those on mitochondrial genomes (Reyes et al., 2000; Otu and Sayood, 2003). An analysis by Lin et al. (2002) produces a phylogeny that supports the latter hypothesis, but when additional data for outgroup are added into the data set, the tree obtained supports the formal hypothesis. This controversy is explained by Lin et al. (2002) that the mitochondrial genomes of some rodents species such as mouse and rat evolve at a much higher rate than other mammals do. Therefore, unrooted placental trees from mitochondrial DNA are often consistent with trees from nuclear DNA, while rooted trees of mitochondrial DNA differ from that of nuclear data. The phylogeny of mammals produced by XMDistance, though is different from the consensus tree, is consistent with most other analyses of mitochondrial DNA.

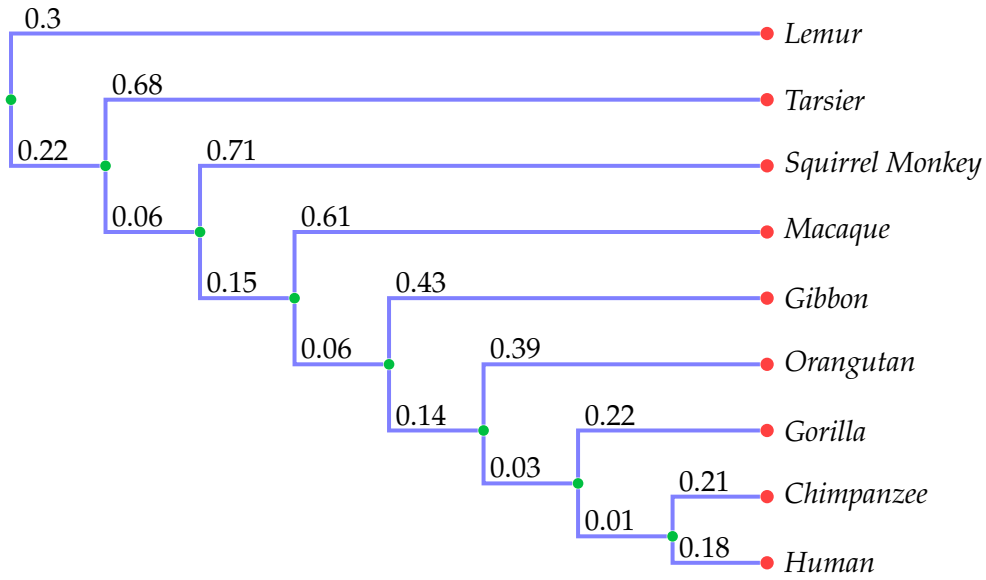


Figure 5.5: The phylogeny of primates inferred by XMDistance. The number in a branch represents the relative length of the branch.

The XMDistance method was also applied to another set of mitochondrial DNA of nine primate species. The data set contains segments of mitochondrial genomes of human, chimpanzee, gorilla, orangutan, gibbon, crab-eating macaque, squirrel monkey, tarsier and lemur. The sequences were reliably aligned by eye Yang (1994). Sites involving deletions and insertions were excluded. The length of each sequence is 888 bases. Again, the neighbour joining was used to generate a phylogeny from the distances computed by XMDistance. The phylogeny is shown in Figure 5.5. The relationship among primate species inferred in this phylogeny and in the phylogeny in Figure 5.4 is consistent with most other primate evolutionary analyses (Hayasaka et al., 1988; Yang, 1994; Yang and Rannala, 1997).

5.6 A Side Application: Phylogenetics Analysis of Genomes

It is well known that phylogenetic analyses of species based on different genes or parts of genomes are often inconsistent. Some parts of a genome may have arisen through some means other than inheritance, for example by viral insertion, DNA transformation, symbiosis or some other forms of horizontal transfer. Such mechanisms contradict the major assumption of phylogenetic analysis and thus can mislead efforts to infer evolutionary relationships among species. Furthermore, it can be argued that a single gene hardly possesses enough evolutionary information as some genes may have evolved faster than others (Gogarten and Townsend, 2005). Due to the variation of evolutionary rates among genes, phylogenetic analysis using different genes may result in different trees (Lerat et al., 2003). The availability of more and more sequenced genomes allows phylogeny construction from complete genomes, which is less sensitive to such inconsistency because all information is used rather than a subset.

However, performing global alignment of whole genomes is often impossible due to many factors such as genome rearrangement and DNA transposition. Furthermore, due to their large sizes, genomes cannot be reliably and practically aligned. For such long sequences, the use of phylogenetic tree construction methods like *maximum parsimony* and *maximum likelihood* is often impractical due to their intensive computational requirements. Distance methods, such as the *neighbour joining* method (Saitou and Nei, 1987) and the *UPGMA* method(Sokal and Michener, 1958), are better suited to data sets that have a large number of sequences. The distance methods require a measure of distances between any two genomes. While traditional genetic distance measures based on alignment are not applicable, it is important to develop alignment-free distance measures to infer phylogenies from whole genomes.

Early approaches to genome phylogenetics rely on the identification of homologues to measure genetic distances. Work by Sankoff et al. (1992) proposes using the number of events needed to rearrange genes in genomes as a measure of genetic dissimilarity. Gene content is considered in (Snel et al., 1999) to measure genome distances. The similarity of two genomes is defined as the number of genes they share. These measures are also computationally expensive or do not perform well when the gene content of the organisms are similar.

Recent years have seen an increasing number of alignment-free methods for sequence analysis. These methods are broadly categorised into two main groups, namely word based and information based (Vinga and Almeida, 2003). Those in the former group map a sequence to a vector defined by the counts of each *k-mer*, and measure genome distances by some linear algebraic and statistical measures such as the Euclidean distance (Blaisdell, 1986) or covariance distance (Gentleman and Mullin, 1989). These methods are still loosely dependent on local alignment as the comparisons are made for fixed word

length. Furthermore, these methods can easily be misled as DNA homologues contain many mutations and indels, and certain genomes have skewed composition distributions.

The second group of alignment-free algorithms are founded on information theory (Shannon, 1948) and Kolmogorov complexity theory. The advantages of these methods are that they are more elegant and do not rely on an evolutionary model. These methods are based on the premise that two related sequences would share some information and thus the amount of information in two sequences together would be less than the sum of the amount of information of each sequence. Information content can be estimated by using a lossless compression algorithm. The better the compression algorithm performs, the closer it can estimate information content of sequences.

Nevertheless, compression of biological sequences, especially long sequences, is very challenging. General text compression algorithms such as Lempel-Ziv (Ziv and Lempel, 1977) and PPM (Cleary and Witten, 1984b) typically fail to compress genomes better than the 2-bits per symbol baseline. A number of biological sequence compression algorithms such as *BioCompress* (Grumbach and Tahi, 1994) and *GenCompress* (Chen et al., 2000) have been developed during the last decade but most of them are too computationally expensive to be applied to sequences in size of over a million bases. The GenCompress algorithm, which is used to measure sequence distances in (Li et al., 2001), is reported to take about one day to compress the human chromosome 22 of 34 million bases and achieves just 12% compression. In (Otu and Sayood, 2003), an information measure is developed based on Lempel-Ziv complexity (Lempel and Ziv, 1976), which relates the number of steps in the production process of a sequence to its complexity. How well the method estimates the complexity is in fact not reported.

None of the existing tree building methods appear to be sufficiently robust to perform phylogenetic analysis on genome-size sequences. The information theoretic approaches scale well on a large data set, but the existing underlying compression algorithms are either too computationally expensive or do not perform well. The compression scheme used in Section 5.4 assumes the sequences have already been aligned and thus cannot be used for genomes. To fill this gap, this section proposes using the expert model compression algorithm, introduced in Chapter 3, for estimating the information content of sequences. The expert model provides many interesting features for the task. Firstly, it has been shown to be superior to other compression algorithms in terms of both compression performance and speed. As a rough comparison against GenCompress, the expert model running on a desktop computer can compress the whole human genome of nearly 3 billion bases in about one day and saves about 20%. Since lossless compression provides an upper bound on the entropy, better compression gives a better approximation to the information content of sequences. Secondly, the expert model can be used to estimate the information content of a sequence as well as to estimate the conditional information content of one sequence given another. Finally, the expert model runs very quickly on long

sequences, and can be used to analyse genome size sequences in practice. The genetic distance between two sequences is measured as in Equation 5.37.

5.6.1 Experimental Results

This subsection presents experiments on the proposed distance measure using two sets of data. The first data set contains the genomes of eight malaria parasites and the second contains 13 bacterial genomes. XMDistance with the expert model was applied to obtain pairwise distances between each pair of genomes in a data set. The phylogenetic trees were then constructed using the *neighbour joining* (Saitou and Nei, 1987) method from the PHYLIP package (Felsenstein, 2005). The experiments were carried out on a desktop with Pentium Dual Core 2 Duo 2.33Ghz CPU and 8GB of memory, running Linux Ubuntu 8.04.

Plasmodium phylogeny

Plasmodium species are the parasites that cause malaria in many vertebrates including human. At different stages of its life-cycle, a *Plasmodium* organism interacts with a mosquito vector and a vertebrate host. In order to adapt to the environment in the host blood, certain *Plasmodium* genes are under more evolutionary pressure than others, which leads to variation in the evolutionary rates among genes. *Plasmodium* species generally co-evolve with their hosts, and hence their evolution depends largely on hosts and geographic distribution. Certain species are thought to have emerged as a result of host switches. For example, the human malaria parasite *Plasmodium falciparum* is speculated to have diverged from the chimpanzee malaria parasite *Plasmodium reichenowi* recently, and thus be more closely related to *Plasmodium reichenowi* than to other human malaria parasites (Rich et al., 2009).

