THE UNIVERSITY OF MANITOBA

STATISTICAL PATTERN RECOGNITION IN GENOMIC DNA SEQUENCES

by

Leo Wang-Kit Cheung

A Thesis submitted to the Faculty of Graduate Studies in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy

Department of Statistics

Winnipeg, Manitoba

Canada R3T 2N2

April, 2002

0-612-76719-1

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

STATISTICAL PATTERN RECOGNITION IN GENOMIC DNA SEQUENCES

BY

Leo Wang-Kit Cheung

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree

of

DOCTOR OF PHILOSOPHY

Contents

Abstract

This thesis is concerned with probabilistic and statistical approaches for pattern recognition in genomic DNA sequences. Building probabilistic models with a hidden Markov model (HMM) structure and investigating runs-related statistics are two distinct mathematical/statistical/computational research topics that have been widely utilized in the area of bioinformatics. This work coalesces both topics and provides ideas on further broadening them.

The use of the finite Markov chain imbedding (FMCI) technique to study DNA patterns under an HMM is introduced. With a vision of studying multiple runs-related statistics simultaneously under an HMM through the FMCI technique, this work establishes an investigation of a double runs statistic under a binary HMM for DNA pattern recognition. An FMCI-based recursive algorithm is derived and implemented for the determination of the exact distribution of this double runs statistic under an independent identically distributed (IID) framework, a Markov chain (MC) framework, and a binary HMM framework. With this algorithm, a conditional runs test is revised and used to test for randomness against clustering of signals in DNA. Having studied the distributions of the double runs statistic under different binary HMM parameter sets, probabilistic profiles of runs are created and shown to be useful for trapping HMM maximum likelihood estimates (MLEs). This MLE-trapping scheme offers good initial estimates which not only jump-start the Expectation-Maximization (EM) algorithm, but also prevent the EM estimates from landing on a local maximum. Based on parametric bootstrapping with the MLE-trapping scheme, simple methods are used and implemented to construct confidence intervals for the HMM parameters. Applications of the conditional runs statistic, the double runs statistic, and the probabilistic profiles in conjunction with binary HMMs for DNA pattern recognition are demonstrated using human DNA data.

A multivariate class of probabilistic models, Hidden Multivariate Markov Models (HM^{3}s), is also introduced for modelling DNA sequences. As biochemical and biophysical evidence indicates that DNA molecules possess many different aspects beyond their compositional content, creating probabilistic models from a multivariate perspective makes natural biological sense for the analysis of DNA. A bivariate version of the HM^3 is developed for exploration of the joint behaviour of the C+G richness pattern and the bendability pattern of DNA.

Acknowledgments

It has been quite a journey. I would first like to sincerely thank my thesis supervisor, Dr. John Brewster, for his openness, guidance and constant support. His openness allowed me to truly follow my heart and enter the field of bioinformatics. His skillful supervision not only provided me freedom to make my own mistakes and discoveries, but also ensured that I remained on the right track. Over the years, he has taught me invaluable skills to become a researcher, a consultant, and a teacher at the professional level. In particular, I feel very fortunate to have been part of the Statistical Advisory Service (SAS) that he led. Having the opportunity to work as a statistical consultant for SAS, I have discovered my passion in applying mathematical, statistical, and computational methods to help answer real-life problems. I am sure that my consulting experiences have both directly and indirectly enhanced my thesis research.

I would also like to thank the other members of my Ph.D. thesis committee — Dr. James Fu, Dr. Ken Mount, Dr. John Braun, and Dr. Brian Fristensky — for their comments and questions. Their positive comments have helped me stay encouraged and build confidence. Their negative comments have helped me not only maintain a critical thinking, but also continuously push my limits and drive for further improvement. Dr. Fu's knowledge in probability and his ways of thinking have always been an inspiration. Dr. Mount's teaching and mentoring in statistical consulting and his philosophy of life have always helped me keep a "level head" on my shoulders. Every time I talk to Dr. Mount, I always realize that I become a bit"more polishe" (both professionally and personally). My association with Dr. Braun has been the longest (even before I started graduate school). Having worked as an undergraduate research assistant on the topic of *Tumor Growth*

Development — Simulation Studies for him at the University of Winnipeg, I was first enlightened by him in the area of mathematical biology. My research experience with him certainly shaped my intense interests in statistical genetics. Special thanks go to Dr. Fristensky (professor at the Department of Plant Science) for making sure that the writing in biological sections of my thesis was rigorous. I really appreciate his efforts in going through my statistical work as well.

Special thanks also go to Dr. Leigh Murphy, Dr. Barbara Triggs-Raine, and Dr. Klaus Wrogemann (professors at the Department of Biochemistry and Medical Genetics) for their patience to help reinforce my understanding of nucleic acids. Thanks also go to Dr. Gary Churchill at the Jackson Laboratory and Dr. Anders Pedersen at the Center for Biological Sequence Analysis for their correspondences over the years. Specifically, Dr. Churchill's work on HMMs for DNA sequences has been a main source of inspiration to me. The supply of the non-redundant human RNA polymerase II promoter sequence data sets by Dr. Pedersen and my discussions on the eukaryotic gene structure and DNA bending/bendability with him have greatly contributed to my research.

My thesis work was supported in part by the University of Manitoba Graduate Fellowship and the Manitoba Health Research Council Studentship. In addition, during my Ph.D. program, I was also supported by local as well as international institutes and organizations to go to different tutorials, workshops and short courses to learn various topics in the areas of statistical genetics and bioinformatics. I would like to acknowledge the Statistical Advisory Service, the Statistical Quality Control Research and Applications Group, the Department of Statistics, the Institute of Industrial Mathematical Sciences, the Faculty of Science, and the Faculty of Graduate Studies at the University of Manitoba; the Summer Institute in Statistical Genetics at North Carolina State University; the Mathematical Sciences Research Institute in Berkeley; the Program in Mathematics and Molecular Biology at Florida State University; the 9th International Conference on Intelligent Systems for Molecular Biology in Denmark; the Jackson Laboratory in Bar Harbor; and the Wellcome Trust Genome Campus in the United Kingdom. I am extremely grateful for their generous support. They allowed me to build a comprehensive foundation for my research, and provided me excellent opportunties to meet and interact with a lot of the world-class researchers.

The list of thanks would not be complete without a mention of my friends and family. I have been extremely fortunate to have great friendships with a lot of wonderful people in Winnipeg. First, I would like to thank three of my best friends, Andrew Willems, Monique Dion ("Miss Putman"), and Craig Hepworth, for their confidence in my ability to pursue a thesis topic in bioinformatics. I first met Andrew in my biology and genetics classes, Monique in my computer science classes, and Craig in my mathematics and statistics classes at the University of Winnipeg. I really cherish the hard-working moments that we shared during our undergraduate studies, the great times that we had outside school, and the longtime friendships that we still hold. In particular, I will never forget the most delicious banana and coconut muffins that Monique makes, they are always a delight. My friendships with them — a great three — over the last 13 years have been both personally and professionally rewarding and have greatly contributed to my intellectural development. I would also like to thank Andrew's parents, Betty and Peter Willems, for their generous love and constant care over the years. Special thanks to them for making me feel part of their family when I have often missed mine in Hong Kong. Other amazing people whom I got to know at the University of Manitoba are Anthony Yeung (a definite computer genius), Dr. Dennis Murphy, Llwellyn Armstrong, Dr. Catherine Njue, and Zak Pritchard. I deeply appreciate Anthony and Dennis for their genuine willingness to always lend a hand to help solve almost all technical difficulties/problems that I experienced on the UNIX system. My discussions and exchanges of consulting experiences with Llwellyn, Catherine, and Zak at the Statistical Advisory Service had certainly helped me enjoy being a statistical consultant even more than I think I would. All these people have helped make my living and learning in Winnipeg enjoyable even at -30°C in the winters. They have also made the city of Winnipeg, which used to be a foreign place to me, feel a place like home.

Last, but certainly not the least, I would like to thank my parents, Sui-Ping and Wai-Kong Cheung, not only for their genetic contributions, but also for their unconditional love, and constant support. They have done so much to shape my life and help me achieve this goal. I can never thank them enough in teaching me to work hard and to never fall short of my dreams. They have always been the anchor of my life. I would also like to thank my elder sister Doris, my younger brother

Hero, and my younger sister Angel for their constant encouragement over the years. I owe a great deal and more to all of them.

To the memory of my dearest grandma Ming and grandpa Kan, who not only had the biggest belief in me, but also taught me to believe in myself and follow my dreams. Unfortunately, I could only share the accomplishment of this work with them in spirit. I would like to dedicate this work to them.

List of Tables

List of Figures

Abbreviations and Notation

EM — **E**xpectation-**M**aximization

FMCI — **F**inite **M**arkov **C**hain **I**mbedding

HMM — **H**idden **M**arkov **M**odel

HM^3 — **H**idden **M**ultivariate **M**arkov **M**odel

MLE — **M**aximum **L**ikelihood **E**stimate

N = The number of "hidden" discrete states

M = The number of discrete outcomes

L = The length of a DNA sequence being modelled

$\Gamma_L = \{1, \ldots, L\}$ — A finite index set

Floor $(\cdot)$ — An operator rounds its operand to the largest integer not exceeding the operand

(N_s, R) — A double runs statistic with the first component being the number of successes and the second component being the number of success runs

$\{X_t : t \in \Gamma_L\} = \{X_1, X_2, \cdots, X_t, \cdots, X_L\}$ — A hidden stochastic process denoted as a *state process*

$\{Y_t : t \in \Gamma_L\} = \{Y_1, Y_2, \cdots, Y_t, \cdots, Y_L\}$ — A stochastic process denoted as an *outcome process*

$X_{[t,t+T]} = \{X_t, X_{t+1}, \cdots, X_{t+T}\}$

$Y_{[t,t+T]} = \{Y_t, Y_{t+1}, \cdots, Y_{t+T}\}$

$\boldsymbol{\theta}$: Parameter set of an HMM or HM^3

$\widehat{\boldsymbol{\theta}}$: MLE of $\boldsymbol{\theta}$

$\{\widehat{\boldsymbol{\theta}}_1^{\diamond}, \widehat{\boldsymbol{\theta}}_2^{\diamond}, \cdots, \widehat{\boldsymbol{\theta}}_{b_s}^{\diamond}, \cdots, \widehat{\boldsymbol{\theta}}_{B_s}^{\diamond}\}$ — B_s bootstrap replications of $\widehat{\boldsymbol{\theta}}$

$\mathcal{L}_Y(\boldsymbol{\theta})$: Likelihood function

$\mathcal{L}_{YX}(\boldsymbol{\theta})$: Augmented-data likelihood function

Chapter 1

INTRODUCTION

1.1 The Field of Statistical Genetics: An Overview

The discipline of *statistics* has strong roots in the discipline of *genetics*. After the rediscovery of the 1865 paper of J. Gregor Mendel* on the inheritance of the characteristics of peas, modern genetics began in about 1900 (after Mendel's death). Both *Mendel's First and Second Laws* were fundamentally probabilistic. The science of genetics attempts to understand the properties of the genetic material. Since *genes* are the basic functional units of *heredity*, they have naturally and historically been the focus of genetics as geneticists try to understand the laws of heredity. Heredity is concerned with the biological similarity of offspring to parents. *Variation* among organisms is the biological difference between parents and offspring and between the offspring themselves. In fact, variation is the raw material of evolution, and it is the key element for genetic studies. As defined by Griffiths, Miller, Suzuki, Lewontin, and Gelbart, genetics is *the study of genes through their variation*. The study and understanding of variation is intrinsically statistical. As a matter of fact, a lot of the concepts and theories in genetics, from independent assortment to the existence of repetitive DNA, have been based on quantitative (especially statistical) analyses of genetic data

*J. Gregor Mendel (1822–1884) was later called the father of modern genetics.

(Thompson, 1999 [128]; Griffiths et al., 2000 [63]).

Early work of Sir Ronald A. Fisher[†], S. Wright, and J. B. S. Haldane had formed some foundations for the interface between statistics and genetics, the field of *statistical genetics.* They realized that probabilistic modelling, rigorous statistical analysis, and well-founded scientific inference could help study and understand observable genetic variation. With advances in recombinant DNA technology, now it is not only possible to study genetic variation at the molecular level from different aspects (such as biochemical and biophysical aspects), but also feasible to have large-scale or major DNA sequencing tasks completed (Thompson, 1999 [128]). In 1990, the *Human Genome Project* formally began. Complete genomes of various organisms, including a number of viruses (such as the AIDS virus, HIV-1[‡]), bacteria (such as *Escherichia coli*) and the budding yeast *Saccharomyces cerevisiae*, had been first determined. Unified maps, such as *gene maps*, that integrate *genetic linkage maps* and *physical maps* with the DNA and protein sequence databases, of these organisms' genomes are now available. At about the end of 1994, an international consortium of genome centers and groups in North America, Europe, and Japan was organized to coordinate a mapping effort for constructing a human gene map. A gene map[§] of the human genome with sufficient accuracy and resolution to within a few megabases was first released in about 1996 through an Internet site hosted by the National Center for Biotechnology Information (NCBI) of the National Institutes of Health (NIH) on the World Wide Web at URL http://www.ncbi.nlm.nih.gov/SCIENCE96/ (Schuler et al., 1996 [121]).

On the 26th of June 2000, the National Human Genome Research Institute (NHGRI) and Celera Genomics announced that they had "working drafts" of the human genome, the sequence of about

[†] Sir Ronald A. Fisher (1890–1962) was later called the father of statistics.

[‡] NOTE: The human immunodeficiency virus (HIV) is the causative agent for the acquired immunodeficiency syndrome (AIDS). The most common type is known as human immunodeficiency virus type 1 (HIV-1) which has led to the worldwide AIDS epidemic.

[§] NOTE: An updated version of the human gene map is available on the World Wide Web at URL http://www.ncbi.nlm.nih.gov/genemap.

3.1 billion subunits of DNA contained in human chromosomes (Branswell, 2000 [23]). Having the complete human genome sequenced is just the first ultimate goal of the Human Genome Project. Another key ultimate goal of the Human Genome Project is to support research, training programs and curricula to nurture the field of *bioinformatics*; that is the development of mathematical techniques, statistical tools and computational strategies for the collection, analysis, annotation and storage of the huge amounts of data from DNA sequencing, mapping, and gene expression experiments/studies (Collins[¶] et al., 1998 [36]; Yarbrough & Thompson, 2000 [158]). Organizing, managing, extracting, analyzing and interpreting massive amounts of sequence data/information are multidisciplinary missions. To reach this goal, close collaboration among many different scientific disciplines has commenced. Since these missions are heavily mathematical, statistical, and computational in nature, they have created a lot of challenges and research opportunities for mathematicians, statisticians and computer scientists. A new branch of science has subsequently emerged by crossing a field in the mathematical sciences (such as statistics, computer science, or a cross between statistics and computer science themselves (e.g. computational statistics)) with a field in the biological sciences (such as genetics or molecular biology). Terms such as *statistical genetics*, *computational molecular biology*, and *bioinformatics* are used to denote these "hybrid" disciplines.

A brief description of the field of *statistical genetics* has been given by professor Elizabeth A. Thompson[‖][**] (Thompson, 1999 [128]), and its definition provided by professor Thompson is quoted

[¶] Dr. Francis S. Collins, M.D., Ph.D., is the director of the National Human Genome Research Institute (NHGRI) of the National Institutes of Health (NIH), Bethesda, U.S.A.

[‖] Dr. Elizabeth A. Thompson (Ph.D. in Statistics) is a professor of the Department of Statistics and the Department of Biostatistics, an adjunct professor of Genetics, a senior faculty and the designated leader of the Statistical Genetics Program at the University of Washington in Seattle, U.S.A. She is also an adjunct professor of Statistics at the North Carolina State University in Raleigh, U.S.A.; and a member of the Program in Mathematics and Molecular Biology (PMMB).

[**] NOTE: The Program in Mathematics and Molecular Biology (PMMB) is a multi-university interdisciplinary national research and training consortium, funded by the National Science Foundation (NSF) and the Burroughs Wellcome Fund (BWF) Interfaces Program. The overall goal of PMMB is the continued expansion of the applications

as follows:

> ***Statistical genetics*** is the development of models and methods for the analysis and interpretation of genetic data observed at any level from the cell nucleus to the species. Genetic data may be used for constructing the evolutionary history of species, characterizing the structure of populations, quantifying genetic diversity in natural populations and in domesticated species, detecting and localizing genes affecting human disease, or economic traits in plants and animals, determining the structure of proteins and the functional and structural domains in DNA sequences. (http://www.biostat.washington.edu/ statgen/Statgen/what_more.shtml)

The title of my thesis is *statistical pattern recognition in genomic DNA sequences.* The work described in my thesis is essentially interdisciplinary in nature, but emphasis is put on the development of probabilistic models and runs-related statistics for DNA pattern recogition. It is an active area of research generally labelled bioinformatics. By professor Thompson's description above, it certainly falls within the realm of statistical genetics. By the same token, Balding, Bishop and Cannings, editors of the *Handbook of Statistical Genetics*, have also identified bioinformatics as a part of statistical genetics (Balding et al., 2001 [10]).