As a result, the study of malaria phylogenetics faces the difficulty of selecting genes or rRNA for analysis. Small subunit rRNA and circumsporozoite proteins have been used in many *Plasmodium* phylogenetic analyses (Waters et al., 1993; Escalante et al., 1997). However, a recent study indicates that these loci are not appropriate for evolutionary studies because *Plasmodium* species possess separate genes, each expresses at a different point in the life cycle(Corredor and Enea, 1993). Likewise, the circumsporozoite protein may be problematic as the gene codes for a surface protein is under strong selective pressure from the vertebrate immune system (Hughes and Hughes, 1995). Indeed, recent phylogeny analyses (Leclerc et al., 2004) using these molecules show results that are inconsistent with those of other loci.

The XMDistance measure was applied to construct the phylogenetic tree of eight malaria parasites, namely *P. berghei*, *P. yoelii*, *P. chabaudi* (rodent malaria), *P. falciparum*, *P. vivax*, *P. knowlesi*, *P. reichenowi* (primate malaria) and *P. gallinaceum* (bird malaria). Their

genomes were obtained from PlasmoDB release 5.5 (PlasmoDB, 2009a). The genome of *P. reichenowi* has not been completed, only 7.8 megabases out of the estimated 25 megabases are available. The genomes of *P. berghei*, *P. chabaudi*, *P. gallinaceum* and *P. vivax* have been completely sequenced, but have not been fully assembled; each genome consists of several thousand contigs. Only the genomes of three species, *P. falciparum*, *P. knowlesi* and *P. yeolii*, have been completely assembled into 14 chromosomes each. Prior to performing analysis, wildcards from the sequences were removed. The characteristics of the genomes are presented in Table 5.5.

Table 5.5: *Plasmodium* genomes characteristics.

Species	Host - Geographic Dist.	Size	%AT	Status
P. berghei	Rodent - Africa	18.0 Mb	76.27%	Partly Assembled
P. chabaudi	Rodent - Africa	16.9 Mb	75.66%	Partly Assembled
P. falciparum	Human - Subtropical	23.3 Mb	80.64%	Fully Assembled
P. gallinaceum	Bird - Southeast Asia	16.9 Mb	79.37%	Partly Assembled
P. knowlesi	Macaque - Southeast Asia	22.7 Mb	61.17%	Fully Assembled
P. reichenowi	Chimpanzee - Africa	7.4 Mb	77.81%	Partly Available
P. vivax	Human - Subtropical	27.0 Mb	57.72%	Partly Assembled
P. yoelii	Rodent- Africa	20.2 Mb	77.38%	Fully Assembled

Statistical analysis of these *Plasmodium* genomes is very challenging. The composition distributions of these genomes are greatly different. The AT content of the *P. falciparum* genome is as high as 80%, whereas the distribution for *P. vivax* is more uniform even though both species are human parasites. Conventional analysis tools would be misled by such statistical bias (Cao et al., 2009b). Because many of the genomes have not been fully assembled, methods taking advantage of gene order or genome rearrangement such as (Blaisdell, 1986; Gentleman and Mullin, 1989) cannot be used. Finally, due to the size of the data set, it is not practical to use methods such as in (Li et al., 2001; Otu and Sayood, 2003).

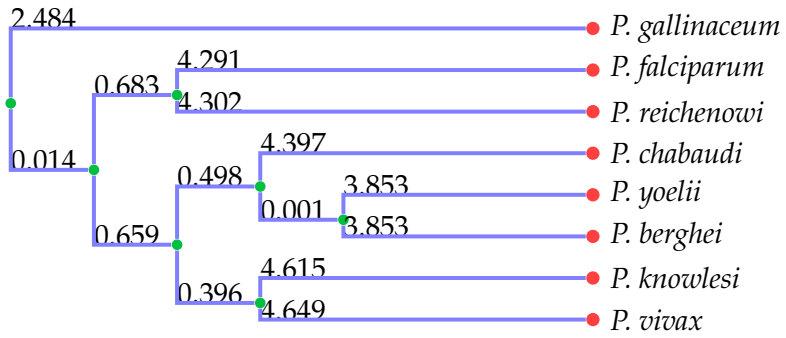


Figure 5.6: The inferred phylogenetic tree of the *Plasmodium* genus. The number in a branch represents the relative length of the branch.

It took just under 8 hours to process the 150 megabase data set and generate a pairwise distance matrix of the genomes using the expert model. The neighbour joining method was then applied to produce an unrooted tree. To make the tree rooted, *P. gallinaceum* was taken as the outgroup because *P. gallinaceum* is bird malaria, whereas the others are mammal parasites. The tree produced is shown in Figure 5.6. The tree is consistent with most earlier works (Siddall and Barta, 1992; Leclerc et al., 2004). In particular, it supports the speculation that the species closest to the human malaria parasite *P. falciparum* is in fact the chimpanzee malaria parasite *P. reichenowi* (Rich et al., 2009).

Bacteria phylogeny

Horizontal gene transfer is found extensively in bacterial genomes. This prevents the establishment of organism relationships based on individual gene (Lerat et al., 2003). In order to perform phylogenetic analysis of such species, typically a number of likely gene orthologs are selected. Phylogenetic hypotheses based on these loci are often inconsistent with each other.

XMDistance was used to perform a whole-genome phylogenetic analysis on the γ-Proteobacteria group for which horizontal gene transfer is frequently documented. The data set for the analysis contains the genomes of 13 species, namely *Escherichia coli* K12 (Genbank accession number NC_000913), *Buchnera aphidicola* APS (NC_002528), *Haemophilus influenzae* Rd (NC_000907), *Pasteurella multocida* Pm70 (NC_002663), *Salmonella typhimurium* LT2 (NC_003197), *Yersinia pestis* CO_92 (NC_003143), *Yersinia pestis* KIM5 P12 (NC_004088), *Vibrio cholerae* (NC_002505 and NC_002506), *Xanthomonas axonopodis* pv. citri 306 (NC_003919), *Xanthomonas campestris* (NC_003902), *Xylella fastidiosa* 9a5c (NC_002488), *Pseudomonas aeruginosa* PA01 (NC_002516), and *Wigglesworthia glossinidia brevipalpis* (NC_004344). The sizes of the genomes range from 1.8 megabases to about 7 megabases, and the total size of the data set is 44 megabases.

An earlier phylogenetic analysis of the 13 species (Lerat et al., 2003) found inconsistency among evolutionary trees constructed from different genes. There are 14,158 gene families found in these genomes. The majority of these families contain only one gene. Only 275 families are represented in all 13 species, and 205 families contain exactly one gene per species. The analysis used the alignments of these 205 families and generated 13 different topologies by various tree construction methods. The likelihood tests of the 13 topologies reported that the four most probable topologies are in agreement with over 180 gene families and that the consensus topology is in agreement with 203 alignments. These four trees differ in regard to the positions of three species, *Wigglesworthia*, *Buchnera* and *Vibrio*.

The tree inferred by the XMDistance method is presented in Figure 5.7. Except for the three contentious species, the tree agrees with the four most likely topologies. Similar

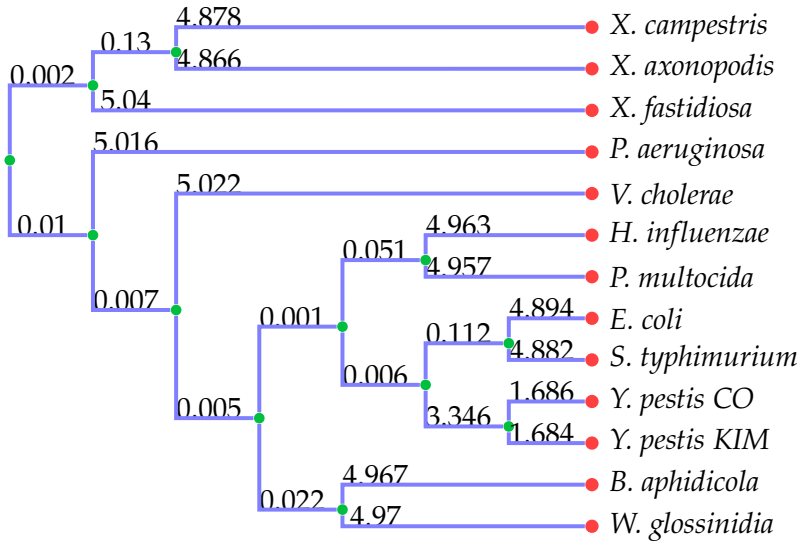


Figure 5.7: The inferred phylogenetic tree of the γ-Proteobacteria group. The number in a branch represents the relative length of the branch.

to the consensus tree, it also supports the hypothesis that *Wigglesworthia* and *Buchnera* are sister species. It only differs from the consensus tree in the positions of the branches involving (*Buchnera*, *Wigglesworthia*), and (*Haemophilus*, *Pasteurella*). A close examination of the tree shows that, the distances from these groups to their parent, and the distance between the most recent ancestor of *Vibrio* to its parent, are very small. This suggests that, these species split from each other at very similar times. This explains the inconsistency among the four most probable trees generated by (Lerat et al., 2003) and the tree inferred by the XMDistance approach.

5.7 Summary

This chapter has presented an information theoretic approach to measuring genetic distances between sequences for phylogenetic analysis. The distance measure is based on compression to estimate the information content of the sequences and uses the information content to calculate genetic distances between species. Appropriate compression techniques are used depending on whether the sequences have been aligned or not. Unlike conventional phylogenetic methods, the new method does not rely on an evolutionary model. Furthermore, the method is able to handle data with considerable biases in genetic composition, which classical statistical analysis approaches often fail to deal with.

On a set of simulated data where the alignment of sequences was known, the new method was found to perform comparably with the standard phylogenetic analysis methods in both inferring the correct tree topologies and estimating branch lengths. In particular, in comparison with the maximum likelihood method, the maximum parsimony method and the standard distance method which estimates genetic distances based on the mutation patterns, the proposed method was only outperformed by the maximum likelihood method, which however required much longer time to run. The new method was also found to infer plausible trees of species from real data sequences.

The method was used to generate phylogenetic trees from the whole genomes of eight *Plasmodium* species, and from 13 species of the γ-Proteobacteria group. The genomes in both data sets are known to contain abundant horizontally transferred genes. Previous analysis of these species using small molecules showed inconsistencies among the trees based on different genes. The trees generated by XMDistance are largely consistent with the consensus trees from previous work.

To the best of the author's knowledge, the approach is the first to be able to infer reliable phylogenetic trees from whole genomes of eukaryote species, with minimal requirement of computation power. Such a tool would be very useful for knowledge discovery from the exponentially increasing databases of genomes resulting from the latest sequencing technologies. As information is the universal measure, the distance measure presented in this chapter can be extended for analysis of other types of data. Potential applications include reconstruction of language history (Mace and Holden, 2005) and analysis of the evolution of computer viruses.

Chapter 6

Conclusion

This thesis has investigated the use of information content for biological sequence analyses. An algorithm for compression of long biological sequences has been developed in the research for the purpose. The algorithm has been applied successfully to pattern discovery, sequence alignment and phylogenetic analysis.

6.1 The Expert Model

Chapter 3 presented the expert model (XM), a novel algorithm for compression of biological sequences. The algorithm was developed by modelling redundancy features in biological sequences such as approximate repeats and long range similarity. Unlike previous biological sequence compression algorithms, the expert model is an adaptive compression algorithm. It employs Bayesian averaging to blend predictions from different contexts: Markov models of varying orders, and repeats from different ranges. The framework for the blending of predictions is extendable; it can be applied to combinations of different predictive models, and can be applied to other types of data.

The expert model is shown to outperform existing biological sequence compression algorithms on standard benchmarks, both for DNA and protein. It is also faster than most existing algorithms. Importantly, the expert model is capable of compressing long sequences such as eukaryotic genomes. It is the first biological sequence compression algorithm reported to be able to compress sequences in lengths of up to a billion bases on a desktop computer. It therefore provides an adequate tool for analysing the genomes of most organisms on the planet. Chapter 3 also presented a compressibility study of the genomes of various species from various organism levels using the expert model.

The expert model can make use of different contexts (or sources of background knowledge) for compression of a sequence, and can estimate the information content of every symbol in the sequence with respect to a context. This is useful for many sequence

analysis tasks. Chapter 3 showed an application of the expert model to detection of repeat patterns. Other applications of the expert model to sequence analysis were presented in chapters 4 and 5.

6.2 Sequence Alignment

Local alignment refers to the identification of regions of local similarity between sequences. Chapter 4 described XMAligner, a novel method to perform local alignment of two genomic sequences. The method is a departure from the conventional character-based matching approaches. It performs sequence alignment at the information level. The method makes use of compression using the expert model (Cao et al., 2007, 2010a) presented in chapter 3 to measure the information content of each sequence, and the conditional information content of a sequence on the background of the other. By examining the information content and the conditional information content, one can identify pairs of similar regions in the two sequences. Here, XMAligner considers two regions to be *similar* if they share some information, that is if the compression of a region on the background knowledge of the other is significantly better than the compression without the background knowledge. Such a pair of similar regions is called a *High-scoring Segment Pair* (HSP).

The chapter showed that the mutual information content of two regions in an HSP is in fact the traditional alignment score of the HSP. However, in contrast to existing alignment methods which often rely on a predetermined scoring scheme, XMAligner estimates the information content of each region in an adaptive manner. As a result, XMAligner can handle areas of statistical bias since the information content in these areas is lower than that in more uniformly distributed regions. This is an advantage over existing alignment methods which often have to "mask out" low information content regions before performing alignment (Wootton and Federhen, 1993; Wootton, 1997).

XMAligner is based on the premise that the best alignment of two sequences leads to the best compression of the two sequences together. Compressibility provides a natural objective function for parameter estimation, which is often lacking in other alignment methods. The objective function is important in obtaining an optimal alignment while not having to rely on some evolutionary assumptions. It is also useful in the computation of a substitution matrix between the genomes of any two species. The substitution matrix is considered as parameters of the alignment, and an expectation maximisation approach is used to obtain the matrix that gives the best compression.

Experiments on simulated data showed that XMAligner outperforms most other existing alignment methods, especially when the data is statistically biased and the sequences are distantly related. XMAligner was used for exon detection on the Jareborg

data set (Jareborg et al., 1999) which contains annotated pairs of genomic sequences from the human and mouse genomes. In comparison with several most common existing methods, XMAligner was the most sensitive, though some other methods such as Promer (Kurtz et al., 2004) and DIALIGN (Morgenstern, 1999) have built in mechanisms specifically to detect exons and to help guide the alignment. XMAligner was also applied to alignment of the genomes of several *Plasmodium* species, each of which is about 20 megabases long. It was compared to Nucmer and Promer from the MUMmer package (Kurtz et al., 2004), which are the only two available tools able to align such long sequences. XMAligner was found to be superior to Nucmer which also detects exons from the comparison of nucleotide sequence. Promer was designed specifically for exon detection since it translates potential exons to proteins and performs sequence comparison on proteins. Despite not using any mechanism for detecting protein coding regions, XMAligner's performance was at least as good as Promer in general and better on statistically biased data and on distantly related sequences.

The output of XMAligner was integrated into InfoV package (Dix et al., 2007) for visualisation of the alignment. Annotations of the sequences can also be imported into the visualisation package for comparison with the alignment. The software suite including XMAligner and the visualisation program is available on the project website (Cao et al., 2010b).

6.3 Phylogenetic Analysis

Chapter 5 presented an application of compression to phylogenetic analysis. In particular, a measure of genetic distance, *XMDistance*, between any two sequences was derived from the estimated information content of each sequence, and the estimated conditional information content of each sequence given the other. The distance measure can be used to generate a matrix of pairwise distances for a group of sequences, each of which represents a species. The distance matrix can then be used to infer a phylogenetic tree, which is a hypothesis about the evolutionary history of these species.

The chapter showed that XMDistance is approximately proportional to the elapsed time from when two species split, assuming the constant evolutionary rate. The computation of the genetic distance does not rely on an evolutionary model as in most existing phylogenetic analysis methods. XMDistance does not *require* a multiple alignment of the sequences. However, this approach can make use of an alignment if it is available. Specifically, if the sequences have been aligned, a simple compression scheme based on a Markov model is used to estimate the respective information content. In the case that the sequences cannot be reliably aligned, the expert model compression algorithm of chapter

3 is used. The method can therefore be applied to infer phylogenetic trees from whole genomes, which contain virtually all the genetic information of the species.

XMDistance was experimentally compared to several standard phylogenetic tree building methods namely maximum likelihood (Felsenstein, 1981), maximum parsimony (Camin and Sokal, 1965), and the standard distance measure. These methods were taken from the PHYLIP package (Felsenstein, 2005). Experiments on a set of simulated data, in which the correct alignment can be obtained and the true phylogenetic trees are known, showed that the XMDistance method outperformed the maximum parsimony and standard distance methods, and was only behind the maximum likelihood method. The XMDistance method however, ran much faster than these three methods especially when the number of sequences was large. The XMDistance approach was also applied to a set of mitochondria genomes of 20 mammal species, and a set of nine primates mitochondria genes. In both cases, the phylogenetic trees inferred by XMDistance were plausible.

XMDistance was applied to phylogenetic analyses of two problematic data sets of whole genomes. The first data set contains the genomes of eight *Plasmodium* species with the total size of 150 megabases. These genomes are well known for the differences in CG content, and the variation of evolutionary rates among their genes. The phylogenetic tree inferred by XMDistance in this case was found to be consistent with most recent works. The second data set consists of the genomes of 13 bacteria in the γ-Proteobacteria group with which the problem is the abundance of horizontal gene transfer. The total size of the data set is 44 megabases and again the phylogenetic tree generated by XMDistance was largely in agreement with the consensus trees.

6.4 Closing Remarks

Biological sequence compression is important, not only for saving storage space and reducing communication bandwidth, but also for many information extraction tasks. Data analyses based on compression have been shown to overcome the problems from statistically biased data. The estimated information content of each base in a sequence is useful for studying many aspects of biology and facilitates many information extraction tasks.

Compression gives one way to access the information content of biological sequences. This research has developed the expert model, an effective and fast algorithm for the purpose. The expert model is shown to outperform existing biological sequence compression algorithms on the standard benchmarks. The expert model offers many attractive features for sequence analysis such as it can produce the approximated per element information content of a biological sequence, and can perform compression using different contexts. Another advantage over the other compression algorithms is that the expert model can

handle long sequences such as whole eukaryotic genomes with modest hardware requirement. The expert model is fully adaptive. It also provides a novel mechanism for blending predictive models, which can be extended to other types of data.