1.2 Research Compendium

The primary focus of this work is on probabilistic and statistical approaches to DNA sequence analysis, but the models and methods described and developed herein are applicable to other sequence data. Carefully designed probabilistic models, such as various hidden Markov models (HMMs) and the like, have been shown to be capable of capturing important biological information buried in biomolecular sequences (e.g. Churchill, 1992 [32]; Krogh et al., 1994 [85]; Pedersen et al., 1996 [108]; Burge & Karlin, 1997 [27]; Baldi & Brunak, 1998 [4]; Durbin et al., 1998 [45]; Vingron, 2001 [132]).

of mathematics, including statistics and computer science, to molecular biology.

Moreover, specific runs-related statistics and their distributions have also been demonstrated to be useful for detecting patterns in biomolecular sequences (e.g. Karlin et al.,1992, 1996, 1997 [75, 79, 80]; Waterman, 1995 [144]; Leung & Yamashita, 1999 [93]; Fu et al., 1999 [59]; Lou, 2000 [99]). Although distinct streams of research on building different HMMs and on investigating various runs statistics for biomolecular sequence pattern recognition have been two influential topics in the area of bioinformatics, the idea of studying runs statistics under an HMM for biomolecular sequence analysis has never been proposed. With ideas on further broadening both topics, this work also coalesces them.

A novel idea of using the finite Markov chain imbedding (FMCI) technique to study runs and patterns in a DNA sequence under an HMM is introduced. With a vision of studying elaborate multiple runs-related statistics simultaneously under an HMM through the FMCI technique, this work establishes an investigation of a double runs statistic under a binary HMM for DNA pattern recognition. The double runs statistic, denoted as (N_s, R), is defined for a binary sequence or a sequence of dichotomous trials (e.g. an outcome is either a success S or a failure F at each trial). It is a bivariate random variable with the first component being the total number of successes (i.e. N_s) and the second component being the number of success runs (i.e R) in a sequence of dichotomous S/F trials. A recursive algorithm based on the FMCI technique is derived and implemented for the determination of the exact distribution of this double runs statistic under an independent identically distributed (IID) framework, a Markov chain (MC) framework, and a binary HMM framework. With this algorithm, a conditional runs test is revised and used to test for randomness against clustering of signals in DNA sequence data. Having studied the distributions of the double runs statistic under different binary HMM parameter sets, probabilistic profiles of (N_s, R) are created. These probabilistic profiles are shown to be useful exploratory tools for trapping parameter estimates of a binary HMM. Specifically, once the double runs statistic of a sequence of outcomes is observed, trapping grids built from the probabilistic profiles of (N_s, R) can be used to locate the neighbourhood of the maximum likelihood estimates (MLEs) of the parameters of a binary HMM. Subsequently, good initial estimates can be selected for the Expectation-Maximization (EM) algorithm in the HMM parameter estimation procedure. Since the trapping scheme offers initial estimates that can substantially push the EM algorithm to "jump-start" in the neighbourhood of the MLEs, it helps prevent the EM estimates from landing on a local maximum or a saddle point of the likelihood surface in most cases. As a result, two difficult issues associated with the EM algorithm (i.e. having a slow rate of convergence and landing on a local maximum or a saddle point) in the HMM parameter estimation procedure are tackled simultaneously. Based on parametric bootstrapping with the MLE-trapping scheme, simple methods are also used and implemented to construct confidence intervals for the HMM parameters.

Applications of the conditional runs statistic, the double runs statistic, and the probabilistic profiles in conjunction with binary HMMs for DNA pattern recognition are demonstrated through studies using human DNA data provided by Dr. Anders Pedersen[††]. The core biological interest of our studies is to recognize the start sites of transcription of RNA polymerase II transcribed genes in experimentally uncharacterized human DNA sequences. Since we are only trying to reveal the start site of transcription from a relatively short sequence, it is a simplified version of the DNA decoding problem in our studies. The statistical treatment for this goal is based on the notion of capturing the mosaic structure of various signals along a DNA sequence. With general biological findings supporting the distribution of certain signals in the upstream promoter region of a gene differing from that in the downstream, we make use of the conditional runs statistic $(R|N_s)$, the double runs statistic (N_s, R), and an appropriate binary HMM for pattern recognition and the prediction of the start site of transcription of a RNA polymerase II transcribed gene. A structural aspect of DNA — DNA bendability or bending propensity — is analyzed. Based on the results reported by Pedersen and his colleagues (Pedersen et al., 1998 [109]), the continuous DNA bendability scales along a DNA sequence are dichotomized in two ways for two separate analyses in our studies.

A novel multivariate class of probabilistic models extended from the typical hidden Markov models, named "Hidden Multivariate Markov Models" or abbreviated as HM^{3}s, is also introduced

[††] NOTE: Dr. Anders G. Pedersen (Ph.D. & candidatus scientarium (Danish degree) in Biochemistry with Molecular Biology as main subject) is a senior researcher and an assistant professor, at the Center for Biological Sequence Analysis (CBSA) of the Technical University of Denmark in Lyngby, Denmark.

for modelling DNA sequences. As more and more biochemical and biophysical evidence indicates that DNA molecules possess many different aspects beyond their compositional content, creating probabilistic models from a multivariate perspective makes natural biological sense for the analysis of DNA. Essentially, the development of this class of models is an attempt to open the door for formal multivariate statistical analyses of multiple biochemical and biophysical aspects of DNA, especially for recognition and prediction of promoter regions of eukaryotic RNA polymerase II transcribed genes. In the light of the fact that both C+G richness and bendability of DNA sequences are now individually found to have a connection with promoter regions of eukaryotic genes (Antequera & Bird, 1993 [3]; Bernardi, 1993, 1995 [18, 19]; Cross & Bird, 1995 [37]; Pedersen et al., 1998, 1999 [109, 110]), a bivariate version of HM^{3}s is theoretically developed. The bivariate HM^{3}s offer a new statistical framework within which the joint behaviour of the C+G richness pattern and the bendability pattern of DNA sequences can be explored. They may help shed new light on computational promoter prediction and gene identification for experimentally uncharacterized DNA sequences, and may aid in getting more insights about the transcriptional control of gene expression and genome organization.

1.3 Organization

The following chapters of this thesis are organized so as to give the general background material first before introducing the specific technical details. In Chapter 2, a brief description of the background and significance on my research work —statistical pattern recognition in genomic DNA sequences — is provided with an emphasis on the recognition of promoter regions of eukaryotic RNA polymerase II transcribed genes. In Chapter 3, the univariate portrayal of probabilistic models for DNA sequences is described with a statement on its development and recent trends. Then, the basics of the typical hidden Markov models (HMMs) are reviewed in order to set the stage for the following chapters. In Chapter 4, the finite Markov chain imbedding (FMCI) technique for the distribution theory of runs is reviewed. The recursive algorithm for the calculation of the distribution of the double runs statistic

(N_s, R) and the examination of HMM parameter estimation with probabilistic profiles of runs are presented. Then, applications of the conditional runs statistic $(R|N_s)$, the double runs statistic (N_s, R), and the probabilistic profiles in conjunction with binary HMMs for pattern recognition in DNA are demonstrated through studies using the human genomic DNA sequence data. In Chapter 5, the logic and impetus for my research on a novel multivariate portrayal of probabilistic models for DNA sequences are provided. A new class of multivariate probabilistic models, hidden multivariate Markov models (HM^{3}s), is presented. Finally, in Chapter 6, further extensions and applications of this work are discussed.

Chapter 2

BACKGROUND AND SIGNIFICANCE

2.1 The Driving Force: The Human Genome Project

Since James Watson and Francis Crick proposed the *double-helix* model for the structure of a DNA (deoxyribo-nucleic acid) molecule, in 1953, a tremendous series of discoveries and breakthroughs have been made. The deciphering of the genetic code completed in 1967 was just the beginning (Frank-Kamenet̂skii, 1997 [54]). With advances in recombinant DNA technology, it is now possible to manipulate biomolecular sequences in order to study the biochemical functions and structural details of genes, and their organizations in genomes. Around 1985, ideas of sequencing the human genome had been proposed. In 1986, after Renato Dulbecco, a Nobel laureate in medicine, published an article on the implications of how sequencing the human genome would lead to new horizons in cancer research, an international *Workshop on Sequencing the Human Genome* was organized. The workshop brought together scientific leaders from industry, academia, and the national laboratories to assess the technical feasibility of sequencing the human genome, the probable cost, and the potential benefits to the nation (Dulbecco, 1986 [44]; DeLisi, 1988 [41]). In 1900, the first international

"Big Science" project in biology and medicine — the *Human Genome Project* — formally began. This ambitious project is dedicated to learning more about human health and disease. With the completion of the whole genomes of various organisms, including a number of viruses (such as the AIDS virus (HIV-1), the herpes simplex virus type 1 (HSV-1), and the hepatitis B virus (HBV)), bacteria (such as *Escherichia coli*), the budding yeast *Saccharomyces cerevisiae*, and the round worm *Caenorhabditis elegans*, the era of whole genome science has begun (Collins et al., 1998 [36]; *Entrez** Databases, 2000).

The Human Genome Project is a truly international effort to understand the structure and function of the human genome. The international Human Genome Sequencing Consortium includes the following sixteen research institutes in the United States, Great Britain, Germany, France, Japan, and China:

1. Baylor College of Medicine, Houston, Texas, U.S.A.
2. Beijing Human Genome Center, Institute of Genetics, Chinese Academy of Sciences, Beijing, China
3. Gesellschaft für Biotechnologische Forschung mbH, Braunschweig, Germany
4. Genoscope, Evry, France
5. Genome Therapeutics Corporation, Waltham, MA, U.S.A.
6. Institute for Molecular Biotechnology, Jena, Germany
7. Joint Genome Institute, U.S. Department of Energy, Walnut Creek, CA, U.S.A.
8. Keio University, Tokyo, Japan
9. Max Planck Institute for Molecular Genetics, Berlin, Germany
10. RIKEN Genomic Sciences Center, Saitama, Japan
11. The Sanger Centre, Hinxton, U.K.
12. Stanford DNA Sequencing and Technology Development Center, Palo Alto, CA, U.S.A.

*NOTE: *Entrez* is a search and retrieval system that integrates information from databases of nucleotide sequences, protein sequences, macromolecular structures, whole genomes, and MEDLINE, at the National Center for Biotechnology Information (NCBI) of the National Institutes of Health (NIH). *Entrez* is on the World Wide Web at URL http://www.ncbi.nlm.nih.gov:80/Database/index.html.

13. University of Washington Genome Center, Seattle, WA, U.S.A.
14. University of Washington Multimegabase Sequencing Center, Seattle, WA, U.S.A.
15. Whitehead Institute for Biomedical Research, MIT, Cambridge, MA, U.S.A.
16. Washington University Genome Sequencing Center, St. Louis, MO, U.S.A.

(Yarbrough & Thompson, 2000 [158])

The first goal of the Human Genome Project is to sequence the complete human genome and genomes of at least five other species or model organisms. On the 26th of June, 2000, the National Human Genome Research Institute (NHGRI) and Celera Genomics announced that they had "working drafts" of the human genome, the sequence of about 3.1 billion subunits of DNA contained in human chromosomes (Branswell, 2000 [23]). Now the immediate focus is on converting these "working drafts" to "finished" forms. This involves filling the gaps in the "working drafts" and increasing the overall sequence accuracy to 99.99%. Pieces of the "working drafts" have already helped the identification of genes for breast cancer susceptibility (BRCA2); hereditary deafness (Pendred syndrome); several hereditary skeletal disorders; hemorrhagic stroke; focal segmental glomerulosclerosis, a puzzling kidney disorder that can lead to end-stage kidney failure; hereditary epilepsy; and one type of diabetes. Recently, based on the information in the early "working drafts", a drug has been developed for leukemia and preliminary reports from clinical trials of the drug are showing very promising results (Yarbrough & Thompson, 2000 [158]).

Other key goals of the Human Genome Project are to create *genome maps* such as *genetic linkage maps* and *physical maps* which can then be used to construct *gene maps*; to study different kinds of sequence variation such as single nucleotide polymorphisms (SNPs), insertions, deletions, duplications, and rearrangements; to understand functions of genes, regulatory elements and various regions of DNA in genomes; to examine the ethical, legal, and social implications of genome research; to implement the results of the Human Genome Project into health care; and to nurture the development of bioinformatics for the collection, analysis, annotation and storage of the huge amounts of data from DNA sequencing, mapping, and gene expression experiments/studies. Full success of the Human Genome Project critically depends on the development of appropriate statistical and computational tools and rigorous criteria for establishing and confirming associations and/or causations among sequence variation, phenotypic variation, and complex diseases. Studying the functions of genes and DNA elements in a genome requires mathematical, statistical, and computational methods for comparison and analysis of sequence patterns (Marshall, 1996 [100]; Collins et al., 1998 [36]; Yarbrough & Thompson, 2000 [158]).

2.2 Computational Molecular Biology and Bioinformatics

Some of the early literature in the analysis of biomolecular sequences includes publications by Waterman et al., 1976, 1978, 1979, 1984, 1986, 1988, 1989 [147, 135, 136, 138, 137, 145, 139, 140, 146, 141, 142]; Karlin et al., 1983, 1984, 1985 [77, 73, 76, 81]; Weir, 1985, 1988 [148, 149]; Benham, 1985, 1989 [14, 15]; Bishop & Thompson, 1986 [22]; Brendel et al., 1986 [24]; Searls, 1988 [122]; and Churchill, 1989 [31]. These publications have enlightened researchers, including myself, from many different disciplines to participate in the multi-faceted genome research. The understanding of biomolecular sequences in turn gives us knowledge of the genetic nature of many diseases, and stimulates scientists to undertake new medical research and to develop new drugs or therapies. As the Human Genome Project progresses, biomolecular sequence data has been generated at an exponential rate. The need to organize and process the information and to find and interpret encoded messages has created problems that are interdisciplinary in nature. For example, questions in classifying various regions of DNA in genomes; aligning DNA sequences for comparison; testing whether two different DNA sequences share significant similarities; searching and recognizing patterns within or between DNA sequences; and predicting secondary and/or tertiary structures and functions of biomolecular sequences are grand mathematical, statistical and computational challenges. New terms such as *computational molecular biology* and *bioinformatics* are used to denote a new field dedicated to mathematical, statistical and computational analyses of biomolecular sequences (Waterman, 1995 [144]; Baldi & Brunak, 1998 [4]; Wong, 2000 [153]).

2.2.1 Statistical Pattern Recognition of Biomolecular Sequence Analysis

Biomolecular sequence analysis is a multidisciplinary subject. It includes mathematical, statistical and computational analyses of DNA, RNA (ribonucleic acid) and protein sequences. Methods for it have roots in areas of combinatorics, speech recognition, linguistics, and statistical modelling. Although most of the problems in biomolecular sequence analysis have generally been thought of and treated as computational, they are essentially statistical. For instance, one of the key problems in biomolecular sequence analysis — the pairwise or multiple sequence alignment problem — has been worked on through dynamic programming techniques. It has generally been treated as a computational optimization/maximization problem with an objective function being the summation of scores assigned to the gaps, matches, and mismatches in an alignment. Mostly, the choice of these scores is somewhat arbitrary, for example, a score of $+1$ is assigned to each match, a score of -1 is assigned to each mismatch, and a score of -2 is assigned to each gap. Aligning sequences with algorithms based on arbitrary scoring schemes often makes little or no reference to the underlying biological process (Thorne & Churchill, 1995 [129]; Waterman, 1995 [144]). As stated by Durbin, Eddy, Krogh, and Mitchison, the development of more sensitive scoring schemes and the evaluation of the significance of alignment scores are more the realm of statistics than computer science (Durbin et al., 1998 [45]).

Probability theory and statistics have contributed significantly to a lot of problems in biomolecular sequence analysis. In particular, research on the distribution theory of runs and patterns has been adopted for pattern recognition in biomolecular sequences. Runs and runs-related statistics are natural statistical descriptions of patterns when we are interested in analyzing sequence data. Knowing the distributions of specific runs and runs-related statistics, we take chance into consideration when we are analyzing patterns (e.g. consensus elements) in sequences. Recently, research work on investigating runs and runs-related statistics and their distributions have been shown to be useful for detecting patterns in biomolecular sequences (e.g. Karlin et al., 1992, 1993, 1994, 1996, 1997 [75, 74, 78, 79, 80]; Leung et al., 1994, 1996 1999 [92, 91, 93]; Waterman, 1995 [144, 143]; Fu et al., 1996, 1999 [57, 59]; and Lou, 2000 [99]).