This thesis argues that since biological sequences are the carriers of genetic information in the biological processing system, sequence analyses at the information level offer many advantages over the conventional character-based analyses. It demonstrates that biological sequence analyses based on information theory can overcome many difficulties such as the statistical bias in data. Information theoretic approaches have been successfully applied to two of the most important sequence analyses, namely sequence alignment and phylogenetics, on whole genomes. In particular, these approaches can deal with problematic data such as statistically biased data and distantly related sequences.

Bibliography

Abouelhoda, M. I., Kurtz, S. and Ohlebusch, E., 2006. Enhanced suffix arrays and applications. In Aluru, S., ed., *Handbook on Computational Molecular Biology*, pp. 7-1 – 7-27. Chapman and Hall/CRC.

Adjeroh, D. and Nan, F., 2006. On compressibility of protein sequences. *Data Compression Conference*, pp. 422–434. doi:10.1109/dcc.2006.56.

Adjeroh, D., Zhang, Y., Mukherjee, A., Powell, M. and Bell, T., 2002. DNA sequence compression using the Burrows-Wheeler transform. In *IEEE Computer Society Conference on Bioinformatics*, p. 303. IEEE Computer Society, Washington, DC, USA.

Allison, L., 2005. Models for machine learning and data mining in functional programming. *Journal of Functional Programming*, 15(01):15–32. doi:10.1017/s0956796804005301.

Allison, L., Edgoose, T. and Dix, T. I., 1998. Compression of strings with approximate repeats. *Intelligent Systems in Molecular Biology (ISMB98), Montreal*, pp. 8–16.

Allison, L., Powell, D. and Dix, T. I., 1999. Compression and approximate matching. *Computer Journal*, 42(1):1–10.

Allison, L. and Wallace, C. S., 1994. The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and to optimization of multiple alignments. *Journal of Molecular Evolution*, 39(4):418–430. doi:10.1007/bf00160274.

Allison, L., Wallace, C. S. and Yee, C. N., 1992. Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution*, 35(1):77–89. doi:10.1007/bf00160262.

Allison, L. and Yee, C. N., 1990. Minimum message length encoding and the comparison of macromolecules. *Bulletin of Mathematical Biology*, 52(3):431–452.

Altschul, S. F., 1991. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219(3):555–565.

Altschul, S. F., Gish, W., Miller, W., Myers, E. and Lipman, D., 1990. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410. doi:10.1006/jmbi.1990.9999.

Altschul, S. F., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D., 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402. doi:10.1093/nar/25.17.3389.

Apostolico, A. and Lonardi, S., 2000. Compression of biological sequences by greedy off-line textual substitution. In Storer, J. A. and Cohn, M., eds., *ProceedingsData Compression Conference*, pp. 143–152. IEEE Computer Society Press, Snowbird, UT.

Arnauld, A. and Baynes, T. S., 1962. *Logic; or, The art of Thinking: being the Port Royal Logic*. Edinburgh.

Arumuganathan, K. and Earle, E., 1991. Nuclear DNA content of some important plant species. *Plant Molecular Biology Reporter*, 9(3):208–218. doi:10.1007/bf02672069.

Baase, W. A., Jose, D., Ponedel, B. C., von Hippel, P. H. and Johnson, N. P., 2009. DNA models of trinucleotide frameshift deletions: the formation of loops and bulges at the primer-template junction. *Nucl. Acids Res.*, 37(5):1682–1689. doi:10.1093/nar/gkn1042.

Baldauf, S. L. and Palmer, J. D., 1993. Animals and fungi are each other's closest relatives: congruent evidence from multiple proteins. *Proceedings of the National Academy of Sciences*, 90(24):11558–11562.

Baxter, R. A. and Oliver, J. J., 1994. MDL and MML: similarities and differences. Technical Report TR 94/207, Department of Computer Science, Monash University.

Bayes, T., 1763. An essay towards solving a problem in the doctrine of chance. *Philosophical Transactions of the Royal Society of London*, 53:370–418. Reprinted in Biometrika, 45, pp.296-315, 1958.

Behzadi, B. and Fessant, F. L., 2005. DNA compression challenge revisited: A dynamic programming approach. In *Combinatorial Pattern Matching*, pp. 190–200. doi:10.1007/b137128.

Bell, T., Witten, I. H. and Cleary, J. G., 1989. Modeling for text compression. *ACM Comput. Surv.*, 21(4):557–591. doi:10.1145/76894.76896.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J. and Sayers, E. W., 2009. GenBank. *Nucleic Acids Research*, 37(suppl_1):D26–31. doi:10.1093/nar/gkn723.

Berg, D. E. and Howe, M. M., eds., 1989. *Mobile DNA*. American Society for Microbiology, Washington, D.C.

Blaisdell, B. E., 1986. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*, 83(14):5155–5159.

Bondi, A., 1964. Van der Waals volumes and radii. *Journal of Physical Chemistry*, 68(3):441–451. doi:10.1021/j100785a001.

Boulton, D. M. and Wallace, C. S., 1969. The information content of a multistate distribution. *Journal of Theoretical Biology*, 23(2):269–278.

Boulton, D. M. and Wallace, C. S., 1970. A program for numerical classification. *Computer Journal*, 13(1):63–69.

Bray, N., Dubchak, I. and Pachter, L., 2003. Avid: A global alignment program. *Genome Research*, 13(1):97–102. doi:10.1101/gr.789803.

Brudno, M., Chapman, M., Gottgens, B., Batzoglou, S. and Morgenstern, B., 2003. Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinformatics*, 4(1):66. doi:10.1186/1471-2105-4-66.

Buard, J. and Jeffreys, A. J., 1997. Big, bad minisatellites. *Nature Genetics*, 15(4):327–328. doi:10.1038/ng0497-327.

Bulmer, M., Wolfe, K. H. and Sharp, P. M., 1991. Synonymous nucleotide substitution rates in mammalian genes: implications for the molecular clock and the relationship of mammalian orders. *Proceedings of the National Academy of Sciences*, 88(14):5974–5978.

Buneman, P., 1971. The recovery of trees from measures of dissimilarity. In *Mathematics in the Archaeological and Historical Sciences*, p. 387–395. Edinburgh University Press.

Burrows, M. and Wheeler, D. J., 1994. A block-sorting lossless data compression algorithm. Technical Report SRC-RR-124, Digital Equipment Corporation, Palo Alto, California.

Burset, M. and Guigó, R., 1996. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367. doi:10.1006/geno.1996.0298.

Burstein, D., Ulitsky, I., Tuller, T. and Chor, B., 2005. Information theoretic approaches to whole genome phylogenies. In *RECOMB*, pp. 283–295. doi:10.1007/11415770_22.

Camin, J. and Sokal, R., 1965. A method for deducing branching sequences in phylogeny. *Evolution*, 19:311–326.

Cao, M. D., Allison, L. and Dix, T. I., 2009a. A distance measure for genome phylogenetic analysis. In *Australian Conference on Artificial Intelligence*, pp. 71–80.

Cao, M. D., Dix, T. I. and Allison, L., 2009b. Computing substitution matrices for genomic comparative analysis. In *PAKDD 2009, LNAI 5476*, pp. 647–655. doi:10.1007/978-3-642-01307-2_64.

Cao, M. D., Dix, T. I. and Allison, L., 2009c. A genome alignment algorithm based on compression. Technical Report 2009/233, Faculty of Information Technology, Monash University, Victoria, Australia.

Cao, M. D., Dix, T. I. and Allison, L., 2010a. A biological compression model and its applications. In *Software Tools and Algorithms for Biological Systems*. Springer. In press.

Cao, M. D., Dix, T. I. and Allison, L., 2010b. Expert model software suite. URL ftp://ftp.infotech.monash.edu.au/software/DNAcompress-XM/.

Cao, M. D., Dix, T. I. and Allison, L., 2010c. A genome alignment algorithm based on compression. *BMC Bioinformatics*. Submitted.

Cao, M. D., Dix, T. I., Allison, L. and Mears, C., 2007. A simple statistical algorithm for biological sequence compression. *Data Compression Conference*, pp. 43–52. doi:10.1109/dcc.2007.7.

Cao, Y., Adachi, J., Janke, A., Paabo, S. and Hasegawa, M., 1994a. Phylogenetic relationships among eutherian orders estimated from inferred sequences of mitochondrial proteins: Instability of a tree based on a single gene. *Journal of Molecular Evolution*, 39(5):519–527. doi:10.1007/bf00173421.

Cao, Y., Adachi, J., Yano, T. and Hasegawa, M., 1994b. Phylogenetic place of guinea pigs: no support of the rodent-polyphyly hypothesis from maximum-likelihood analyses of multiple protein sequences. *Molecular Biology and Evolution*, 11(4):593–604.

Cao, Y., Janke, A., Waddell, P. J., Westerman, M., Takenaka, O., Murata, S., Okada, N., Paabo, S. and Hasegawa, M., 1998. Conflict among individual mitochondrial proteins in resolving the phylogeny of eutherian orders. *Journal of Molecular Evolution*, 47(3):307–322. doi:10.1007/pl00006389.

Carter, R., 2003. Speculations on the origins of plasmodium vivax malaria. *Trends in Parasitology*, 19(5):214 – 219. doi:10.1016/s1471-4922(03)00070-9.

Cavalli-Sforza, L. L. and Edwards, A. W. F., 1967. Phylogenetic analysis. models and estimation procedures. *American Journal of Human Genetics*, 19(3 Pt 1):233–257.

Chaitin, G. J., 1966. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13(4):547–569. doi:10.1145/321356.321363.

Chen, X., Kwong, S. and Li, M., 2000. A compression algorithm for DNA sequences and its applications in genome comparison. In *RECOMB*, p. 107.

Chen, X., Li, M., Ma, B. and Tromp, J., 2002. DNACompress: Fast and effective DNA sequence compression. *Bioinformatics*, 18(2):1696–1698.

Cleary, J. G. and Witten, I. H., 1984a. A comparison of enumerative and adaptive codes. *IEEE Transactions on Communications*, IT-30(2):306–315.

Cleary, J. G. and Witten, I. H., 1984b. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, COM-32(4):396–402.

Comeron, J. M. and Aguade, M., 1998. An evaluation of measures of synonymous codon usage bias. *Journal of Molecular Evolution*, 47(3):268–274. doi:10.1007/pl00006384.

Corredor, V. and Enea, V., 1993. Plasmodial ribosomal RNA as phylogenetic probe: a cautionary note. *Molecular Biology and Evolution*, 10(4):924–926.

Cover, T. M. and Thomas, J. A., 1991. *Elements of Information Theory*. John Wiley & Sons. Inc.

Crick, F. H. C., 1958. On protein synthesis. *Symposia of the Society for Experimental Biology*, 12:138–163.

Crick, F. H. C., 1970. Central dogma of molecular biology. *Nature*, 227(5258):561–563.

Darwin, C., 1859. *On the Origin of Species*. John Murray.

Das, A., Sharma, M., Gupta, B. and Dash, A., 2009. Plasmodium falciparum and Plasmodium vivax: so similar, yet very different. *Parasitology Research*, 105(4):1169–1171. doi:10.1007/s00436-009-1521-y.

Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C., 1978. *A Model for Evolutionary Change in Proteins*, volume 5. National Biochemical Research Foundation, Washington DC.

Deininger, P. L., Batzer, M. A., Hutchison, C. A. and Edgell, M. H., 1992. Master genes in mammalian repetitive DNA amplification. *Trends in Genetics*, 8(9):307 – 311. doi:10.1016/0168-9525(92)90262-3.

Delcher, A. L., Phillippy, A., Carlton, J. M. and Salzberg, S. L., 2002. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research*, 30(11):2478–2483. doi:10.1093/nar/30.11.2478.

Dix, T. I., Powell, D., Allison, L., Bernal, J., Jaeger, S. and Stern, L., 2007. Comparative analysis of long DNA sequences by per element information content using different contexts. *BMC Bioinformatics*, 8(Suppl 2):S10. doi:10.1186/1471-2105-8-s2-s10.

Dobzhansky, T., 1973. Nothing in biology makes sense except in the light of evolution. *The American Biology Teacher*, 35:125–129.

Dowe, D. L., Allison, L., Dix, T. I., Hunter, L., Wallace, C. S. and Edgoose, T., 1996. Circular clustering of protein dihedral angles by minimum message length. *Pacific Symposium on Biocomputing*, pp. 242–255.

Dunham, I., Hunt, A. R., Collins, J. E., Bruskiewich, R., Beare, D. M., Clamp, M., Smink, L. J., Ainscough, R., Almeida, J. P., Babbage, A. et al., 1999. The DNA sequence of human chromosome 22. *Nature*, 402(6761):489–495. doi:10.1038/990031.

Easteal, S., 1990. The pattern of mammalian evolution and the relative rate of molecular evolution. *Genetics*, 124(1):165–173.

Eck, R. V. and Dayhoff, M. O., 1966. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Silver Springs, Maryland.

Edwards, A. W. F. and Cavalli-Sforza, L. L., 1963. The reconstruction of evolution. *Heredity*, 18:533.

Escalante, A., Goldman, I. F., Rijk, P. D., Wachter, R. D., Collins, W. E., Qari, S. H. and Lal, A. A., 1997. Phylogenetic study of the genus plasmodium based on the secondary structure-based alignment of the small subunit ribosomal RNA. *Molecular and Biochemical Parasitology*, 90(1):317–321.

Farris, J. S., 1969. A successive approximations approach to character weighting. *Systematic Zoology*, 18(4):374–385. doi:10.2307/2412182.

Felsenstein, J., 1978. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology*, 27(4):401–410. doi:10.1093/sysbio/27.4.401.

Felsenstein, J., 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Biology*, 76(6):368–376. doi:10.1007/BF01734359.

Felsenstein, J., 2005. PHYLIP (Phylogeny Inference Package) version 3.6. department of genome sciences, university of washington, seattle. Distributed by the author.

Felsenstein, J. and Churchill, G., 1996. A Hidden Markov Model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, 13(1):93–104.

Feng, D.-F. and Doolittle, R., 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4):351–360. doi:10.1007/bf02603120.

Ferragina, P. and Manzini, G., 2000. Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, p. 390. IEEE Computer Society, Washington, DC, USA.

Fitch, W. M., 1971. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20(4):406–416.

Fitch, W. M. and Margoliash, E., 1967. Construction of Phylogenetic Trees. *Science*, 155(3760):279–284. doi:10.1126/science.155.3760.279.

Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J.-F., Dougherty, B. A., Merrick, J. M. et al., 1995. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269(5223):496–512. doi:10.1126/science.7542800.

Florea, L., Hartzell, G., Zhang, Z., Rubin, G. M. and Miller, W., 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, 8:967–974. doi:10.1101/gr.8.9.967.

Fraenkel, A. S. and Klein, S. T., 1996. Robust universal complete codes for transmission and compression. *Discrete Applied Mathematics*, 64:31–55.

Gentleman, J. and Mullin, R., 1989. The distribution of the frequency of occurrence of nucleotide subsequences, based on their overlap capability. *Biometrics*, 45(1):35–52.

Giancarlo, R., Scaturro, D. and Utro, F., 2009. Textual data compression in computational biology: a synopsis. *Bioinformatics*, 25(13):1575–1586. doi:10.1093/bioinformatics/btp117.

Gibbs, A. J. and Mcintyre, G. A., 1970. The diagram, a method for comparing sequences. *European Journal of Biochemistry*, 16(1):1–11. doi:10.1111/j.1432-1033.1970.tb01046.x.

Gogarten, P. and Townsend, F., 2005. Horizontal gene transfer, genome innovation and evolution. *Nature Reviews Microbiology*, 3(9):679–687. doi:10.1038/nrmicro1204.

Goldman, N., 1993. Statistical tests of models of DNA substitution. *Journal of Molecular Evolution*, 36(2):182–198. doi:10.1007/bf00166252, 10.1007/bf00166252.

Goldman, N. and Yang, Z., 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular Biology and Evolution*, 11(5):725–736.

Gonnet, G. H., Cohen, M. A. and Benner, S. A., 1992. Exhaustive matching of the entire protein sequence database. *Science*, 256(5062):1443–1445. doi:10.1126/science.1604319.

Good, I. J., 1979. Studies in the history of probability and statistics. XXXVII A. M. Turing's statistical work in World War II. *Biometrika*, 66(2):393–396. doi:10.1093/biomet/66.2.393.

Gregory, R. T., 2001. Coincidence, coevolution, or causation? DNA content, cell size, and the C-value enigma. *Biological Reviews*, 76(01):65–101. doi:10.1017/s1464793100005595.

Grumbach, S. and Tahi, F., 1993. Compression of DNA sequences. In *Data Compression Conference*, pp. 340–350.

Grumbach, S. and Tahi, F., 1994. A new challenge for compression algorithms: Genetic sequences. *Journal of Information Processing and Management*, 30(6):875–866. doi:10.1016/0306-4573(94)90014-0.

Gusfield, D., Balasubramanian, K. and Naor, D., 1992. Parametric optimization of sequence alignment. In *SODA '92: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pp. 432–439. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Hamady, M., Betterton, M. and Knight, R., 2006. Using the nucleotide substitution rate matrix to detect horizontal gene transfer. *BMC Bioinformatics*, 7(1):476. doi:10.1186/1471-2105-7-476.

Hartigan, J. A., 1973. Minimum mutation fits to a given tree. *Biometrics*, 29(1):53–65.

Hartley, R. V. L., 1928. Transmission of information. *The Bell System Technical Journal*, p. 535–563.

Hasegawa, M., Kishino, H. and Yano, T., 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22(2):160–174.

Hategan, A. and Tabus, I., 2004. Protein is compressible. In *Signal Processing Symposium, 2004. NORSIG 2004*, pp. 192–195.

Hategan, A. and Tabus, I., 2005. A compression algorithm based on global sequence alignments for proteome sequence analysis. In *Proceedings of the 2nd International conference on Computational Intelligence in Medical and Healthcare, CIMED 2005, Costa da Caparica, Portugal,*, pp. 44–50.

Hayasaka, K., Gojobori, T. and Horai, S., 1988. Molecular phylogeny and evolution of primate mitochondrial DNA. *Molecular Biology and Evolution*, 5(6):626–644.

Hendy, M. D. and Penny, D., 1982. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences*, 59(2):277 – 290. doi:10.1016/0025-5564(82)90027-x.