Carefully designed probabilistic models have been proven to be valuable tools for biomolecular sequence analysis. Probabilistic modelling can be used to extract information from complex sequence data and reveal important features to answer specific biological or genetic questions. It provides a consistent way of reasoning in the presence of uncertainty and establishes a framework for scientific inferences. The success of various hidden Markov models (HMMs) in capturing and describing intricate biological structures has demonstrated the usefulness of probabilistic and statistical approaches to pattern recognition in biomolecular sequences. A small selected sample of recent research work on using and modifying HMMs for biomolecular sequence analysis includes publications by Churchill et al., 1992, 1995, 1999 [32, 33, 35]; Baldi et al., 1994, 1996, 1998 [9, 5, 7, 4]; Krogh et al., 1994, 1997, 1998 [85, 84, 82, 83]; Pedersen et al., 1996 [108]; Yada et al., 1996, 1997, 1998 [155, 156, 157]; Burge & Karlin, 1997 [27]; Durbin et al., 1998 [45]; Schmidler et al., 2000 [120]; Wong, 2000 [153]; and Balding et al., 2001 [10].

Since biological systems are inherently complex and many facts at the molecular level are still missing, there are many questions that are unanswered or partly answered. The high degree of uncertainty gives researchers relatively little theory, and drives the need for quantitative research. The increase in the availability of large amounts of data allows researchers to gain knowledge and insights from a *data-mining* approach. Recently, researchers in *articial intelligence* have been actively developing new pattern recognition techniques for biomolecular sequence analysis from a *machine-learning* perspective (Baldi & Brunak, 1998 [4]). The basic idea of machine-learning is to learn the theory from the data through a model fitting process that is constantly iteratively updated. Their philosophy is, in fact, similar to the *Bayesian* school of thinking in statistics. Their goal is to extract important information by building probabilistic models from a Bayesian point of view. Although the Bayesian approach has been getting more and more appealing due to recent advances in computer power, the *Maximum Likelihood* approach is still considered as the "yardstick" in the statistics (especially the *classical statistics*) community. Nevertheless, they are two mainstreams in statistics, and both have been adopted for biomolecular sequence analyses. A lot of complex statistical issues in estimating model parameters, testing hypotheses, and comparing various classes of models have arisen, and the need for statistical research has greatly increased. Biomolecular sequence analysis has created a lot of challenging problems and opportunities for *statistical pattern recognition* research.

2.3 Recognition of Patterns in Genomic DNA Sequences

2.3.1 DNA Sequence, Structure and Function

A DNA molecule consists of two strands of *nucleotides.* A nucleotide is a unit which has three subunits: a phosphate group, a sugar (deoxyribose) molecule with five carbon atoms in a ring form, and a nitrogenous base. There are four different nitrogenous bases: *Adenine* (A), *Cytosine* (C), *Guanine* (G), and *Thymine* (T). These nitrogenous bases are of two types: *purines* (A and G), and *pyrimidines* (C and T). Each nucleotide is characterized by the nitrogenous base it carries. The locations of the carbon atoms in the sugar molecule are labelled as $1'$ to $5'$ in order to define the orientation of the molecule. A chain of nucleotides (i.e. a *polynucleotide*) is formed by linking the sugar molecule of one nucleotide to the phosphate group of another nucleotide. Specifically, a $3'$-$5'$ *phosphodiester link* is formed with a phosphate molecule bridging between the $3'$-OH group on the sugar molecule of one nucleotide to the $5'$-OH group on the sugar molecule of the next nucleotide. Thus, a polynucleotide has a *sugar-phosphate backbone* which consists of $3'$-$5'$ *phosphodiester linkages.* One end of a polynucleotide has a $5'$-OH group on the sugar molecule of a nucleotide which is termed the $5'$ end, and the other end has a $3'$-OH group on the sugar molecule of a nucleotide which is termed the $3'$ end. By convention, a single-stranded DNA is always specified in the $5'$ to $3'$ orientation (Refer to Figure 2.1) (Alberts et al., 1994 [1]).

The two strands of nucleotides of a DNA molecule have opposite *polarity* (i.e. they run in opposite directions), and they are described as *antiparallel.* Two antiparallel strands of nucleotides are held by *hydrogen bonds* between complementary nitrogenous bases. The *complementary base pairing* refers to the fact that the nitrogenous base A pairs with the nitrogenous base T, and the nitrogenous base C pairs with the nitrogenous base G (Refer to Figure 2.1). A complete double-stranded DNA

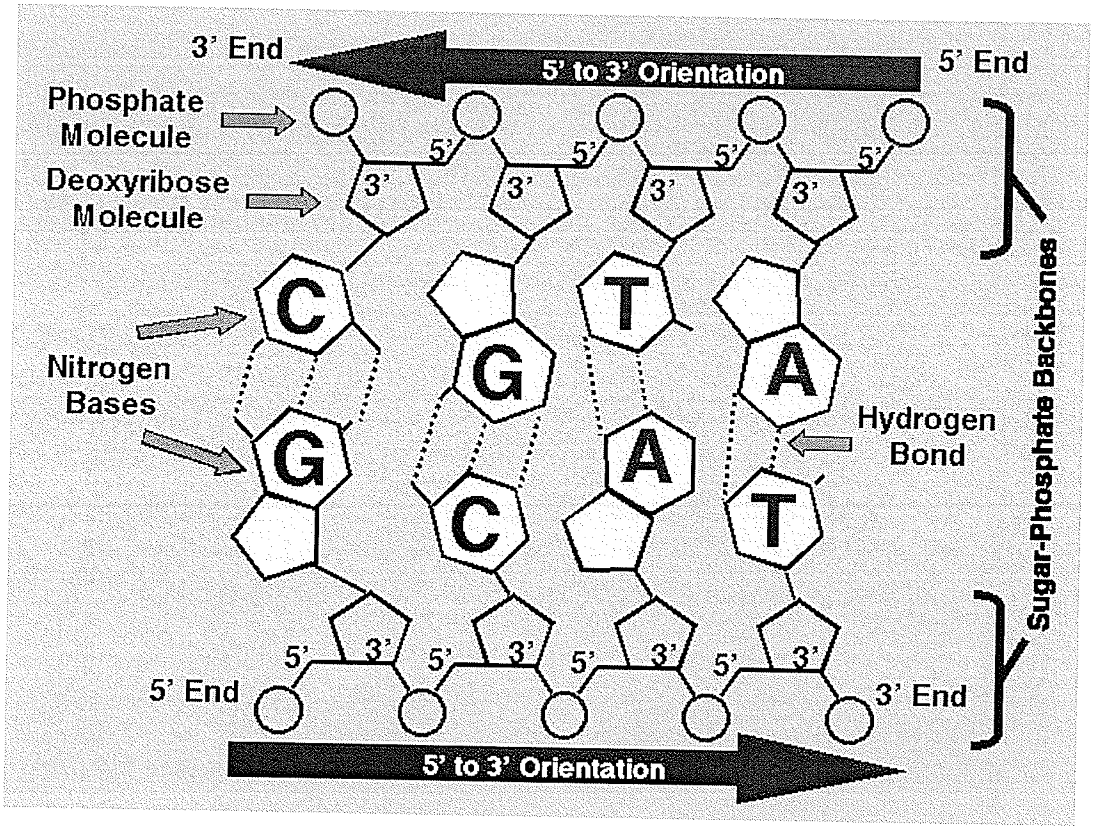


Figure 2.1: Schematic Overview of the Arrangement of Nucleotides of a DNA Segment (Diagram was inspired from *Interdepartmental Course 36.724* notes (Murphy, 1998 [106]))

molecule exists in a three-dimensional *double-helix* structure with the bases projected inwards, and the sugar-phosphate backbones exposed to the outside of the helix. This structure shields all the important atoms of the bases, the genetic information, from physical and/or chemical attacks by the environment. *Base stacking interactions* force the outer surface of a DNA molecule to form two grooves, a wide *major groove* and a narrow *minor groove* which facilitate sequence-specific DNA-protein binding and/or interactions (Refer to Figure 2.2) (Sinden, 1994 [123]; Elliott & Elliott, 1997 [51]).

Historically, DNA structures were initially derived from *X-ray diffraction* patterns of DNA fibers at a high relative humidity condition. In the early 1950's, Maurice Wilkins and Rosalind Franklin had shown that DNA fibers exist in two principal forms, one at a low (65-75%) relative humidity and the other at a high (92%) relative humidity (Dickerson, 1992 [43]). In 1953, James Watson and

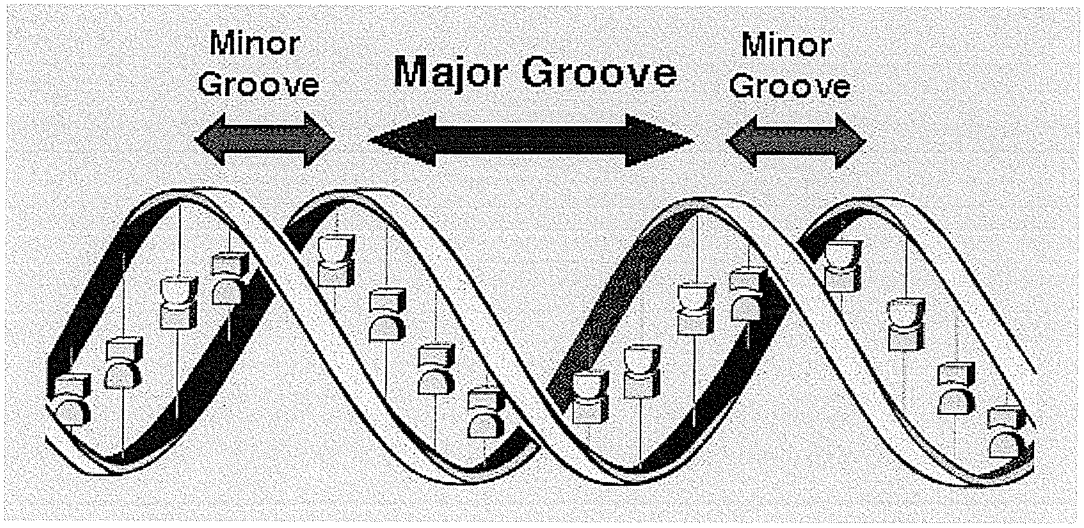


Figure 2.2: Illustration of Major and Minor Grooves of the DNA Double-Helix Structure (Double-helix image was downloaded from http://academy.d20.co.edu/kadets/lundberg/dna.html)

Francis Crick first used X-ray diffraction studies of Wilkins and Franklin to deduce a model for the structure of DNA: the *double-helix model.* In the early 1960's, two structural forms — the *A-form* structure and the *B-form* structure — of DNA were identified by X-ray diffraction analyses of the sodium salt of DNA fibers at 75% relative humidity (Fuller et al., 1965 [60]) and at 92% relative humidity (Langridge et al., 1960 [88, 87]) respectively. Subsequently, two names — the A-form DNA and the B-form DNA, or simply the A-DNA and B-DNA — were given to these structural forms of DNA, and the classic double-helix structure described by Watson and Crick was referred to as the B-DNA. In 1979, *X-ray crystallographic techniques* allowed scientists to study DNA structures at a high resolution. It helped reveal a number of major sequence-dependent structural features that could not be observed in DNA fiber diffraction studies. Furthermore, NMR (Nuclear Magnetic Resonance) methods enable DNA structures to be determined in solution rather than in their solid crystallized states, and they provide useful information for studying DNA structure and dynamics (Bates & Maxwell, 1993 [11]; Neidle, 1994 [107]).

Within the last two decades our understanding of the biochemical and biophysical aspects of DNA molecule has grown tremendously. One of the major structural features of a DNA molecule is its considerable degree of dependence on the sequence context of the nucleotide involved. It is now known that dosDNA (defined, ordered sequence DNA) including *inverted repeats*, *mirror repeats*, *direct repeats*, and *homopurine-homopyrimidine elements* can form a number of alternative DNA structures other than the A-DNA and the B-DNA. Some of these alternative structures are termed C-DNA, D-DNA, H-DNA, S-DNA, T-DNA, and Z-DNA. Experimental work has revealed that DNA can exhibit a diverse range of conformational states, and two of the most striking manifestations are the phenomena of *intrinsic curvature of DNA* and *DNA bending or flexibility* (Sinden, 1994 [123] & Sinden et al., 1998 [124]).

> ***Intrinsic curvature of DNA*** refers to the deformation of the DNA helix axis arising from the preferred conformation of particular DNA sequences.
>
> ***DNA bending or flexibility*** refers to the ease with which certain DNA sequences can be bent, for example by being wrapped around a protein.
>
> (Bates & Maxwell, 1993 [11])

There are many biological processes that involve certain regions of DNA to be flexible enough for bending and/or looping, and one of such processes is the *transcriptional control of gene expression* (Murphy, 1998 [105]).

2.3.2 Transcriptional Control of Gene Expression in Eukaryotes

In brief, *gene expression* can be conceptualized by the revised *central dogma* of molecular biology, i.e. flows of genetic information in all forms of life (Refer to Figure 2.3).

Gene expression is a regulated process that proceeds through a series of different control levels before the genetic information is released as a final product. Within each control level, there are many intermediate fine-tuning stages which may modify the information. The relative predominance of regulation at each stage and/or control level differs between *prokaryotes* and *eukaryotes* (Twyman, 1998 [131]).

Technically, a eukaryote is a cell or an organism with membrane-bound, structurally discrete

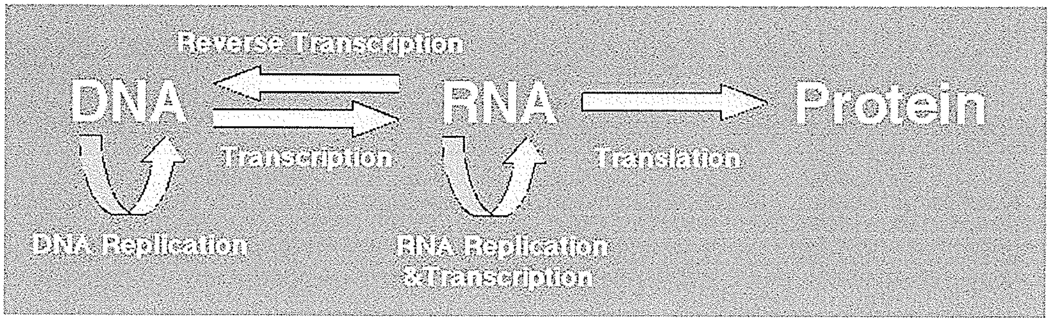


Figure 2.3: Schematic Overview of the Revised Central Dogma in Molecular Biology (Diagram was reproduced from *Advanced Molecular Biology: A Concise Reference* by Twyman (Twyman, 1998 [131]))

nucleus and other well-developed subcellular compartments. Eukaryotes include all forms of life except viruses, bacteria, and blue-green algae (Lewin, 1997 [94]). The genome of a eukaryotic cell contains all the information for making many different RNA molecules and proteins. When a protein-encoding gene on a DNA sequence is expressed in a cell, the corresponding protein product(s) is(are) produced. A eukaryotic cell can control the proteins it makes by regulating the passage of information at any of the many different stages and/or control levels along the pathway from DNA to RNA to protein (Refer to Figure 2.4) (Murphy, 1998 [105]).

For most genes the level of *transcriptional control* is considered to be the most important point of control. It controls when and how often a given gene is transcribed. It is the first level of control in gene expression, and it sets off the initiation event of DNA *transcription* of the gene. In eukaryotes, there are three enzymes called RNA polymerase I, RNA polymerase II, and RNA polymerase III that are responsible for DNA transcription of different sets of genes. RNA polymerase II transcribes all protein-encoding genes. However, numerous proteins are involved and crucial in assisting RNA polymerase II in initiating DNA transcription. The multiprotein complex so-called PIC (pre-initiation complex), which includes the enzyme RNA polymerase II and special proteins so-called GTFs (general transcription factors), recognizes and binds to certain sites or regions of DNA sequences to initiate transcription. One GTF called TFIID is known to be involved in site-specific DNA binding, and several other GTFs are known to be in close contact with the DNA

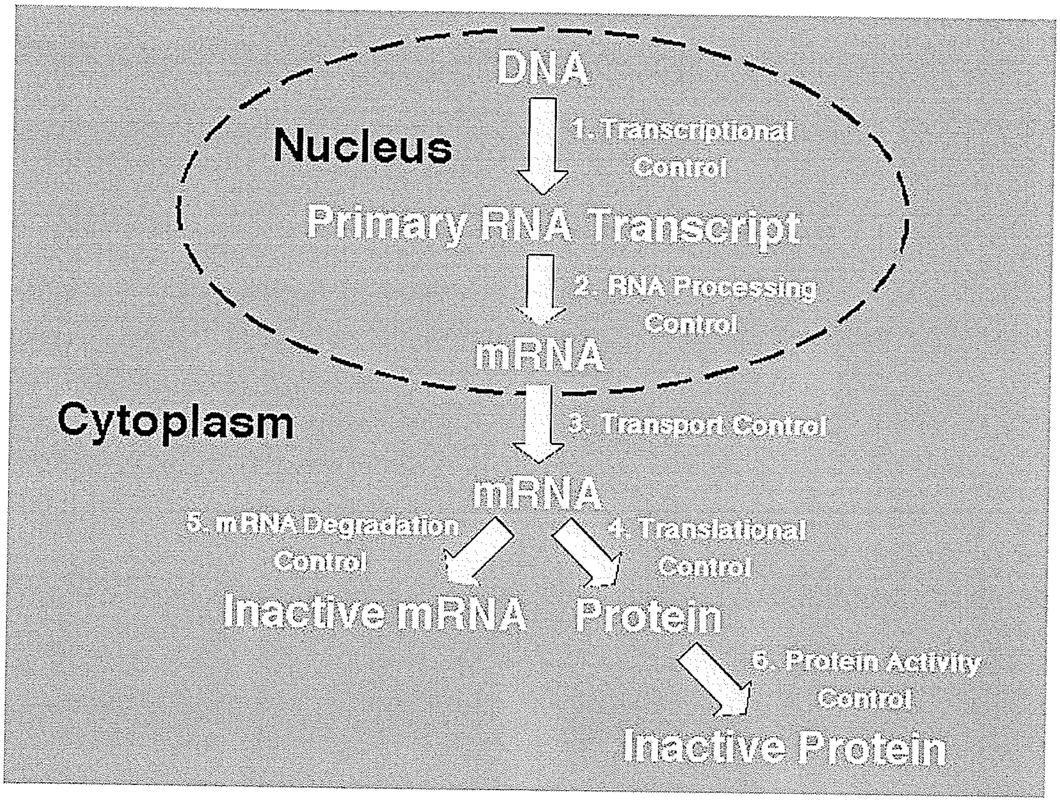


Figure 2.4: Schematic Overview of Different Levels of Control for the Expression of a Protein-Encoding Gene in Eukaryotes (Diagram was modified from *Interdepartmental Course 36.724* notes (Murphy, 1998 [105]))

during initiation of transcription (Fickett & Hatzigeorgiou, 1997 [53]; Pedersen et al., 1999 [110]).

2.3.3 Recognition of Promoters of Eukaryotic RNA Polymerase II Transcribed Genes

In 1977, molecular biologists discovered that most eukaryotic genes have their DNA sequences interrupted by intervening sequences termed *introns.* They called segments of a gene which are decoded to give a RNA product as *exons* (Frank-Kamenetškii, 1997 [54]). This discovery raised a lot of biochemical issues regarding specifying not only the structure of a gene, but also related sequence regions of a gene. Now, the complex series of biochemical mechanisms that control initiation of transcription of a eukaryotic RNA polymerase II transcribed gene are under intense investigation.

We know that certain sites or regions of DNA sequences in the genome have a regulatory role to play in controlling initiation of transcription. These regions are referred to as transcriptional regulatory regions. Generally, they define which DNA strand will be transcribed, and ensure accurate and efficient initiation of transcription. Transcriptional regulatory regions are classified into three groups of elements termed basal promoter elements, UPEs (**u**pstream **p**romoter **e**lements), and enhancers/silencers (Fickett & Hatzigeorgiou, 1997 [53]; Pedersen et al., 1999 [110]). Since basal promoter elements and UPEs are usually found in genes that are constitutively expressed (so-called "housekeeping" genes), they are usually referred to as *promoters* of a gene. Also, the region where the promoters of a gene are located is often referred to as the *promoter region* of the gene (Refer to Figure 2.5) (Murphy, 1998 [105]).

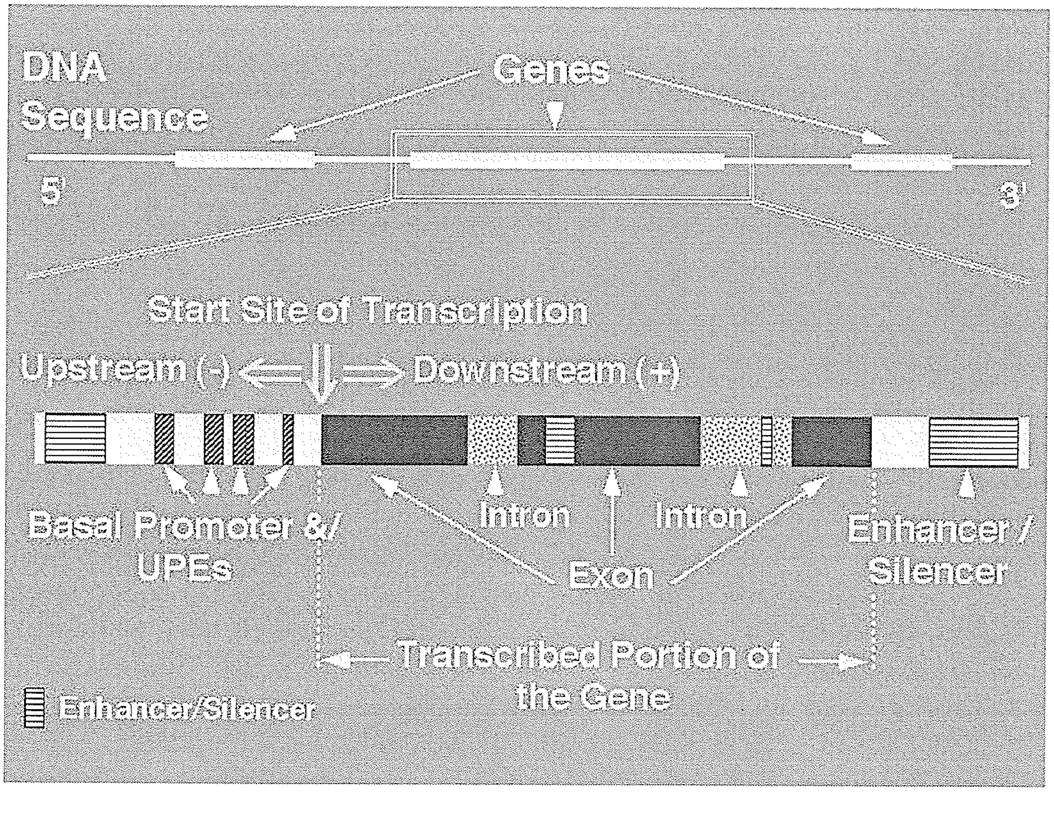


Figure 2.5: Schematic Overview of the Structure of a Eukaryotic Gene (Diagram was modified from *Interdepartmental Course 36.724* notes (Murphy, 1998 [106]))

Basal promoter elements are located "before", or "*upstream*" (upstream elements are also annotated with "−" symbols by molecular biologists) of, the *start point of transcription.* They determine the site of initiation of transcription. Basal promoter elements are a short highly conserved TA rich region termed a TATA-box, and/or a loosely conserved region termed an Inr (**in**itiator **r**egion), and/or CG rich regions (such as CpG islands). Upstream promoter elements (UPEs) are also called promoter-proximal elements, and they are located upstream of the start point of transcription. They increase the rate of transcription by increasing the frequency of initiation. Enhancers/silencers are also called promoter-distal elements, and they may be located either upstream or *downstream* (downstream elements are also annotated with "+" symbols by molecular biologists) of the start point of transcription. They are used for greater modulation of transcriptional rate. They can act over considerable distances (i.e. up to several thousands base pairs from the start point of transcription) and function in either orientation. They are often subject to hormonal control, developmental control and/or tissue specificity (Lewin, 1997 [94]; Murphy, 1998 [105]; Davie, 1999 [40]).

In eukaryotes, most TATA-box containing promoters have the consensus TATAAAA sequence motif centered about 25 base pairs upstream of the start point of transcription. Some Inr containing promoters have the consensus YYCAYYYYY sequence motif, where Y is a pyrimidine (i.e. C or T) (Pedersen et al., 1996 [108]). In addition to TATA-box and/or Inr, there are many CG rich regions also located upstream of the start point of transcription of many genes. CpG islands are frequently found in animal genomes, and CpNpG islands are found in plant genomes, where N can be any nitrogenous base. Some UPEs of RNA polymerase II transcribed genes that are common in mammals are the consensus GGCCAATCT sequence motif termed a CAAT-box and the consensus GGGCGG sequence motif termed a GC-box (Lewin, 1997 [94]; Smith et al., 1997 [125]).

Recognizing promoters of a eukaryotic RNA polymerase II transcribed gene can help improve gene identification task of a protein-encoding gene. More importantly, it also provides us more knowledge in the area of gene expression at the level of transcriptional control. However, the problem of recognizing promoters of a eukaryotic RNA polymerase II transcribed gene is very difficult because the variable distance between various DNA elements for the recognition by different GTFs, and a large number of other proteins are involved in the regulation process (Pedersen et al., 1996, 1998, 1999 [108, 109, 110]). Computer software for recognition of patterns are now routinely used by genome sequencing laboratories to help locate genes and identify different signals in new sequences. With more powerful computational facilities, the use of carefully designed statistical models will become increasingly important for improving existing tools in bioinformatics. Effective use of statistical methods will extend the horizons of genome research.

Chapter 3

PROBABILISTIC MODELS FOR DNA: THE UNIVARIATE PORTRAYAL

3.1 Development and Recent Trends

A DNA molecule consists of two strands of nucleotides, and each nucleotide is characterized by the nitrogenous base it carries. By the complementary base pairing, the two strands of nucleotides are complementary to each other. In most mathematical/statistical/computational studies of DNA, a double-stranded DNA molecule is treated as a single-stranded sequence of nucleotides, and it is represented as an ordered string of its nitrogenous bases: *Adenine*, *Cytosine*, *Guanine*, and *Thymine*. The four nitrogenous bases are often symbolized by the letters A, C, G, and T. For a given integer L and a finite index set $\Gamma_L = \{1, \ldots, L\}$, a DNA sequence of length L has been viewed as a stochastic process $\{Y_t : t \in \Gamma_L\}$ with an underlying probabilistic structure. The random variable Y_t takes an element in the alphabet set $\{A, C, G, T\}$, and corresponds to the nitrogenous base at the t-th position of the DNA sequence. Various classes of models have been designed in the past to model

DNA sequences. Since our focus is on probabilistic models with a hidden Markov model structure, we only briefly mention other classes of probabilistic models in the literature.

One of the simplest probabilistic structures is the independent and identically distributed (IID) model structure. The nucleotides of a DNA sequence are assumed to have their nitrogenous bases occurring independently with probabilities P_A, P_C, P_G and P_T. This IID model structure is subject to a contraint $\sum_{y_t \in \{A,C,G,T\}} P_{y_t} = 1$. Thus, there are $4 - 1 = 3$ free parameters. Then, the probability of a particular sequence of length L is given by:

$$Pr(Y_{[1,L]} = y_{[1,L]}) = \prod_{t=1}^{L} Pr(Y_t = y_t) = \prod_{t=1}^{L} P_{y_t}, \quad \text{for every} \quad y_t \in \{A, C, G, T\}. \tag{3.1}$$

When we allow the probabilities of the nitrogenous bases to change with the sequence position index t, we have the more general independent and non-identically distributed (INID) model structure. The nucleotides of a DNA sequence are then assumed to have their nitrogenous bases occurring independently with probabilities $P_A(t), P_C(t), P_G(t)$ and $P_T(t)$. This INID model structure is subject to contraints $\sum_{y_t \in \{A,C,G,T\}} P_{y_t}(t) = 1$, $\forall t \in \Gamma_L$. Thus, there are $L \times (4 - 1) = 3L$ free parameters. Similar to formula (3.1), the probability of a particular sequence of length L is given by:

$$Pr(Y_{[1,L]} = y_{[1,L]}) = \prod_{t=1}^{L} Pr(Y_t = y_t) = \prod_{t=1}^{L} P_{y_t}(t), \quad \text{for every} \quad y_t \in \{A, C, G, T\}. \tag{3.2}$$

When we allow for dependence between neighbouring nitrogenous bases, we often model a DNA sequence with a Markov chain (MC). A first order homogeneous Markov chain with four states has been used to model certain forms of short range dependence in DNA sequences. Such an MC model with its state space $\{A, C, G, T\}$ is characterized by its transition probability matrix:

$$\mathcal{A}_Y = \begin{array}{c|cccc} Y_{t-1}\backslash Y_t & A & C & G & T \\ \hline A & P_{AA} & P_{AC} & P_{AG} & P_{AT} \\ C & P_{CA} & P_{CC} & P_{CG} & P_{CT} \\ G & P_{GA} & P_{GC} & P_{GG} & P_{GT} \\ T & P_{TA} & P_{TC} & P_{TG} & P_{TT} \end{array},$$

where $\mathcal{A}_Y$ is subject to the contraints $\sum_{j\in\{A,C,G,T\}} P_{ij} = 1$, $\forall i \in \{A, C, G, T\}$. Thus, there are $4 \times 4 - 4 = 12$ free parameters. The probability of a particular sequence of length L is given by:

$$\begin{aligned} Pr(Y_{[1,L]} = y_{[1,L]}) &= Pr(Y_1 = y_1)Pr(Y_2 = y_2|y_1)\cdots Pr(Y_{L-1} = y_{L-1}|y_{L-2})Pr(Y_L = y_L|y_{L-1}) \\ &= Pr(Y_1 = y_1)\prod_{t=2}^{L} P_{y_{t-1}y_t}, \quad \text{for every} \quad y_{t-1}, y_t \in \{A, C, G, T\}. \end{aligned} \tag{3.3}$$

There are also non-homogeneous Markov chain models. A first order non-homogeneous Markov chain with its state space $\{A, C, G, T\}$ is characterized by its transition probability matrices:

$$\mathcal{A}_Y(t) = \begin{array}{c|cccc} Y_{t-1}\backslash Y_t & A & C & G & T \\ \hline A & P_{AA}(t) & P_{AC}(t) & P_{AG}(t) & P_{AT}(t) \\ C & P_{CA}(t) & P_{CC}(t) & P_{CG}(t) & P_{CT}(t) \\ G & P_{GA}(t) & P_{GC}(t) & P_{GG}(t) & P_{GT}(t) \\ T & P_{TA}(t) & P_{TC}(t) & P_{TG}(t) & P_{TT}(t) \end{array}$$

where $\mathcal{A}_Y(t)$ is subject to the contraints $\sum_{j\in\{A,C,G,T\}} P_{ij}(t) = 1$, $\forall i \in \{A, C, G, T\}$ & $\forall t \in \Gamma_L$. Thus, there are $L \times (4 \times 4 - 4) = 12L$ free parameters. Similar to formula (3.3), the probability of a particular sequence of length L is given by:

$$Pr(Y_{[1,L]} = y_{[1,L]}) = Pr(Y_1 = y_1) \prod_{t=2}^{L} P_{y_{t-1}y_t}(t), \quad \text{for every} \quad y_{t-1}, y_t \in \{A, C, G, T\}. \tag{3.4}$$

It is worth mentioning that an order zero homogeneous MC model is an IID model, and an order zero non-homogeneous MC model is an INID model. Higher order homogeneous and non-homogeneous MC models have also been proposed for modelling DNA sequences. For example, a fifth order homogeneous MC has been implemented for modelling non-coding regions of DNA and a fifth order non-homogeneous MC has been implemented for modelling coding regions of DNA in several recent gene finding systems (Burge & Karlin, 1998 [28]).

In recent years, two particular classes of probabilistic models — hidden Markov models (HMMs) and artificial neural networks (ANNs) — have made considerable progress in genomic DNA sequence pattern recognition and signal prediction. They have attracted a great deal of research attention in many disciplines. A non-exhaustive list of good references* on HMMs and ANNs includes publications by Churchill, 1992, 1995 [32, 33]; Presnell & Cohen, 1993 [115]; Snyder & Stormo, 1993 [126]; Baldi et al., 1994 [5, 9]; Krogh et al., 1994 [85]; Rawlings & Fox, 1994 [117]; Eddy, 1995, 1996, 1997, 1998 [46, 47, 48, 49]; Pedersen et al., 1995, 1996, 1997, 1998 [111, 108, 113, 109]; Felsenstein & Churchill, 1996 [52]; Hatzigeorgiou et al., 1996 [64]; Kulp et al., 1996 [86]; Burge & Karlin, 1997, 1998 [27, 28]; Crowley et al., 1997 [38]; Krogh, 1997 [82]; Wu, 1997 [154]; Yada et al., 1997, 1998 [156, 157]; Amitar, 1998 [2]; Baldi & Brunak, 1998 [4]; Durbin et al., 1998 [45]; Churchill & Lazareva, 1999 [35]; and Balding et al., 2001.