Hendy, M. D. and Penny, D., 1989. A framework for the quantitative study of evolutionary trees. *Systematic Zoology*, 38(4):297–309. doi:10.2307/2992396.

Henikoff, S. and Henikoff, J. G., 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919.

Herzel, H., 1988. Complexity of symbol sequences. *Systems Analysis Modelling Simulation*, 5(5):435–444.

Hess, J. F., Fox, M., Schmid, C. and Shen, C.-K. J., 1983. Molecular evolution of the human adult alpha -globin-like gene region: Insertion and deletion of Alu family repeats and non-Alu DNA sequences. *Proceedings of the National Academy of Sciences*, 80(19):5970–5974. doi:10.1073/pnas.80.19.5970.

Higgins, D. G., Thompson, J. D. and Gibson, T. J., 1996. Using CLUSTAL for multiple sequence alignments. *Methods in Enzymology*, 266:383–402.

Hillier, L. W., Graves, T. A., Fulton, R. E., Fulton, L. A., Pepin, K. H., Minx, P., Wagner-McPherson, C., Layman, D., Wylie, K., Sekhon, M. et al., 2005. Generation and annotation of the dna sequences of human chromosomes 2 and 4. *Nature*, 434(7034):724–731. doi:10.1038/nature03466.

Hoeting, J. A., Madigan, D., Raftery, A. E. and Volinsky, C. T., 1999. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417.

Höhl, M., Kurtz, S. and Ohlebusch, E., 2002. Efficient multiple genome alignment. *Bioinformatics*, 18(Suppl. 1):S312–S320.

Holland, R. C. G., Down, T. A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Drager, A., Yates, A., Heuer, M. et al., 2008. BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097. doi:10.1093/bioinformatics/btn397.

Huelsenbeck, J. P., 1995. Performance of phylogenetic methods in simulation. *Systematic Biology*, 44(1):17–48. doi:10.1093/sysbio/44.1.17.

Huestis, R., 2008. Personal communication.

Huffman, D., 1952. A method for the construction of minimum-redundancy codes. *Proceedings of IRE*, 40(9):1098–1102. doi:10.1007/bf02837279.

Hughes, M. K. and Hughes, A. L., 1995. Natural selection on plasmodium surface proteins. *Molecular and Biochemical Parasitology*, 71(1):99–113.

Janke, A., Feldmaier-Fuchs, G., Thomas, W. K., von Haeseler, A. and Paabo, S., 1994. The marsupial mitochondrial genome and the evolution of placental mammals. *Genetics*, 137(1):243–256.

Jareborg, N., Birney, E. and Durbin, R., 1999. Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Research*, 9(9):815–824. doi:10.1101/gr.9.9.815.

Johnson, N. L. and Kotz, S., 1969. *Distributions in Statistics. Discrete Distributions*. Houghton Mifflin, Boston.

Jones, D. T., Taylor, W. R. and Thornton, J. M., 1992. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282. doi:10.1093/bioinformatics/8.3.275.

Jukes, T. H. and Cantor, C., 1969. Evolution of protein molecules. *Mammalian Protein Metabolism*, pp. 21–132.

Jurka, J., 2003. Repetitive DNA: Detection, annotation and analysis. In Krawetz, S. A. and Womble, D. D., eds., *Introduction to Bioinformatics, A Theoretical and Practical Approach*, chapter 8, pp. 151–167. Humana Press, Totowa, New Jersey.

Jurka, J., Kapitonov, V. V., Pavlicek, A., Klonowski, P., Kohany, O. and Walichiewicz, J., 2005. Repbase update, a database of eukaryotic repetitive elements. *Cytogentic and Genome Research*, 110:462–467. doi:10.1159/000084979.

Kärkkäinen, J., Sanders, P. and Burkhardt, S., 2006. Linear work suffix array construction. *Journal of the ACM*, 53(6):918–936. doi:10.1145/1217856.1217858.

Karlin, S. and Altschul, S. F., 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87(6):2264–2268.

Kazazian, J., Haig H., 2004. Mobile elements: Drivers of genome evolution. *Science*, 303(5664):1626–1632. doi:10.1126/science.1089670.

Keich, U., Li, M., Ma, B. and Tromp, J., 2004. On spaced seeds for similarity search. *Discrete Appl. Math.*, 138(3):253–263. doi:10.1016/s0166-218x(03)00382-2.

Kent, W. J., 2002. BLAT - the BLAST-like alignment tool. *Genome Research*, 12(4):656–664. doi:10.1101/gr.229202.

Kidd, K. K. and Sgaramella-Zonta, L. A., 1971. Phylogenetic analysis: concepts and methods. *American Journal of Human Genetics*, 23(3):235–252.

Kimura, M., 1980. A simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120. doi:10.1007/bf01731581.

Kimura, M., 1981. Estimation of evolutionary distances between homologous nucleotide sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 78(1):454–458.

Kishino, H. and Hasegawa, M., 1989. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea. *Journal of Molecular Evolution*, 29(2):170–179.

Kocher, T. D. and Wilson, A. C., 1991. Sequence evolution of mitochondrial DNA in humans and chimpanzees: control region and a protein-coding region. In Osawa, S. and Honio, T., eds., *Evolution of Life: Fossils, molecules, and culture*, pp. 391–413. Springer-Verlag, Tokyo.

Kocsor, A., Kertesz-Farkas, A., Kajan, L. and Pongor, S., 2005. Application of compression-based distance measures to protein sequence classification: a methodological study. *Bioinformatics*, p. bti806. doi:10.1093/bioinformatics/bti806.

Kolmogorov, A., 1933. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer,, Berlin.

Kolmogorov, A., 1965. Three approaches to the quantitative definition of information. *Problems of Information and Transmission*, 1(1):1–7.

Korodi, G. and Tabus, I., 2005. An efficient normalized maximum likelihood algorithm for DNA sequence compression. *ACM Transactions on Information Systems*, 23(1):3–34. doi:10.1145/1055709.1055711.

Korodi, G. and Tabus, I., 2007. Normalized maximum likelihood model of order-1 for the compression of DNA sequences. In *Proceedings of the 2007 Data Compression Conference*, pp. 33–42. IEEE Computer Society, Washington, DC, USA. doi:10.1109/dcc.2007.60.

Kuhner, M. K. and Felsenstein, J., 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates [published erratum appears in Mol Biol Evol 1995 May;12(3):525]. *Molecular Biology and Evolution*, 11(3):459–468.

Kuma, K. and Miyata, T., 1994. Mammalian phylogeny inferred from multiple protein data. *The Japanese Journal of Genetics*, 69(5):555–566.

Kumar, S., 1996. A stepwise algorithm for finding minimum evolution trees. *Molecular Biology and Evolution*, 13(4):584–593.

Kurtz, S., Phillippy, A., Delcher, A. L., Smoot, M., Shumway, M., Antonescu, C. and Salzberg, S., 2004. Versatile and open software for comparing large genomes. *Genome Biology*, 5(2). doi:10.1186/gb-2004-5-2-r12.

Kyte, J. and Doolittle, R. F., 1982. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105 – 132. doi:10.1016/0022-2836(82)90515-0.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J. and Devon, K., 2001. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921.

Laplace, P., 1814. *Essai Philosophique sur les Probabilités*. Paris.

Leclerc, M. C., Hugot, J. P., Durand, P. and Renaud, F., 2004. Evolutionary relationships between 15 Plasmodium species from new and old world primates (including humans): an 18s rDNA cladistic analysis. *Parasitology*, 129(16):677–684.

Lempel, A. and Ziv, J., 1976. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, 22(1):75–81.

Lerat, E., Daubin, V. and Moran, N. A., 2003. From gene trees to organismal phylogeny in prokaryotes:the case of the gamma-proteobacteria. *PLoS Biology*, 1(1):e19. doi:10.1371/journal.pbio.0000019.

Li, H. and Durbin, R., 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760. doi:10.1093/bioinformatics/btp324.

Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P. and Zhang, H., 2001. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154.

Li, W.-H., 1993. Unbiased estimation of the rates of synonymous and nonsynonymous substitution. *Journal of Molecular Evolution*, 36(1):96–99. doi:10.1007/bf02407308.

Li, W.-H., Wu, C. I. and Luo, C. C., 1985. A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes. *Molecular Biology and Evolution*, 2(2):150–174.

Lin, Y.-H., McLenachan, P. A., Gore, A. R., Phillips, M. J., Ota, R., Hendy, M. D. and Penny, D., 2002. Four new mitochondrial genomes and the increased stability of evolutionary trees of mammals from improved taxon sampling. *Molecular Biology and Evolution*, 19(12):2060–2070.

Lio, P. and Goldman, N., 1998. Models of molecular evolution and phylogeny. *Genome Research*, 8(12):1233–1244. doi:10.1101/gr.8.12.1233.

Loewenstern, D. and Yianilos, P. N., 1999. Significantly lower entropy estimates for natural DNA sequences. *Journal of Computational Biology*, 6(1):125–142.

Ma, B., Tromp, J. and Li, M., 2002. Patternhunter: Faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445. doi:10.1093/bioinformatics/18.3.440.

Mace, R. and Holden, C. J., 2005. A phylogenetic approach to cultural evolution. *Trends in Ecology & Evolution*, 20(3):116–121. doi:10.1016/j.tree.2004.12.002.

Maher, K. A. and Stevenson, D. J., 1988. Impact frustration of the origin of life. *Nature*, 331(6157):612–614. doi:10.1038/331612a0.

Manber, U. and Myers, G., 1991. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948.

Manzini, G. and Rastero, M., 2004. A simple and fast DNA compressor. *Software: Practice and Experience*, 34(14):1397–1411. doi:10.1002/spe.619.

Martin-Lof, P., 1966. On the definition of random sequences. *Information and Control*, 9:602–619.

Mascarenhas, D., Mettler, I. J., Pierce, D. A. and Lowe, H. W., 1990. Intron-mediated enhancement of heterologous gene expression in maize. *Plant Molecular Biology*, 15(6):913–920. doi:10.1007/bf00039430.

Matsumoto, T., Sadakane, K. and Imai, H., 2000. Biological sequence compression algorithms. *Genome Informatics*, 11:43–52.

McCarthy, B. J. and Holland, J. J., 1965. Denatured DNA as a direct template for in vitro protein synthesis. *Proceedings of the National Academy of Sciences*, 54(3):880–886.

McCreight, E. M., 1976. A space-economical suffix tree construction algorithm. *Journal of Algorithms*, 23(2):262–272.

McCulloch, S. D. and Kunkel, T. A., 2008. The fidelity of DNA synthesis by eukaryotic replicative and translesion synthesis polymerases. *Cell Research*, 18(1):148–161. doi:10.1038/cr.2008.4.

McGrath, C. L. and Katz, L. A., 2004. Genome diversity in microbial eukaryotes. *TRENDs in Ecology and Evolution*, 19(1):32–38.

Meyer, B. and Tischer, P. E., 1998. Extending TMW for near lossless compression of greyscale images. In *Data Compression Conference*, pp. 458–470. doi:10.1109/dcc.1998.672194.

Mighell, A. J., Markham, A. F. and Robinson, P. A., 1997. Alu sequences. *FEBS Letters*, 417(1):1 – 5. doi:10.1016/s0014-5793(97)01259-3.

Milosavljevic, A. and Jurka, J., 1993a. Discovering simple DNA sequences by the algorithmic significance method. *CABIOS*, 9(4):407–411.

Milosavljevic, A. and Jurka, J., 1993b. Discovery by minimal length encoding: A case study in molecular evolution. *Machine Learning*, 12(1-3):69–87. doi:10.1007/bf00993061.

Miyata, T. and Yasunaga, T., 1980. Molecular evolution of mRNA: A method for estimating evolutionary rates of synonymous and amino acid substitutions from homologous nucleotide sequences and its application. *Journal of Molecular Evolution*, 16(1):23–36. doi:10.1007/bf01732067.

Moffat, A., 1990. Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, 38(11):1917 – 1921. doi:10.1109/26.61469.

Morgenstern, B., 1999. DIALIGN 2: Improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15:211–218. doi:10.1093/bioinformatics/15.3.211.

Morgenstern, B., Rinner, O., Abdeddaim, S., Haase, D., Mayer, K., Dress, A. and Mewes, H.-W., 2002. Exon discovery by genomic sequence alignment. *Bioinformatics*, 18:777–787. doi:10.1093/bioinformatics/18.6.777.

Murphy, W. J., Eizirik, E., Johnson, W. E., Zhang, Y. P., Ryder, O. A. and O'Brien, S. J., 2001a. Molecular phylogenetics and the origins of placental mammals. *Nature*, 409(6820):614–618. doi:10.1038/35054550.

Murphy, W. J., Eizirik, E., O'Brien, S. J., Madsen, O., Scally, M., Douady, C. J., Teeling, E., Ryder, O. A., Stanhope, M. J., de Jong, W. W. et al., 2001b. Resolution of the early placental mammal radiation using bayesian phylogenetics. *Science*, 294(5550):2348–2351. doi:10.1126/science.1067179.

Nalbantoglu, Ö. U., Russell, D. J. and Sayood, K., 2010. Data compression concepts and algorithms and their applications to bioinformatics. *Entropy*, 12(1):34–52. doi:10.3390/e12010034.

Nan, F., 2006. Personal communication.

NCBI, 2003. Human genome release 36.
URL ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/April_14_2003. Accessed 2007.

Needleman, S. B. and Wunsch, C. D., 1970. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *Journal of Molecular Biology*, 48:443–453.

Nei, M., 1996. Phylogenetic analysis in molecular evolutionary genetics. *Annual Review of Genetics*, 30(1):371–403. doi:10.1146/annurev.genet.30.1.371.

Nei, M. and Gojobori, T., 1986. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Molecular Biology and Evolution*, 3(5):418–426.

Nei, M. and Kumar, S., 2000. *Molecular Evolution and Phylogenetics*. Oxford University Press.

Nei, M., Tajima, F. and Tateno, Y., 1983. Accuracy of estimated phylogenetic trees from molecular data. *Journal of Molecular Evolution*, 19(2):153–170. doi:10.1007/bf02300753.

Nei, M., Takezaki, N. and Sitnikova, T., 1995. Assessing molecular phylogenies. *Science*, 267(5195):253–254. doi:10.1126/science.7809633.

Nevill-Manning, C. G. and Witten, I. H., 1999. Protein is incompressible. In *Data Compression Conference*, pp. 257–266.

Ning, Z., Cox, A. J. and Mullikin, J. C., 2001. SSAHA: A fast search method for large DNA databases. *Genome Research*, 11(10):1725–1729. doi:10.1101/gr.194201.

Nishihara, H., Hasegawa, M. and Okada, N., 2006. Pegasoferae, an unexpected mammalian clade revealed by tracking ancient retroposon insertions. *Proceedings of the National Academy of Sciences*, 103(26):9929–9934. doi:10.1073/pnas.0603797103.

Ogura, Y., Bonen, D. K., Inohara, N., Nicolae, D. L., Chen, F. F., Ramos, R., Britton, H., Moran, T., Karaliuskas, R., Duerr, R. H. et al., 2001. A frameshift mutation in NOD2 associated with susceptibility to Crohn's disease. *Nature*, 411(6837):603–606. doi:10.1038/35079114.

Ohno, S., 1972. So much junk DNA in our genome. *Brookhaven Symposium on Biology*, 23:366–370.

Ondov, B. D., Varadarajan, A., Passalacqua, K. D. and Bergman, N. H., 2008. Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications. *Bioinformatics*, 24(23):2776–2777. doi:10.1093/bioinformatics/btn512.

Otto, S. P., 2003. The advantages of segregation and the evolution of sex. *Genetics*, 164(3):1099–1118.

Otu, H. H. and Sayood, K., 2003. A new sequence distance measure for phylogenetic tree construction. *Bioinformatics*, 19(16):2122–2130. doi:10.1093/bioinformatics/btg295.

Pamilo, P. and Bianchi, N. O., 1993. Evolution of the Zfx and Zfy genes: rates and interdependence between the genes. *Molecular Biology and Evolution*, 10(2):271–281.

Pandey, V., Nutter, R. C. and Prediger, E., 2008. *Applied Biosystems SOLiD System: Ligation-Based Sequencing*, chapter 3. Next-Generation Genome Sequencing: Towards Personalized Medicine. Wiley InterScience. doi:10.1002/9783527625130.ch3.

Pearson, W. R. and Lipman, D. J., 1988. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448.

PlasmoDB, 2008. Plasmodb: Plasmodium genome resource, release 5.4. URL http://www.plasmodb.org/common/downloads/release-5.4/. Accessed Sep 2008.

PlasmoDB, 2009a. Plasmodb: Plasmodium genome resource, release 5.5. URL http://www.plasmodb.org/common/downloads/release-5.5/. Accessed Jul 2009.

PlasmoDB, 2009b. Plasmodb: Plasmodium genome resource, release 6.2. URL http://www.plasmodb.org/common/downloads/release-6.2/. Accessed Nov 2009.

Powell, D. R., Allison, L. and Dix, T. I., 2004. Modelling-alignment for non-random sequences. In *AI 2004: Advances in Artificial Intelligence*, pp. 203–214.

Powell, D. R., Allison, L., Dix, T. I. and Dowe, D. L., 1998a. Alignment of low information sequences. In *Australian Computer Science Theory Symposium, CATS '98*, pp. 215–230. Springer Verlag.

Powell, D. R., Dowe, D. L., Allison, L. and Dix, T. I., 1998b. Discovering simple DNA sequences by compression. In *Pacific Symposium on Biocomputing '98*, pp. 597–608. World Scientic.

Raven, P. H., Evert, R. F. and Eichhorn, S. E., 2005. *Biology of Plants*. W. H. Freeman, New York, sixth edition.

Reyes, A., Gissi, C., Pesole, G., Catzeflis, F. M. and Saccone, C., 2000. Where do rodents fit? evidence from the complete mitochondrial genome of Sciurus vulgaris. *Molecular Biology and Evolution*, 17(6):979–983.

Rich, S. M., Leendertz, F. H., Xu, G., LeBreton, M., Djoko, C. F., Aminake, M. N., Takang, E. E., Diffo, J. L. D., Pike, B. L., Rosenthal, B. M. et al., 2009. The origin of malignant malaria. *Proceedings of the National Academy of Sciences*, 106(35):14902–14907. doi:10.1073/pnas.0907740106.

Rissanen, J., 1978. Modelling by the shortest data description. *Automatica*, 14:465–471.

Rissanen, J. and Langdon, G. J., 1979. Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162.

Rissanen, J. and Langdon, G. J., 1981. Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1):12–23.

Rivals, E., Delahaye, J.-P., Dauchet, M. and Delgrange, O., 1996. A guaranteed compression scheme for repetitive DNA sequences. In *Data Compression Conference*, p. 453.