With surprisingly good performances, probabilistic models in the forms of hidden Markov models (HMMs), artificial neural networks (ANNs) and their variants have become very popular in DNA sequence analysis. Among the modifications and extensions of HMMs and ANNs, a recent modification of the HMM is to implement a semi-Markov process rather than a Markov process to create a hidden semi-Markov model (HSMM). A computer program named *GENSCAN*, which is based on an HSMM structure developed by Burge and Karlin (Burge & Karlin, 1997 [27]), has be-

*NOTE: The selection of these references is completely subject to my personal preference.

come one of the best performing computational gene-finders in recent literature. Moreover, another recent scheme incorporates an HMM with an ANN to create an HMM/ANN hybrid model, which was named a generalized HMM (Kulp et al., 1996 [86]) or a hidden neural network (Riis & Krogh, 1997 [119]). A computer program named *Genie*, which is based on a generalized HMM structure developed by Kulp, Haussler, Reese and Eeckman (Kulp et al., 1996 [86]; Reese et al., 1997 [118]), has also been relatively successful for recognition of human genes in DNA sequences.

Furthermore, with the rising increase in the use of Bayesian methods due to recent advances in computer power, creating hybrid models from a Bayesian point of view becomes a leading-edge strategy. With carefully designed model structures, these hybrid models are believed to have great potential for further improvements on the recognition of promoter regions of eukaryotic RNA polymerase II transcribed genes (Pedersen et al., 1996 [108]; Baldi & Brunak, 1998 [4]). However, this new strategy has also created other new problems, such as the control for *over-fitting* in the ANN component and the justification for the specific choice of a *prior distribution* in the Bayesian framework, which are still very difficult to deal with and highly debatable.

Having reviewed the literature on probabilistic models for pattern recognition in biomolecular sequences (particularly DNA sequences), I realized that almost all of these models are created or designed to capture properties of biomolecules based on their compositional content (for example, the C+G content of DNA sequences) from a *"one-dimensional perspective"*. From a statistical or probabilistic point of view, these models are *univariate models*. Having learned that real DNA, RNA, and protein molecules all possess many different aspects beyond their compositional content, I have decided to create a new generation of probabilistic models for biomolecules from a *"multi-dimensional perspective"*. In statistical terminology, these models are *multivariate models*. Multivariate models can be used to capture multiple aspects of biomolecules simultaneously. In other words, it is a multivariate portrayal of biomolecules rather than the univariate portrayal. My research work on this new class of probabilistic models is presented in Chapter 5: "Probabilistic Models for DNA: A Multivariate Portrayal". Since both the investigation of runs statistics and the creation of the new multivariate probabilistic models require knowledge of the *univariate* hidden Markov models

(HMMs), an introduction on the typical HMMs is given in the next section so as to familiarize readers with the notation and terminology as well as the basic problems associated with HMMs and their common solutions and algorithms.

3.2 Modelling DNA by Hidden Markov Models (HMMs): A Closer Look

3.2.1 Origin, Model Definition and Notation

Hidden Markov models were originally introduced and studied by Baum and Petrie in 1966 (Baum & Petrie, 1966 [12]). In 1975, the theory of HMMs was first implemented for speech processing applications by Baker at the Carnegie Mellon University, and by Jelinek and his colleagues at the IBM Corporation. Although the basic theory of HMMs was published, widespread understanding and application occurred only in the late 1980's (Rabiner, 1989 [116]). A majority of applications of HMMs have been in machine speech and character recognition. However, after Churchill[†] pioneered the use of HMMs to model DNA sequences (Churchill, 1989 [31]), HMMs have become very popular for pattern recognition in biomolecular sequence analysis.

A typical hidden Markov model (HMM) has been defined as a double stochastic process because of its "double layering" structure. One layer is an underlying unobservable/hidden discrete stochastic process which is assumed to follow a first order Markov chain, and is denoted as the *state process* or $\{X_t,\ t = 1, 2, \ldots\}$ of the model. The other layer is an observable stochastic process that is conditionally independent given the current state of the hidden Markov chain, and it is denoted as the *outcome process* or $\{Y_t,\ t = 1, 2, \ldots\}$ of the model. Specifically, for an HMM of finite length L (Refer to Figure 3.1), an outcome random variable (Y_t) follows a probability distribution determined by the current state ($X_t = x_t$) of the hidden Markov chain. Therefore, an HMM of finite length can

[†] Dr. Gary A. Churchill (Ph.D. in Biostatistics) is currently a staff scientist, who is leading the Statistical Genetics Group, at the Jackson Laboratory in Bar Harbor, U.S.A.

be viewed as a generative system which eventually produces sequences of observations or outcomes. Owing to this special structure, they are also referred to as statistical models of Markov sources or probabilistic functions of Markov chains in the communications literature (Rabiner, 1989 [116]). Although most applications of HMMs in the literature are to modelling *discrete outcomes*, with minor modifications, the HMM theory can also be applied to model *continuous outcomes* (Churchill, 1998 [34]). A rather thorough account of the typical HMMs for modelling discrete outcomes (e.g. DNA base-compositional outcomes) is provided below.

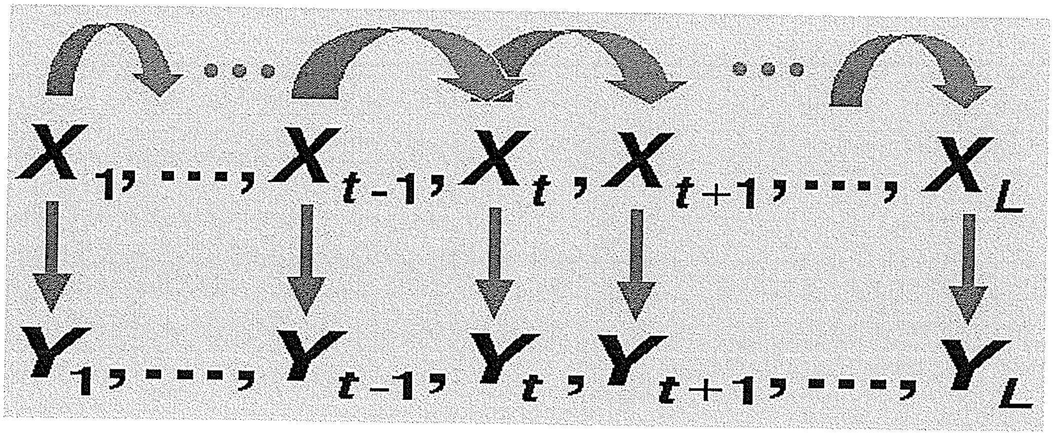

$X_1, \ldots, X_{t-1}, X_t, X_{t+1}, \ldots, X_L$
$Y_1, \ldots, Y_{t-1}, Y_t, Y_{t+1}, \ldots, Y_L$

Figure 3.1: Graphical Representation of the Dependence Structure of a Typical HMM of Length L

Assuming the underlying hidden Markov chain is homogeneous, an HMM of length L with N hidden discrete states and M observable discrete outcomes is fully described by the following elements:

- A set of *hidden discrete states*: $\{1, \ldots, N\}$.
    - Hence, the state process/hidden Markov chain is defined on the state space $\{1, \ldots, N\}$.
- A set of *possible discrete outcomes*: $\{1, \ldots, M\}$.
    - Hence, the outcome random variable takes values in the set $\{1, \ldots, M\}$.
- The *initial state distribution* of the hidden Markov chain, $\pi = (\pi_1, \ldots, \pi_N)$ with

$$\pi_i = Pr(X_1 = i); \qquad i \in \{1, \ldots, N\}. \tag{3.5}$$

- The *transition probability matrix*, $\mathcal{A}_X = [a_{ij}]$, of the hidden Markov chain with

$$a_{ij} = Pr(X_{t+1} = j | X_t = i); \qquad i, j \in \{1, \ldots, N\}. \tag{3.6}$$

- The *outcome-emission distribution* of Y_t conditioning on $X_t = j$ or the *outcome-emission probability matrix*, $\mathcal{B}_{Y|X} = [b_{j(k)}]$, with

$$b_{j(k)} = Pr(Y_t = k | X_t = j); \qquad j \in \{1, \ldots, N\}, k \in \{1, \ldots, M\}. \tag{3.7}$$

Since both $\mathcal{A}_X$ and $\mathcal{B}_{Y|X}$ are stochastic matrices, all their entries are non-negative numbers and their row sums are equal to one (i.e. $\mathcal{A}_X$ is subject to the contraints $\sum_{j \in \{1,\ldots,N\}} a_{ij} = 1$, $\forall i \in \{1, \ldots, N\}$. $\mathcal{B}_{Y|X}$ is subject to the contraints $\sum_{k \in \{1,\ldots,M\}} b_{j(k)} = 1$, $\forall j \in \{1, \ldots, N\}$). When every hidden state can be reached in a single step from every other hidden state (i.e. $a_{ij} > 0$, $\forall i, j \in \{1, \ldots, N\}$), we have a "fully connected" HMM. In general, when all hidden states are *recurrent*, the HMM is described as having a *recurrent architecture*. Clearly, a fully connected HMM has a recurrent architecture. Another HMM architecture is called a *left-to-right architecture*. All hidden states are *transient* under a left-to-right architecture. The "mutation-deletion-insertion" (MDI) model for multiple DNA (or protein) sequence alignment is an example of an HMM with a left-to-right architecture. The design of an architecture is often driven by the practical application, so other HMM architectures with both recurrent and non-recurrent components are also constructed in the literature. Depending on the application, the parameter space of an HMM can be prohibitively large, and additional constraints and assumptions may need to be imposed in practice (Churchill, 1998 [34]).

In a typical HMM, there are at most $N - 1$ free parameters from the initial state distribution, at most $N(N-1)$ free parameters from the transition probability matrix, and at most $N(M-1)$ free parameters from the outcome-emission probability matrix. When a fully connected HMM is considered, the duration in the hidden state i will have a geometric distribution with its mean equals to $\frac{1}{1-a_{ii}}$, where $i \in \{1, \ldots, N\}$ (Rabiner, 1989 [116]). For convenience, the parameters of an HMM are sometimes referred to as $\boldsymbol{\theta}$; where $\boldsymbol{\theta} = (\pi, \mathcal{A}_X, \mathcal{B}_{Y|X})$. The probability of a particular sequence of length L under the HMM model structure is given by:

$$\begin{aligned}
Pr(Y_{[1,L]} = y_{[1,L]}) &= \sum_{x_1=1}^{N} \cdots \sum_{x_L=1}^{N} Pr(Y_{[1,L]} = y_{[1,L]}, X_{[1,L]} = x_{[1,L]}) \\
&= \sum_{x_1=1}^{N} \cdots \sum_{x_L=1}^{N} Pr(Y_{[1,L]} = y_{[1,L]} | x_{[1,L]}) \times Pr(X_{[1,L]} = x_{[1,L]}) \\
&= \sum_{x_1=1}^{N} \cdots \sum_{x_L=1}^{N} b_{x_1(y_1)} \cdot b_{x_2(y_2)} \cdots b_{x_L(y_L)} \times \pi_{x_1} \cdot a_{x_1 x_2} \cdots a_{x_{L-1} x_L} \\
&= \sum_{x_1=1}^{N} \cdots \sum_{x_L=1}^{N} \pi_{x_1} b_{x_1(y_1)} \prod_{t=2}^{L} a_{x_{t-1} x_t} b_{x_t(y_t)}.
\end{aligned} \tag{3.8}$$

The use of HMMs with a recurrent architecture was first introduced to DNA sequence analysis by Churchill (Churchill, 1989, 1992 [31, 32]). In particular, an HMM with binary hidden states (i.e. $X_t \in \{0, 1\}$) and binary observable outcomes (i.e. $Y_t \in \{0, 1\}$) has been used to analyze the C+G richness pattern of certain human DNA (Refer to Figure 3.2 for an illustration).

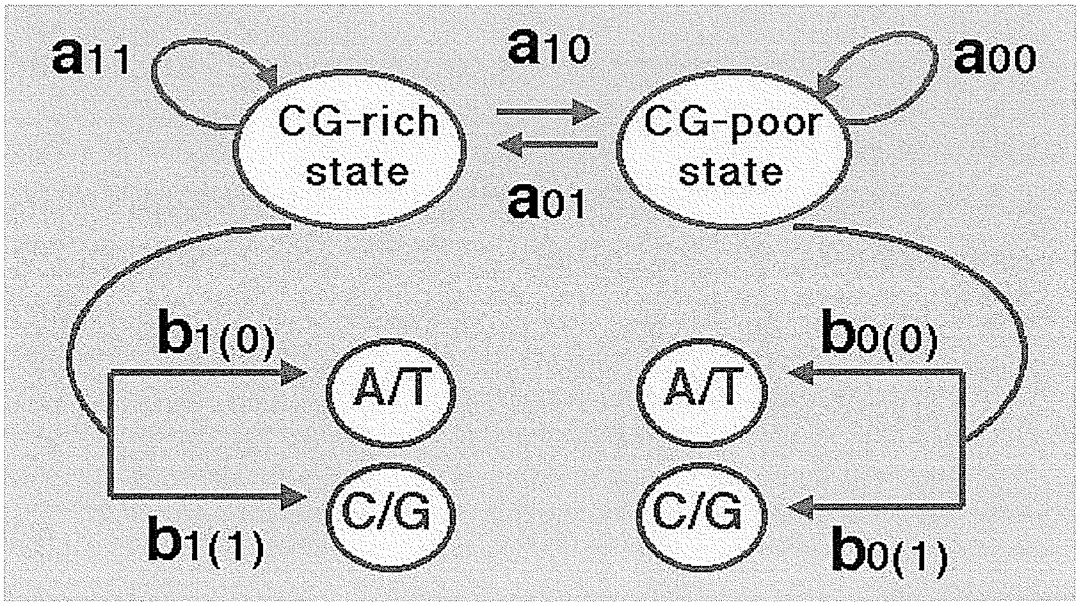


Figure 3.2: A Binary HMM with a Recurrent Architecture for the Analysis of DNA C+G Richness

Briefly, a DNA sequence is represented as a sequence of dichotomous (or binary) outcomes with each outcome being either a "C/G" (1) or a "A/T" (0). The two hidden states of the HMM are defined as the "CG-rich" state (or state 1) and the "CG-poor" state (or state 0). There are at most

$1+2+2=5$ HMM parameters in this structure. In the following chapters, we refer to this simple HMM as a binary HMM structure.

Once an HMM architecture is designed, there are three problems of interest associated with an HMM in pattern recognition. They are the scoring problem, the alignment problem, and the training problem. These three problems are often linked together (Eddy, 1996 [47]). It is worth mentioning that we often need to attack the training problem first in practice. Since the algorithm for solving the training problem is built from components of the algorithms for the other two problems, we first describe the scoring and alignment problems with their common algorithms in the following subsections. Before proceeding to the details of the three HMM problems and their common solutions/algorithms[‡], I would like to explicitly re-emphasize the two key model assumptions under the typical HMM model defined in this subsection.

$\boxed{\textbf{HMM ASSUMPTION 1}}$ (I.e. $X_{[1,L]}$ is a first order Markov chain)

For $t = 1, \ldots, L$;

$$Pr(X_t = x_t | X_{[1,t-1]} = x_{[1,t-1]}, \pi, \mathcal{A}_X) = Pr(X_t = x_t | X_{t-1} = x_{t-1}, \pi, \mathcal{A}_X).$$

$\boxed{\textbf{HMM ASSUMPTION 2}}$ (I.e. $Y_{[1,t-1]}$ & Y_t are conditionally independent given X_t.)

For $t = 1, \ldots, L$;

$$Pr(Y_t = y_t | X_t = x_t, Y_{[1,t-1]} = y_{[1,t-1]}, \mathcal{B}_{Y|X}) = Pr(Y_t = y_t | X_t = x_t, \mathcal{B}_{Y|X}).$$

NOTE: The following is also true under the HMM model structure.

$$Pr(Y_t = y_t | X_{[1,L]} = x_{[1,L]}, Y_{[1,t-1]} = y_{[1,t-1]}, \mathcal{B}_{Y|X}) = Pr(Y_t = y_t | X_t = x_t, \mathcal{B}_{Y|X}).$$

[‡]NOTE: It is noted that direct implementations of these algorithms will result in an underflow problem when L is large (e.g. $L > 100$), so a scaling procedure is also implemented to prevent numerical underflow in this work. Since numerical underflow will become a serious problem easily in the hidden multivariate Markov models (HM^{3}s), technical details on the implementation of the scaling procedure are described in Chapter 5.