Robinson, D. F. and Foulds, L. R., 1981. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131 – 147. doi:10.1016/0025-5564(81)90043-2.

Robinson-Rechavi, M., Ponger, L. and Mouchiroud, D., 2000. Nuclear gene LCAT supports rodent monophyly. *Molecular Biology and Evolution*, 17(9):1410–1412.

Rodin, A. and Li, W.-H., 2000. A rapid heuristic algorithm for finding minimum evolution trees. *Molecular Phylogenetics and Evolution*, 16(2):173 – 179. doi:10.1006/mpev.1999.0728.

Rybicki, E. P., 1990. The classification of organisms at the edge of life, or problems with virus systematics. *South African Journal of Science*, 86:182–186.

Rzhetsky, A. and Nei, M., 1992. A simple method for estimating and testing minimum-evolution trees. *Molecular Biology and Evolution*, 9(5):945–967.

Rzhetsky, A. and Nei, M., 1993. Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Molecular Biology and Evolution*, 10:1073–1095.

Saitou, N. and Nei, M., 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

Sanger, F., Nicklen, S. and Coulson, A. R., 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467.

Sankoff, D., Leduc, G., Antoine, N., Paquin, B., Lang, B. F. and Cedergren, R., 1992. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences*, 89(14):6575–6579.

Scally, M., Madsen, O., Douady, C. J., de Jong, W. W., Stanhope, M. J. and Springer, M. S., 2001. Molecular evidence for the major clades of placental mammals. *Journal of Mammalian Evolution*, 8(4):239–277. doi:10.1023/a:1014446915393.

Schürmann, K.-B. and Stoye, J., 2007. An incomplex algorithm for fast suffix array construction. *Software: Practice and Experience*, 37(3):309–329. doi:10.1002/spe.v37:3.

Schuster, S. C., 2008. Next-generation sequencing transforms today's biology. *Nature Methods*, 5(1):16–18. doi:10.1038/nmeth1156.

Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C., Haussler, D. and Miller, W., 2003. Human-mouse alignments with BLASTZ. *Genome Research*, 13(1):103–107. doi:10.1101/gr.809403.

Segre, G., 2000. The big bang and the genetic code. *Nature*, 404(6777):437–437. doi:10.1038/35006517.

Shannon, C. E., 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Siddall, M. E. and Barta, J. R., 1992. Phylogeny of plasmodium species: Estimation and inference. *The Journal of Parasitology*, 78(3):567–568.

Singer, M. F., 1982. SINEs and LINEs: Highly repeated short and long interspersed sequences in mammalian genomes. *Cell*, 28(3):433 – 434. doi:10.1016/0092-8674(82)90194-5.

Smith, T. F. and Waterman, M. S., 1981. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–147.

Snel, B., Bork, P. and Huynen, M. A., 1999. Genome phylogeny based on gene content. *Nat Genet*, 21(1):66–67.

Sokal, R. and Michener, C., 1958. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 28:1409–1438.

Solomonoff, R., 1964. A formal theory of inductive inference. *Information and Control*, 7(2):1–22,224–2.

Springer, M. S., Stanhope, M. J., Madsen, O. and de Jong, W. W., 2004. Molecules consolidate the placental mammal tree. *Trends in Ecology & Evolution*, 19(8):430 – 438. doi:10.1016/j.tree.2004.05.006.

Stern, L., Allison, L., Coppel, R. L. and Dix, T. I., 2001. Discovering patterns in Plasmodium falciparum genomic DNA. *Molecular and Biochemical Parasitology*, 118:175–186.

Stoye, J., Evers, D. and Meyer, F., 1998. Rose: Generating sequence families. *Bioinformatics*, 14(2):157–163. doi:10.1093/bioinformatics/14.2.157.

Sullivan, J. and Joyce, P., 2005. Model selection in phylogenetics. *Annual Review of Ecology, Evolution, and Systematics*, 36(1):445–466. doi:10.1146/annurev.ecolsys.36.102003.152633.

Swofford, D. L. and Begle, D. P., 1993. PAUP*: Phylogenetic analysis using parsimony, ver. 3.1 user manual. Illinois Natural History Survey.

Tabus, I., Korodi, G. and Rissanen, J., 2003. DNA sequence compression using the normalized maximum likelihood model for discrete regression. *DCC*, p. 253. doi:10.1109/dcc.2003.1194016.

Tajima, F. and Nei, M., 1984. Estimation of evolutionary distance between nucleotide sequences. *Molecular Biology and Evolution*, 1(3):269–285.

Takezaki, N. and Nei, M., 1996. Genetic distances and reconstruction of phylogenetic trees from microsatellite DNA. *Genetics*, 144(1):389–399.

Tavare, S., 1986. Some probabilistic and statistical problems on the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences*, 17:262–272.

Thomas, C. A., 1971. The genetic organization of chromosomes. *Annual Review of Genetics*, 5(1):237–256. doi:10.1146/annurev.ge.05.120171.001321.

Thomas, C. M. and Nielsen, K. M., 2005. Mechanisms of, and barriers to, horizontal gene transfer between bacteria. *Nature Reviews Microbiology*, 3(9):711–721. doi:10.1038/nrmicro1234.

Uzzell, T. and Corbin, K. W., 1971. Fitting Discrete Probability Distributions to Evolutionary Events. *Science*, 172(3988):1089–1096. doi:10.1126/science.172.3988.1089.

Vendrely, R. and Vendrely, C., 1948. La teneur du noyau cellulaire en acide désoxyribonucléique à travers les organes, les individus et les espèces animales: Techniques et premiers résultats. *Experientia*, 4:434–436.

Venugopal, K., Srinivasa, K. and Patnaik, L., 2009. *Non-repetitive DNA Compression Using Memoization*, chapter 15, pp. 291–301. Studies in Computational Intelligence. Springer Berlin / Heidelberg. doi:10.1007/978-3-642-00193-2_15.

Vinga, S. and Almeida, J., 2003. Alignment-free sequence comparison - a review. *Bioinformatics*, 19(4):513–523.

Wallace, C. S., 2005. *Statistical and Inductive Inference by Minimum Message Length*. Springer.

Wallace, C. S. and Boulton, D. M., 1968. An information measure for classification. *Computer Journal*, 11(2):185–194.

Wallace, C. S. and Freeman, P. R., 1987. Estimation and inference by compact coding. *Journal of the Royal Statistical Society series*, 49(3):240–265.

Waters, A. P., Higgins, D. G. and McCutchan, T. F., 1991. Plasmodium falciparum appears to have arisen as a result of lateral transfer between avian and human hosts. *Proceedings of the National Academy of Sciences*, 88(8):3140–3144.

Waters, A. P., Higgins, D. G. and McCutchan, T. F., 1993. Evolutionary relatedness of some primate models of plasmodium. *Molecular Biology and Evolution*, 10(4):914–923.

Watson, J. D., Baker, T. A., Bell, S. P., Gann, A., Levine, M. and Losick, R., 2008. *Molecular Biology of the Gene*. Peason Education, sixth edition.

Watson, J. D. and Crick, F. H. C., 1953. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737 – 738. doi:10.1038/171737a0.

Weiner, P., 1973. Linear pattern matching algorithms. In *14th Annual IEEE Symposium on Switching and Automata Theory*.

Willems, F., Shtarkov, Y. M. and Tjalkens, T. J., 1995. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, pp. 653–664.

Williams, P. L. and Fitch, W. M., 1990. Phylogeny determination using dynamically weighted parsimony method. *Methods in Enzymology*, 183:615–626.

Williams, R. N., 1991. *Adaptive Data Compression*. Kluwer Academic Publishers.

Witten, I. H., Neal, R. M. and Cleary, J. G., 1987. Arithmetic coding for data compression. *Comm ACM*, 30(6):520–540.

Woese, C. R., Kandler, O. and Wheelis, M. L., 1990. Towards a natural system of organisms: Proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences*, 87(12):4576–4579.

Wootton, J. C., 1997. Simple sequences of protein and DNA. In Bishop, M. J. and Rawlings, C. J., eds., *DNA and Protein Sequence Analysis: A Practical Approach*, pp. 169–183. Oxford University Press.

Wootton, J. C. and Federhen, S., 1993. Statistics of local complexity in amino acid sequences and sequence databases. *Computers & Chemistry*, 17(2):149 – 163. doi:10.1016/0097-8485(93)85006-x.

Yang, Z., 1994. Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 39(1):105–111. doi:10.1007/bf00178256.

Yang, Z. and Nielsen, R., 2008. Mutation-selection models of codon substitution and their use to estimate selective strengths on codon usage. *Molecular Biology and Evolution*, 25:568–579.

Yang, Z. and Rannala, B., 1997. Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method. *Molecular Biology and Evolution*, 14(7):717–724.

Yap, V. B. and Speed, T. P., 2004. Modeling DNA base substitution in large genomic regions from two organisms. *Journal of Molecular Evolution*, 58(1):12–18. doi:10.1007/s00239-003-2520-8.

Zhi, D., Raphael, B., Price, A., Tang, H. and Pevzner, P., 2006. Identifying repeat domains in large genomes. *Genome Biology*, 7(1):R7. doi:10.1186/gb-2006-7-1-r7.

Ziv, J. and Lempel, A., 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–342.

Ziv, J. and Lempel, A., 1978. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536. doi:10.1109/TIT.1978.1055934.