3.2.2 The HMM Scoring Problem and Its Common Solution/Algorithm

The scoring problem associated with an HMM is the problem of computing the value of the likelihood function $\mathcal{L}_Y(\boldsymbol{\theta})$ for an observed sequence $y_{[1,L]}$, as a function of the parameters. When $y_{[1,L]}$ is a sequence of discrete outcomes, computing the value of the likelihood function at a specific $\boldsymbol{\theta}$ can be thought of as equivalent to calculating the probability of getting the observed sequence under a fully-specified model (although the likelihood function is only defined up to a multiplicative constant). The probability scoring scheme provides us with a consistent way of reasoning in the presence of uncertainty. It can also help us to choose the "best" model. Multiple summations in expression (3.8) generally make the scoring problem intractable through direct computation. However, an efficient algorithm called the *forward algorithm* has been developed for solving the scoring problem. The key idea behind this procedure is based on rewriting the likelihood function $\mathcal{L}_Y(\boldsymbol{\theta})$ in terms of the *"forward"* probabilities. The forward probabilities are denoted as $\alpha_t(i)$'s; where t is the time/position index (i.e. $t = 1, \ldots, L$), and i is one of the possible discrete states of the hidden Markov chain (i.e. $i \in \{1, \ldots, N\}$).

DEFINITION

For $t = 1, \ldots, L$ and $i \in \{1, \ldots, N\}$;

$$\alpha_t(i) = Pr(Y_1 = y_1, \ldots, Y_t = y_t, X_t = i|\boldsymbol{\theta}) = Pr(Y_{[1,t]} = y_{[1,t]}, X_t = i|\boldsymbol{\theta}).$$

DERIVATION

$$\begin{aligned}
\mathcal{L}_Y(\boldsymbol{\theta}) &= Pr(Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta}) \\
&\qquad \text{(Taking the constant of proportionality to be 1, without loss of generality)} \\
&= \sum_{i=1}^{N} Pr(Y_{[1,L]} = y_{[1,L]}, X_L = i|\boldsymbol{\theta}) \\
&= \sum_{i=1}^{N} \alpha_L(i).
\end{aligned} \tag{3.9}$$

Hence, the scoring problem becomes a problem of computing $\alpha_L(i)$'s, and a neat recursive relationship has been established for the computations of $\alpha_L(i)$'s (Rabiner, 1989 [116]).

- **Algorithmic Solution**: The recursive forward algorithm has been the common solution for the HMM scoring problem. As in all recursive procedures, the forward algorithm consists of an initialization step, a recursion/induction step, and a termination step. These three steps are explicitly shown below.

  1. **Initialization Step:**

     For $i = 1, \ldots, N$;

     $$\begin{aligned}\alpha_1(i) &= Pr(Y_1 = y_1, X_1 = i|\boldsymbol{\theta}) = Pr(X_1 = i|\boldsymbol{\theta})Pr(Y_1 = y_1|X_1 = i, \boldsymbol{\theta}) \\ &= \pi_i b_{i(y_1)}.\end{aligned} \tag{3.10}$$

  2. **Recursion/Induction Step:**

     For $t = 1, \ldots, L-1$ and $i, j = 1, \ldots, N$;

     $$\begin{aligned}\alpha_{t+1}(j) &= Pr(Y_{[1,t+1]} = y_{[1,t+1]}, X_{t+1} = j|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} Pr(Y_{[1,t+1]} = y_{[1,t+1]}, X_t = i, X_{t+1} = j|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} Pr(X_t = i, X_{t+1} = j|\boldsymbol{\theta})Pr(Y_{[1,t+1]} = y_{[1,t+1]}|X_t = i, X_{t+1} = j, \boldsymbol{\theta}) \\ &\qquad\qquad \text{(BY HMM ASSUMPTIONS 1 \& 2)} \\ &= \sum_{i=1}^{N} Pr(X_t = i|\boldsymbol{\theta})a_{ij}Pr(Y_{[1,t]} = y_{[1,t]}|X_t = i, \boldsymbol{\theta})Pr(Y_{t+1} = y_{t+1}|X_{t+1} = j, \boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} Pr(Y_{[1,t]} = y_{[1,t]}, X_t = i|\boldsymbol{\theta})a_{ij}b_{j(y_{t+1})} \\ &= \left(\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right)b_{j(y_{t+1})}.\end{aligned} \tag{3.11}$$

  3. **Termination Step:**

$$\mathcal{L}_Y(\boldsymbol{\theta}) = \sum_{i=1}^{N} \alpha_L(i). \tag{3.12}$$

3.2.3 The HMM Alignment Problem and Its Common Solution/Algorithm

The alignment problem associated with an HMM is the problem of reconstructing estimates of the hidden part of the model, that is the sequence of the hidden discrete states, when we have the observed sequence of outcomes $y_{[1,L]}$. This is often the core matter of interest in a specific application of pattern recognition, and it is also called the *decoding problem* in speech recognition. For example, an HMM alignment problem in DNA sequence analysis would be to reconstruct an estimate of the underlying hidden state sequence so as to reveal the C+G richness pattern along an experimentally uncharacterized observed DNA sequence. Two inspiring papers by Churchill have illustrated different applications of HMMs to reveal different patterns and structures within DNA sequences as well as whole genomes of different organisms (Churchill, 1989, 1992 [31, 32]).

Conventionally, an HMM needs to be fully specified before attacking the alignment problem. Since the parameters of the HMM are usually unknown, estimates (often the maximum likelihood estimates) of them are used to specify the model. There are two general approaches for the alignment problem when the model is fully specified: a *local approach* and a *global approach.* Under the HMM model structure, algorithms called the *backward algorithm* and the *Viterbi algorithm* have been developed for solving the alignment problem locally and globally when the model is fully specified. The key idea behind these procedures is similar in implementation to the forward algorithm, and their similarity will become clear after the explicit details of these algorithms are given.

A Local Approach for the HMM Alignment Problem

A common local approach is to find the most probable hidden state at each time/position t by using the corresponding marginal conditional probability $Pr(X_t = x_t | y_{[1,L]}, \boldsymbol{\theta})$. In other words, it is to find the x_t which will give the highest $Pr(X_t = x_t | y_{[1,L]}, \boldsymbol{\theta})$ at each time/position t. Therefore, this local approach can be viewed as an estimation of X_t in a "pointwise" fashion, and it suggests

$$\widehat{X_t} = \underset{x_t}{\operatorname{argmax}}\ Pr(X_t = x_t | y_{[1,L]}, \boldsymbol{\theta}).$$

The backward algorithm has been commonly used for solving the HMM local alignment problem. The idea behind the backward algorithm is based on rewriting $Pr(X_t = x_t | y_{[1,L]}, \boldsymbol{\theta})$ in terms of the *"forward"* and *"backward"* probabilities. The definition and computations of the forward probabilities have been discussed in the previous subsection on the HMM scoring problem. Now, the definition and computations of the backward probabilities are discussed below. The backward probabilities are denoted as $\beta_t(i)$'s; where t is the time/position index (i.e. $t = 1, \ldots, L$), and i is one of the possible discrete states of the hidden Markov chain (i.e. $i \in \{1, \ldots, N\}$).

DEFINITION

For $t = 1, \ldots, L-1$ and $i \in \{1, \ldots, N\}$;

$$\beta_t(i) = Pr(Y_{t+1} = y_{t+1}, \ldots, Y_L = y_L | X_t = i, \boldsymbol{\theta}) = Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_t = i, \boldsymbol{\theta}).$$

DERIVATION

$$\begin{aligned} Pr(X_t = x_t | y_{[1,L]}, \boldsymbol{\theta}) &= \frac{Pr(Y_{[1,L]} = y_{[1,L]}, X_t = x_t | \boldsymbol{\theta})}{Pr(Y_{[1,L]} = y_{[1,L]} | \boldsymbol{\theta})} \\ &= \frac{Pr(Y_{[1,L]} = y_{[1,L]}, X_t = x_t | \boldsymbol{\theta})}{\mathcal{L}_Y(\boldsymbol{\theta})} \\ &= \frac{Pr(X_t = x_t | \boldsymbol{\theta}) Pr(Y_{[1,L]} = y_{[1,L]} | X_t = x_t, \boldsymbol{\theta})}{\mathcal{L}_Y(\boldsymbol{\theta})} \\ &\qquad \text{(BY HMM ASSUMPTION 2)} \\ &= \frac{Pr(X_t = x_t | \boldsymbol{\theta}) Pr(Y_{[1,t]} = y_{[1,t]} | X_t = x_t, \boldsymbol{\theta}) Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_t = x_t, \boldsymbol{\theta})}{\mathcal{L}_Y(\boldsymbol{\theta})} \\ &= \frac{Pr(Y_{[1,t]} = y_{[1,t]}, X_t = x_t | \boldsymbol{\theta}) Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_t = x_t, \boldsymbol{\theta})}{\mathcal{L}_Y(\boldsymbol{\theta})} \\ &= \frac{\alpha_t(x_t) \beta_t(x_t)}{\mathcal{L}_Y(\boldsymbol{\theta})}. \end{aligned} \tag{3.13}$$

NOTE: $\mathcal{L}_Y(\boldsymbol{\theta})$ can also be rewritten in terms of the $\alpha_t(x_t)$'s and $\beta_t(x_t)$'s.

$$\begin{aligned}\mathcal{L}_Y(\boldsymbol{\theta}) &= Pr(Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} Pr(Y_{[1,L]} = y_{[1,L]}, X_t = i|\boldsymbol{\theta}) \\ &\qquad \text{(Refer to the above derivation)} \\ &= \sum_{i=1}^{N} \alpha_t(i)\beta_t(i).\end{aligned}$$

Hence, we need to use the $\alpha_t(i)$'s calculated from the forward algorithm and compute the $\beta_t(i)$'s in order to complete the backward algorithm.

- **Algorithmic Solution** (Local Approach): The recursive backward algorithm has been the common "local" solution for the HMM alignment problem. It is very similar to the forward algorithm except it is in a somewhat reverse sense. As in the forward algorithm, it consists of an initialization step, a recursion/induction step, and a termination step. These three steps are explicitly shown below.

  1. **Initialization Step:**

     For $i = 1, \ldots, N$;

     $$\beta_L(i) = 1. \tag{3.14}$$

  2. **Recursion/Induction Step:**

For $t = L-1, \ldots, 1$ and $i, j = 1, \ldots, N$;

$$\begin{aligned}
\beta_t(i) &= Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_t = i, \boldsymbol{\theta}) \\
&= \sum_{j=1}^{N} \frac{Pr(Y_{[t+1,L]} = y_{[t+1,L]}, X_t = i, X_{t+1} = j | \boldsymbol{\theta})}{Pr(X_t = i|\boldsymbol{\theta})} \\
&= \sum_{j=1}^{N} \frac{Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_t = i, X_{t+1} = j, \boldsymbol{\theta}) Pr(X_t = i, X_{t+1} = j | \boldsymbol{\theta})}{Pr(X_t = i|\boldsymbol{\theta})} \\
&\qquad \text{(BY HMM ASSUMPTION 2)} \\
&= \sum_{j=1}^{N} Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_{t+1} = j, \boldsymbol{\theta}) \frac{Pr(X_t = i, X_{t+1} = j | \boldsymbol{\theta})}{Pr(X_t = i|\boldsymbol{\theta})} \\
&\qquad \text{(BY HMM ASSUMPTION 1)} \\
&= \sum_{j=1}^{N} Pr(Y_{[t+1,L]} = y_{[t+1,L]} | X_{t+1} = j, \boldsymbol{\theta}) a_{ij} \\
&\qquad \text{(BY HMM ASSUMPTION 2)} \\
&= \sum_{j=1}^{N} Pr(Y_{t+1} = y_{t+1} | X_{t+1} = j, \boldsymbol{\theta}) Pr(Y_{[t+2,L]} = y_{[t+2,L]} | X_{t+1} = j, \boldsymbol{\theta}) a_{ij} \\
&= \sum_{j=1}^{N} b_{j(y_{t+1})} \beta_{t+1}(j) a_{ij}.
\end{aligned} \tag{3.15}$$

3. **Termination Step:**

For $t = 1, \ldots, L$ and $i = 1, \ldots, N$;

We can put the $\alpha_t(i)$ and the $\beta_t(i)$ into expression (3.13), and can find the "pointwise" estimate for each X_t by

$$\begin{aligned}
\widehat{X_t} &= \operatorname*{argmax}_i Pr(X_t = i | y_{[1,L]}, \boldsymbol{\theta}) \\
&= \operatorname*{argmax}_i \frac{\alpha_t(i)\beta_t(i)}{\mathcal{L}_Y(\boldsymbol{\theta})} \\
&\qquad (\because \mathcal{L}_Y(\boldsymbol{\theta}) \text{ is a constant}) \\
&= \operatorname*{argmax}_i \alpha_t(i)\beta_t(i).
\end{aligned} \tag{3.16}$$

A Global Approach for the HMM Alignment Problem

A common global approach is to find the most probable hidden state sequence as a whole by using the full conditional distribution $Pr(X_{[1,L]} = x_{[1,L]} | y_{[1,L]}, \boldsymbol{\theta})$. In other words, it is to find a *path*,

$x_{[1,L]}$, which will give the highest $Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]}, \boldsymbol{\theta})$. This global approach can be viewed as an estimation of $X_{[1,L]}$, and it suggests

$$\widehat{X_{[1,L]}} = \underset{x_{[1,L]}}{\operatorname{argmax}}\, Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]}, \boldsymbol{\theta}).$$

Since $\quad Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]}, \boldsymbol{\theta}) = \dfrac{Pr(X_{[1,L]} = x_{[1,L]}, Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta})}{Pr(Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta})}$,

we could work with the joint distribution instead of the full conditional distribution to find the most likely path. The resulting estimate is

$$\widehat{X_{[1,L]}} = \underset{x_{[1,L]}}{\operatorname{argmax}}\, Pr(X_{[1,L]} = x_{[1,L]}, Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta}).$$

The Viterbi algorithm has been commonly used for solving the HMM global alignment problem. The idea behind the Viterbi algorithm is based on the following factorization of the joint distribution

$$\begin{aligned} Pr(X_{[1,L]} = x_{[1,L]}, Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta}) &= Pr(Y_{[1,L]} = y_{[1,L]}|X_{[1,L]} = x_{[1,L]}, \boldsymbol{\theta}) Pr(X_{[1,L]} = x_{[1,L]}|\boldsymbol{\theta}) \\ &\qquad \text{(By HMM Assumptions 1 \& 2)} \\ &= \left(\prod_{t=1}^{L} b_{x_t(y_t)}\right)\left(\pi_{x_1}\prod_{t=2}^{L} a_{x_{t-1}x_t}\right) \\ &= \pi_{x_1} b_{x_1(y_1)} \prod_{t=2}^{L} a_{x_{t-1}x_t} b_{x_t(y_t)}. \end{aligned}$$

With the factorization result, a recursive relationship can be established and it forms the important part of the Viterbi algorithm. Before providing the explicit details of the Viterbi algorithm, we need to define two quantities. They are denoted by $\delta_t(i)$ and $\psi_t(j)$, where t is the time/position index (i.e. $t = 1, \dots, L$), and i is one of the possible discrete states of the hidden Markov chain (i.e. $i \in \{1, \dots, N\}$).

DEFINITION

For $t = 2, \dots, L$ and $i, j \in \{1, \dots, N\}$;

$$\delta_t(j) = \max_{x_{[1,t-1]}} Pr(X_{[1,t-1]} = x_{[1,t-1]}, X_t = j, Y_{[1,t]} = y_{[1,t]}|\boldsymbol{\theta});$$

and

$$\psi_t(j) = \operatorname*{argmax}_i \Big(\delta_{t-1}(i)a_{ij}\Big).$$

In other words, the quantity $\delta_t(j)$ is the highest probability of a path at time/position t which accounts for the first t observed outcomes and ends in state j; whereas the quantity $\psi_t(j)$ is the state at time/position $t-1$, when the present state is $X_t = j$, which maximizes the joint probability over all past state sequences.

- **Algorithmic Solution** (Global Approach): The Viterbi algorithm, originally developed by Viterbi (Viterbi, 1967 [133]), has been the common "global" solution for the HMM alignment problem. It consists of four steps: an initialization step, a recursion/induction step, a termination step, and a path backtracking step. These steps are explicitly shown below:

  1. **Initialization Step:**

     For $i = 1, \ldots, N$;

     $$\delta_1(i) = Pr(X_1 = i, Y_1 = y_1|\boldsymbol{\theta}) = Pr(X_1 = i|\boldsymbol{\theta})Pr(Y_1 = y_1|X_1 = i, \boldsymbol{\theta}) = \pi_i b_{i(y_1)}. \quad (3.17)$$

  2. **Recursion/Induction Step:**

     For $t = 2, \ldots, L$ and $i, j = 1, \ldots, N$;

     $$\begin{aligned} \delta_t(j) &= \max_{x_{[1,t-1]}} Pr(X_{[1,t-1]} = x_{[1,t-1]}, X_t = j, Y_{[1,t]} = y_{[1,t]}|\boldsymbol{\theta}) \\ &= \max_i \Bigg[\Big(\max_{x_{[1,t-2]}} Pr(X_{[1,t-2]} = x_{[1,t-2]}, X_{t-1} = i, Y_{[1,t-1]} = y_{[1,t-1]}|\boldsymbol{\theta})\Big) \\ &\qquad \times Pr(X_t = j|X_{t-1} = i, \boldsymbol{\theta})Pr(Y_t = y_t|X_t = j, \boldsymbol{\theta})\Bigg] \\ &= \max_i \Big[\delta_{t-1}(i)a_{ij}\Big] b_{j(y_t)}; \end{aligned} \quad (3.18)$$

     and

     $$\psi_t(j) = \operatorname*{argmax}_i \Big(\delta_{t-1}(i)a_{ij}\Big). \quad (3.19)$$

3. **Termination Step:**

   For $i = 1, \dots, N$;

$$\max_{x_{[1,L]}} Pr(X_{[1,L]} = x_{[1,L]}, Y_{[1,L]} = y_{[1,L]}|\boldsymbol{\theta}) = \max_i \delta_L(i); \tag{3.20}$$

   and

$$\widehat{X_L} = \operatorname*{argmax}_i \delta_L(i). \tag{3.21}$$

4. **Path Backtracking Step:**

   For $t = L-1, \dots, 1$;

$$\widehat{X_t} = \psi_{t+1}(\widehat{X_{t+1}}). \tag{3.22}$$

Essentially, the implementation of the recursion/induction step in the Viterbi algorithm is similar to the recursion/induction step in the forward algorithm, except that the summation routine in equation (3.11) on page 35 is replaced by a maximization routine in equation (3.18) above. It is noted that the results from the above local and global approaches for the HMM alignment problem do not necessarily agree with each other.

So far, we have only discussed the HMM alignment problem when the model is fully specified, i.e. we need to know the parameter set $\boldsymbol{\theta}$, or at least an estimate of it in order to use the backward algorithm and the Viterbi algorithm. Recently, the alignment problem associated with an HMM has been viewed from a Bayesian perspective. Using a Bayesian formulation, one can obtain or approximate the posterior distributions $Pr(X_t = x_t|y_{[1,L]})$ and $Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]})$. Various estimates and features of X_t and $X_{[1,L]}$ based on these posterior distributions can then be explored under a Bayesian framework (e.g. Churchill & Lazareva, 1999 [35]). Since there is a strong logical connection between the HMM alignment problem and the HMM training problem under a Bayesian framework, the use of a Bayesian approach for the HMM alignment problem will be discussed in the next subsection.

3.2.4 The HMM Training Problem and Its Common Solution/Algorithm

The training problem associated with an HMM is the problem of estimating the parameter set $\boldsymbol{\theta}$ of the model so as to best account for the observed sequence of outcomes. This is often the most difficult and challenging problem. There are two main approaches for the parameter estimation of an HMM: the *maximum likelihood approach* and the *Bayesian approach.* The maximum likelihood approach is to find the parameter set that maximizes the likelihood function, whereas the Bayesian approach is to make use of the *posterior distribution* of the parameter set to do estimation.

Maximum Likelihood Approach for the HMM Training Problem

Direct maximization of the likelihood function $\mathcal{L}_Y(\boldsymbol{\theta})$ is generally intractable. However, treating the hidden state sequence $X_{[1,L]}$ as "missing" augmented-data, we can apply the *Expectation-Maximization (EM) algorithm* by working with the *augmented-data likelihood function* $\mathcal{L}_{YX}(\boldsymbol{\theta})$ to estimate the parameters of the HMM. Under the HMM structure, the augmented-data likelihood function is:

$$\begin{aligned} \mathcal{L}_{YX}(\boldsymbol{\theta}) &= Pr(Y_{[1,L]} = y_{[1,L]}, X_{[1,L]} = x_{[1,L]} | \boldsymbol{\theta}) \\ &= \pi_{x_1} b_{x_1(y_1)} \prod_{t=2}^{L} a_{x_{t-1}x_t} b_{x_t(y_t)}. \end{aligned} \tag{3.23}$$

Taking the logarithm of the augmented-data likelihood function, we have:

$$\begin{aligned} \log \mathcal{L}_{YX}(\boldsymbol{\theta}) &= \log \pi_{x_1} + \sum_{t=1}^{L} \log b_{x_t(y_t)} + \sum_{t=2}^{L} \log a_{x_{t-1}x_t} \\ &\qquad \text{(Capturing the "missing data" by indicator variables } u_j(t)\text{'s \& } v_{ij}(t)\text{'s.)} \\ &= \sum_{i=1}^{N} u_i(1) \log \pi_i + \sum_{j=1}^{N}\sum_{t=1}^{L} u_j(t) \log b_{j(y_t)} + \sum_{i=1}^{N}\sum_{j=1}^{N} (\log a_{ij}) \sum_{t=2}^{L} v_{ij}(t); \end{aligned} \tag{3.24}$$

$$\text{where for } t = 1, \ldots, L,\ u_j(t) = \begin{cases} 1 & \text{if } x_t = j, \\ 0 & \text{otherwise;} \end{cases}$$

$$\text{and for } t = 2, \ldots, L,\ v_{ij}(t) = \begin{cases} 1 & \text{if } x_{t-1} = i \ \&\ x_t = j, \\ 0 & \text{otherwise.} \end{cases}$$

In many cases, the problem of maximizing $\log \mathcal{L}_{YX}(\boldsymbol{\theta})$ yields a closed-form solution. The maximum likelihood approach has been widely used to deal with the HMM training in different forms of the DNA decoding problem or the problem of locating functional domains and/or genes in DNA (e.g. Churchill, 1992 [32]; Krogh, Mian & Haussler, 1994 [85]; Felsenstein & Churchill, 1996 [52]; Henderson, Salzberg & Fasman, 1997 [65]; and Durbin, Eddy, Krogh & Mitchison, 1998 [45]).

- **Algorithmic Solution** (Maximum Likelihood Approach): The iterative procedure called the *Baum-Welch algorithm* developed by Baum and his colleagues (Baum et al., 1966, 1970 [12, 13]) or equivalently the *EM algorithm* developed by Dempster and his colleagues (Dempster et al., 1977 [42]) has been a common solution for training HMMs. The implementation of the EM algorithm for HMMs involves the use of both the forward algorithm and the backward algorithm, so it is also referred to as the *forward-backward algorithm* (Baldi & Brunak, 1998 [4]). The algorithm starts with an initial guess/estimate $\widehat{\boldsymbol{\theta}}^{(0)} = (\widehat{\pi_i}^{(0)}$'s, $\widehat{a_{ij}}^{(0)}$'s, $\widehat{b_{j(k)}}^{(0)}$'s), where $i, j \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, M\}$. Then the EM cycles begin. For $c = 0, 1, \ldots,$ where c is the EM cycle index, each cycle has an expectation step (E-step) and a maximization step (M-step) as follows:

  1. **E-step:**

     Replace the "missing data" represented by $u_j(t)$'s and $v_{ij}(t)$'s in the $\log \mathcal{L}_{YX}(\boldsymbol{\theta})$ by their conditional expectations given the observed outcomes $y_{[1,L]}$ and the current parameter estimate $\widehat{\boldsymbol{\theta}}^{(c)}$. I.e.

$$\begin{aligned} \widehat{u_j(t)}^{(c)} &= E(u_j(t)|y_{[1,L]}, \widehat{\boldsymbol{\theta}}^{(c)}) = Pr(X_t = j|y_{[1,L]}, \widehat{\boldsymbol{\theta}}^{(c)}) \\ &\quad \text{(By expression (3.13) on page 37)} \qquad (3.25) \\ &= \frac{\alpha_t(j)^{(c)}\beta_t(j)^{(c)}}{\mathcal{L}_Y(\widehat{\boldsymbol{\theta}}^{(c)})}; \end{aligned}$$

and

$$
\begin{aligned}
\widehat{v_{ij}(t)}^{(c)} &= E(v_{ij}(t)|y_{[1,L]}, \widehat{\boldsymbol{\theta}}^{(c)}) = Pr(X_{t-1}=i, X_t=j|y_{[1,L]}, \widehat{\boldsymbol{\theta}}^{(c)}) \\
&= \frac{Pr(X_{t-1}=i, X_t=j, Y_{[1,L]}=y_{[1,L]}|\widehat{\boldsymbol{\theta}}^{(c)})}{\mathcal{L}_Y(\widehat{\boldsymbol{\theta}}^{(c)})} \\
&= \frac{Pr(Y_{[1,L]}=y_{[1,L]}|X_{t-1}=i, X_t=j, \widehat{\boldsymbol{\theta}}^{(c)})Pr(X_{t-1}=i|\widehat{\boldsymbol{\theta}}^{(c)})\widehat{a_{ij}}^{(c)}}{\mathcal{L}_Y(\widehat{\boldsymbol{\theta}}^{(c)})} \\
&\qquad \text{(By HMM Assumptions 1 \& 2)} \\
&= \frac{1}{\mathcal{L}_Y(\widehat{\boldsymbol{\theta}}^{(c)})}\Big(Pr(Y_{[1,t-1]}=y_{[1,t-1]}|X_{t-1}=i, \widehat{\boldsymbol{\theta}}^{(c)})Pr(Y_{[t,L]}=y_{[t,L]}|X_t=j, \widehat{\boldsymbol{\theta}}^{(c)}) \\
&\qquad \times Pr(X_t=j|\widehat{\boldsymbol{\theta}}^{(c)})\widehat{a_{ij}}^{(c)}\Big) \\
&= \frac{1}{\mathcal{L}_Y(\widehat{\boldsymbol{\theta}}^{(c)})}\Big(Pr(Y_{[1,t-1]}=y_{[1,t-1]}|X_{t-1}=i, \widehat{\boldsymbol{\theta}}^{(c)})Pr(Y_t=y_t|X_t=j, \widehat{\boldsymbol{\theta}}^{(c)}) \\
&\qquad \times Pr(Y_{[t+1,L]}=y_{[t+1,L]}|X_t=j, \widehat{\boldsymbol{\theta}}^{(c)})Pr(X_t=j|\widehat{\boldsymbol{\theta}}^{(c)})\widehat{a_{ij}}^{(c)}\Big) \\
&= \frac{\alpha_{t-1}(i)^{(c)}\widehat{b_{j(y_t)}}^{(c)}\beta_t(j)^{(c)}\widehat{a_{ij}}^{(c)}}{\mathcal{L}_Y(\widehat{\boldsymbol{\theta}}^{(c)})}.
\end{aligned}
\tag{3.26}
$$

2. **M-step:**

   Treat the conditional expected values $\widehat{u_j(t)}^{(c)}$ and $\widehat{v_{ij}(t)}^{(c)}$ from the E-step as data and solve the augmented-data likelihood maximization problem to get an updated estimate $\widehat{\boldsymbol{\theta}}^{(c+1)}$. The closed-form solution is:

$$
\widehat{\pi_i}^{(c+1)} = \widehat{u_i(1)}^{(c)}; \tag{3.27}
$$

$$
\widehat{a_{ij}}^{(c+1)} = \frac{\sum_{t=2}^{L}\widehat{v_{ij}(t)}^{(c)}}{\sum_{t=2}^{L}\sum_{j'=1}^{N}\widehat{v_{ij'}(t)}^{(c)}} = \frac{\sum_{t=2}^{L}\widehat{v_{ij}(t)}^{(c)}}{\sum_{t=1}^{L-1}\widehat{u_i(t)}^{(c)}}; \tag{3.28}
$$

and

$$\begin{aligned} \widehat{b_{j(k)}}^{(c+1)} &= \frac{\sum_{\{t:y_t=k\}} \widehat{u_j(t)}^{(c)}}{\sum_{t=1}^{L} \widehat{u_j(t)}^{(c)}} \\ &\qquad \text{(Using an indicator variable } 1_{y_t}\text{; where} \\ &\qquad 1_{y_t} = 1 \text{ if } Y_t = k\text{, otherwise } 1_{y_t} = 0.\text{)} \\ &= \frac{\sum_{t=1}^{L} \widehat{u_j(t)}^{(c)} 1_{y_t}}{\sum_{t=1}^{L} \widehat{u_j(t)}^{(c)}}. \end{aligned} \tag{3.29}$$

The EM cycles are iterated until convergence.

The convergence of $\widehat{\boldsymbol{\theta}}^{(c)}$ does not necessarily give us the global maximum, i.e. the maximum likelihood estimate (MLE) of the parameter set $\boldsymbol{\theta}$, for the likelihood functon. The likelihood surface of a general HMM often has multiple local maxima (or multiple modes), so different choices of initial estimates may result in landing on different local maxima of the likelihood surface. Depending on the specific application, the rate of convergence and the degree of multimodality of the likelihood surface vary, and they have created challenging mathematical/statistical/computational problems. Many variants of the EM algorithm have been currently proposed in order to speed up the rate of convergence (Details discussed in McLachlan & Krishnan, 1997 [101]). In the next chapter (§4.2.3 p.76–), a novel idea of using the finite Markov chain imbedding technique to calculate distributions of runs under an HMM is introduced. The investigation of the distributions of runs has led to an MLE-trapping scheme to substantially enhance the EM algorithm in the HMM parameter estimation procedure. Briefly, the trapping scheme is based on probabilistic profiles of runs to create a trapping grid (or trapping grids) to locate the neighbourhood of the MLE of $\boldsymbol{\theta}$. Details are illustrated through the investigation of a double runs statistic under a binary HMM.

Bayesian Approach for the HMM Training Problem

Briefly, the main focus of Bayesian statistics is on getting the *posterior distribution* of the parameter based on prior information and data. Prior information is represented by a prior probability distribution/density $Pr(Model)$ as the "degree of belief" on the $Model$ (NOTE: $Model$ is often expressed in terms of model parameter $\boldsymbol{\theta}$) prior to receiving current data; whereas data is represented by the likelihood function $Pr(Data|Model)$. In summary, Bayesian estimation/inference is based on the posterior distribution/density $Pr(Model|Data)$. As an alternative to the maximum likelihood estimation, the *maximum a posteriori* (MAP) estimation (or also referred to as the generalized maximum likelihood estimation (Carlin & Louis, 1996 [29])) has also been used. The MAP estimation maximizes the posterior distribution $Pr(Model|Data)$ rather than the likelihood function $Pr(Data|Model)$ in the maximum likelihood estimation. By Bayes' rule:

$$Pr(Model|Data) = \frac{Pr(Data|Model)\, Pr(Model)}{Pr(Data)}; \tag{3.30}$$

where $Pr(Data) = \int_{Model} Pr(Data|Model)\, Pr(Model)$. Since $Pr(Data)$ can be viewed as a normalizing constant, maximizing the posterior distribution $Pr(Model|Data)$ is same as maximizing $Pr(Data|Model)\, Pr(Model)$ of equation (3.30).

Under an HMM structure, equation (3.30) can be expressed as follows:

$$\begin{aligned} Pr(\boldsymbol{\theta}, X_{[1,L]}|Y_{[1,L]}) &= \frac{Pr(Y_{[1,L]}|\boldsymbol{\theta}, X_{[1,L]})Pr(\boldsymbol{\theta}, X_{[1,L]})}{Pr(Y_{[1,L]})} \\ &= \frac{Pr(Y_{[1,L]}|\boldsymbol{\theta}, X_{[1,L]})Pr(X_{[1,L]}|\boldsymbol{\theta})Pr(\boldsymbol{\theta})}{Pr(Y_{[1,L]})} \\ &= \frac{Pr(Y_{[1,L]}, X_{[1,L]}|\boldsymbol{\theta})Pr(\boldsymbol{\theta})}{Pr(Y_{[1,L]})} \\ &= \frac{\mathcal{L}_{YX}(\boldsymbol{\theta})Pr(\boldsymbol{\theta})}{Pr(Y_{[1,L]})} \end{aligned} \tag{3.31}$$

From this formulation, we can get the marginal posterior distribution $Pr(\boldsymbol{\theta}|y_{[1,L]})$ by summing over all possible paths $X_{[1,L]} = x_{[1,L]}$ in equation (3.31). We can also get any marginal posterior distribution by integrating out the corresponding selected components of $\boldsymbol{\theta}$ in $Pr(\boldsymbol{\theta}|y_{[1,L]})$. For example, the marginal posterior distribution of $b_{1(M)}$ is

$$Pr(b_{1(M)}|y_{[1,L]}) \\ = \int \cdots \int Pr(\boldsymbol{\theta}|y_{[1,L]})\, d\pi_1 \cdots d\pi_N\, da_{11} \cdots da_{NN}\, db_{1(1)} \cdots db_{1(M-1)}\, db_{2(1)} \cdots\cdots db_{N(M)}, \tag{3.32}$$

where $b_{1(M)}$ is treated as a random variable with a corresponding prior probability distribution. Maximizing these marginal posterior distributions (e.g. $Pr(\boldsymbol{\theta}|y_{[1,L]})$ and $Pr(b_{1(M)}|y_{[1,L]})$) will give us the MAP estimates for $\boldsymbol{\theta}$ and any selected component(s) of $\boldsymbol{\theta}$. The Bayesian approach has been used to deal with the HMM training problem of various applications in DNA sequence analysis (e.g. Churchill, 1995 [33]; Hughey & Krogh, 1996 [68]; Crowley, Roeder & Bina, 1997 [38]); Baldi & Brunak, 1998 [4]; and Churchill & Lazareva, 1999 [35]).

Bayesian Approach for the HMM Alignment Problem

As mentioned earlier, the HMM alignment problem can also be viewed from a Bayesian perspective. Using a Bayesian formulation such as equation (3.31), we can get the marginal posterior distribution $Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]})$ by integrating out the parameter $\boldsymbol{\theta}$ in equation (3.31). We can also get any marginal posterior distribution by summing over the corresponding selected components of $x_{[1,L]}$ in $Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]})$. For example,

$$Pr(X_t = x_t|y_{[1,L]}) = \sum_{x_1=1}^{N} \cdots \sum_{x_{t-1}=1}^{N} \sum_{x_{t+1}=1}^{N} \cdots \sum_{x_L=1}^{N} Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]}). \tag{3.33}$$

A searching routine may then be implemented to find a *path*, $x_{[1,L]}$, which will give the highest marginal posterior probability $Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]})$; or to find an x_t which will give the highest marginal posterior probability $Pr(X_t = x_t|y_{[1,L]})$. Again, the key idea of using the Bayesian approach for both the training problem and the alignment problem associated with an HMM is to get the appropriate marginal posterior distributions. Having these marginal posterior distributions, we can perform Bayesian estimation/inference. However, computations involved in getting these marginal posterior distributions are difficult in practice. If our goal is to perform the MAP estimation alone, then the EM algorithm or its variants can, in principle, be used as in the maximum likelihood estimation. Many "EM-like" algorithms have been proposed for the MAP estimation. For a more ambitious goal to do higher levels of Bayesian inference, other methods are needed to overcome the computational difficulty involved in getting the exact posterior distributions. Recently, different *Markov Chain Monte Carlo* (MCMC) methods have been proposed for approximating the desired posterior distributions. Briefly, MCMC methods are computer-intensive statistical techniques used to generate random variables from a distribution* indirectly. Instead of dealing with the distribution directly, an MCMC method uses the conditional distributions iteratively. The *Gibbs sampler*, a specific MCMC method, has been recently applied to iteratively simulate random samples from the conditional distributions $Pr(X_{[1,L]} = x_{[1,L]}|y_{[1,L]}, \theta)$ and $Pr(\theta|y_{[1,L]}, x_{[1,L]})$ for the alignment problem of an HMM (Churchill & Lazareva, 1999 [35])). There are still a lot of open mathematical/statistical/computational questions and issues that need further investigation when an MCMC method is employed.

*NOTE: MCMC methods can be applied to simulate not only the desired posterior distribution, but also the likelihood function (Casella & George, 1992 [30]).

Chapter 4

RUNS STATISTICS FOR DNA PATTERN RECOGNITION

4.1 Distribution Theory of Runs: A Review

Historically, problems of counting runs and patterns have been closely associated with probability, and the original motivation for studying probability was in games of chance: rolling dice, flipping coins, and card games. After Laplace first defined probability in a finite sample space, there was much interest in games of chance during the early nineteenth century (Tucker, 1995 [130]). However, according to Mood, the distribution theory of runs is believed to have started at about the end of the nineteenth century. In 1897 Karl Pearson started to treat certain distributions of runs as special cases of the multinomial distribution. The multinomial method for studying the distribution of runs raised questions of random sampling, and it was known to give incorrect results when it was used without caution. In 1925 Ising described the number of ways of obtaining a given total number of runs (regardless of their lengths) from arrangements of two kinds of elements, as a result a formal function/formula for the distribution of the number of runs (regardless of their lengths) appeared (Mood, 1940 [103]). Later in 1939 Stevens proved and discussed the same distribution as a χ^2

criterion for significance (Stevens, 1939 [127]). In 1940 Mood, one of the well-known researchers who had a great influence in the area of the distribution theory of runs, wrote that

> *"The distribution problem is, of course, a combinatorial one, and the whole development depends on some identities in combinatory analysis, ...* "(Mood, 1940 [103]).

In the 1940's research on the distribution theory of runs was primarily concentrated on combinatorial studies of the conditional distributions of success (and/or failure) runs given the total number of successes (and/or failures) in a sequence of independent identically distributed (IID) trials (e.g. Mood, 1940 [103]; Wald & Wolfowitz, 1940 [134]; Mosteller, 1941 [104]; Wolfowitz, 1943 [152]; David, 1947 [39]). After a dormant period of about 30 years, the interest in finding the exact distributions of different versions of runs had been awakened since the mid-1980's (e.g. Fu et al., 1986, 1994, 1996 [55, 58, 56]; Philippou & Makri, 1986 [114]; Hirano, 1986 [66]; Godbole, 1990 [61]; Hirano & Aki, 1993 [67]; Lou, 1996, 1997 [97, 98]). Recently, the use of runs statistics and their distributions has been brought to the area of bioinformatics, especially for pattern recognition in DNA sequences. For example, current selected work on using various runs and runs-related statistics includes publications by Karlin et al., 1992, 1993, 1994, 1996, 1997 [75, 74, 78, 79, 80]; Leung et al., 1994, 1996 1999 [92, 91, 93]; Waterman, 1995 [144, 143]; Fu et al., 1996, 1999 [57, 59]; and Lou, 2000 [99]. In the following subsections, definitions of runs and commonly used runs statistics are first provided. Then, the traditional combinatorial approach and the recent finite Markov chain imbedding (FMCI) approach for studying the distribution theory of runs are described with illustrative examples.

4.1.1 What Are Runs and Runs Statistics?

In general, a run is defined as a succession of similar events preceded and succeeded by different events in a sequence of observations. When a sequence of L dichotomous trials (e.g. an outcome is either a success (S) or a failure (F) at each trial) is considered, a success run is defined as a sequence of consecutive outcomes of successes preceded and succeeded by failures. The size or length of a success run refers to the number of successes in the run. Analogously, these definitions are also adopted for numerous studies of other runs and patterns. However, it is worth mentioning that new and/or modified definitions often emerge from different ways of counting when studies of runs and patterns are applied to practical problems of interest. For example, the five most commonly used runs statistics in the literature reviewed by Fu and Koutras (Fu & Koutras, 1994 [58]):

Given L and l, where $1 \leq l \leq L$;

- R_l is the number of success runs of size (or length) exactly l.
- $R_{l\uparrow}$ is the number of success runs of size (or length) greater than or equal to l.
- $R_{l_{)(}}$ is the number of non-overlapping consecutive l successes.
- $R_{l_{\between}}$ is the number of overlapping consecutive l successes.
- $\underset{\leftrightsquigarrow}{R}$ is the size (or length) of the longest success run.

NOTE: For convenience, R_{stats} is used to denote any of the above five runs statistics.

For example, if a sequence of 15 outcomes (i.e. $L = 15$): $FSSSSFFFSFFSSFF$ is observed and we are interested in setting $l = 2$, then we have:

$$R_{stats} = \begin{cases} R_l = R_2 = 1 & \because \quad FSSSSFFFSFF\underline{SS}FF, \\ R_{l\uparrow} = R_{2\uparrow} = 2 & \because \quad F\underline{SSSS}FFFSFF\underline{SS}FF, \\ R_{l_{)(}} = R_{2_{)(}} = 3 & \because \quad F\underline{SS}\,\overline{SS}FFFSFF\underline{SS}FF, \\ R_{l_{\between}} = R_{2_{\between}} = 4 & \because \quad F\underline{S\overline{S}}\,\underline{\overline{S}S}FFFSFF\underline{SS}FF, \\ \underset{\leftrightsquigarrow}{R} = 4 & \because \quad F\,\underset{\text{max. size}}{\underline{SSSS}}\,FFFSFFSSFF. \end{cases}$$

NOTE: In this thesis the number of success runs regardless of their lengths is denoted as R, and the total number of successes is denoted as N_s. Based on all the definitions of runs statistics mentioned above, we have the following:

- Setting $l = 1$, $\overset{\text{implies}}{\Longrightarrow}$ $\begin{cases} R_{l\uparrow} \equiv R_{1\uparrow} \equiv R, \\ R_{l_{)(}} \equiv R_{1_{)(}} \equiv R_{1_{\between}} \equiv N_s; \end{cases}$
- $\sum_{l=1}^{L} R_l = R_{1\uparrow} \equiv R;$
- $\sum_{l=1}^{L} l \cdot R_l = N_s;$
- $R_l \leq R_{l\uparrow} \leq R_{l_{)(}} \leq R_{l_{\between}};$
- $R_l = R_{l\uparrow} - R_{(l+1)\uparrow};$ and
- $\underset{\leftrightsquigarrow}{R} < l \quad \overset{\text{iff}}{\Longleftrightarrow} \quad R_{l_{)(}} = 0.$

Runs statistics are natural statistical descriptions of patterns when we are interested in analyzing sequence data. Although a number of creative considerations of different runs statistics can be designed, this thesis concentrates on studying the joint distribution of the total number of successes N_s and the number of success runs R in a sequence of dichotomous trials. The bivariate random variable (N_s, R) is called a "double runs statistic".

As Fu and Koutras stated, except for a few special cases, the exact distributions of many runs statistics through the combinatorial approach have remained unknown, especially when the sequence of trials is not IID (Fu & Koutras, 1994 [58]). In the next subsection, the combinatorial approach for finding the distributions of a few selected runs statistics is reviewed. The complexity of the combinatorial approach may explain why it often becomes intractable quickly when we are dealing with non-trivial runs statistics and/or non-IID trials.

4.1.2 Combinatorial Approach

Traditionally, the distribution theory of runs has been studied through the combinatorial approach. The basis of the combinatorial approach is generally making use of the binomial coefficient (i.e. $\binom{L}{n} = \frac{L!}{n!\,(L-n)!}$) and the multinomial coefficient (i.e. $\binom{r_S}{r_{S_1}, r_{S_2}, \cdots, r_{S_n}} = \frac{r_S!}{\prod_{i=1}^{n}(r_{S_i}!)}$, where $r_S = \sum_{i=1}^{n} r_{S_i}$ and $r_{S_i} \geq 0$). As mentioned previously, early research on the combinatorial studies of the distribution theory of runs was primarily focused on the conditional distributions of success (and/or failure) runs given the total number of successes (denoted as n here) in a sequence of L independent identically distributed (IID) trials. For example, given L and n, let $\boldsymbol{R_{S_iF_j}}$ be a random vector with its first n coordinates being the numbers of success runs of lengths $i = 1, \dots, n$, and its last $L - n$ (i.e. $(n+1)$-th to L-th) coordinates being the numbers of failure runs of lengths $j = 1, \dots, L - n$; and let $\boldsymbol{r_{S_iF_j}} = (r_{S_1}, \dots, r_{S_i}, \dots, r_{S_n}, r_{F_1}, \dots, r_{F_j}, \dots, r_{F_{L-n}})$ be a vector of nonnegative integers corresponding to the collection of all the observed numbers of runs of successes and failures of specific lengths respectively. Then one can examine the combinatorial arrangements of these runs for studying the distribution of $\boldsymbol{R_{S_iF_j}}$. Specifically, if a sequence of 15 outcomes is $FSSSSFFFSFFSSFF$, which has 7 successes and $15 - 7 = 8$ failures (i.e. $L = 15$ and $n = 7$), then the collection of the observed numbers of runs of successes and failures of specific lengths is:

$$\boldsymbol{r_{S_iF_j}} = (1,1,0,1,0,0,0,1,2,1,0,0,0,0,0) \quad \text{because}$$

$$F\underline{SSSS}FFF\underline{S}FF\underline{SS}FF \quad \overset{\text{implies}}{\Longrightarrow} \quad \begin{cases} r_{S_1} = r_{S_2} = r_{S_4} = 1, \\ r_{S_i} = 0 \quad \text{for } i = 3,5,6,7, \\ r_{F_1} = r_{F_3} = 1, \\ r_{F_2} = 2 \quad \text{and,} \\ r_{F_j} = 0 \quad \text{for } j = 4,5,6,7,8. \end{cases}$$

NOTE: $\sum_i r_{S_i} = r_S$ and $\sum_i i \cdot r_{S_i} = n$; whereas $\sum_j r_{F_j} = r_F$ and $\sum_j j \cdot r_{F_j} = L - n$.

In 1940 Mood stated that there are $\binom{r_S}{r_{S_1}, r_{S_2}, \dots, r_{S_n}}$ and $\binom{r_F}{r_{F_1}, r_{F_2}, \dots, r_{F_{L-n}}}$ possible different arrangements of the success runs and the failure runs in a sequence of L dichotomous outcomes respectively. Moreover, the total number of ways of having a particular set $\boldsymbol{r_{S_iF_j}}$ was provided by Mood (Mood, 1940 [103]):

$$Total_{comb}(\boldsymbol{r_{S_iF_j}}) = \binom{r_S}{r_{S_1}, r_{S_2}, \cdots, r_{S_n}} \binom{r_F}{r_{F_1}, r_{F_2}, \cdots, r_{F_{L-n}}} Arr(r_S, r_F). \tag{4.1}$$

NOTE: $Arr(r_S, r_F)$ is the number of ways of arranging r_S runs of successes and r_F runs of failures so that there are no two adjacent runs of the same kind. I.e.

$$Arr(r_S, r_F) = \begin{cases} 0 & \text{if } |r_S - r_F| > 1, \\ 1 & \text{if } |r_S - r_F| = 1, \\ 2 & \text{if } r_S = r_F. \end{cases}$$

Since there are $\binom{L}{n}$ possible arrangements of the n successes and $(L-n)$ failures in a sequence of L outcomes, Mood obtained the distribution of $\boldsymbol{R_{S_iF_j}}$ as follows (Mood, 1940 [103]):

$$Pr(\boldsymbol{R_{S_iF_j}} = \boldsymbol{r_{S_iF_j}}) = \frac{Total_{comb}(\boldsymbol{r_{S_iF_j}})}{\binom{L}{n}}; \quad \text{provided that } n \text{ is known.} \tag{4.2}$$

Furthermore, summing over appropriate terms in the distribution formula (4.2) of $\boldsymbol{R_{S_iF_j}}$ with combinatorial arguments in a sequence of L IID (or Bernoulli) trials, the following distributions were also derived by Mood (Mood, 1940 [103]):

$$Pr(R_{success} = r_S, R_{failure} = r_F) = \frac{\binom{n-1}{r_S-1}\binom{L-n-1}{r_F-1} Arr(r_S, r_F)}{\binom{L}{n}}; \quad \text{provided that } n \text{ is known.} \tag{4.3}$$

$$Pr(R_{success} = r_S) = \frac{\binom{n-1}{r_S-1}\binom{L-n+1}{r_S}}{\binom{L}{n}}; \quad \text{provided that } n \text{ is known.} \tag{4.4}$$

NOTE: The total number of success runs is denoted as $R_{success}$ (or also denoted as R) and the total number of failure runs is denoted as $R_{failure}$.

The above distribution formula (4.3) — $Pr(R_{success} = r_S, R_{failure} = r_F)$ — was also proven by Wald and Wolfowitz (Wald & Wolfowitz, 1940 [134]); and the formula (4.4) — $Pr(R_{success} = r_S)$, or $Pr(R = r)$ in our notation — was also previously proven and discussed by Stevens (Stevens, 1939 [127]). However, when the number of successes (i.e. n) is unknown, there has been no explicit formula derived from the combinatorial approach for the distribution of the number of success runs of length exactly l, i.e. the unconditional distribution of R_l still poses problems when the combinatorial approach is adopted (Fu & Koutras, 1994 [58]).

For a non-trivial runs statistic such as the number of non-overlapping consecutive l successes (i.e. $R_{l)(}$) in a sequence of L Bernoulli trials, the combinatorial approach for finding its distribution becomes rather complex. In 1986 the distribution of $R_{l)(}$ in a sequence of L Bernoulli trials was established by Philippou and Makri and also by Hirano independently. And in 1990 a specific formula of this distribution was provided by Godbole as follows (Godbole, 1990 [61]):

$$Pr(R_{l)(} = r) = \sum_{Floor(\frac{L-lr}{k}) \le i \le L-lr} \left[P_F^i P_S^{L-i} \binom{i+r}{r} \times \sum_{0 \le j \le Floor(\frac{L-lr-i}{l})} (-1)^j \binom{i+1}{j} \binom{L-lr-jl}{i} \right]. \tag{4.5}$$

NOTE: $r = 0, 1, \ldots, r_{max} = Floor\left(\frac{L}{l}\right)$; and $Floor(\cdot)$ denotes an operator that rounds its operand to the largest integer not exceeding the operand. $P_F = Pr(Y_t = F)$ and $P_S = Pr(Y_t = S)$.

The combinatorial approach for studying distributions of runs in a non-IID probabilistic framework can easily get very complex and computationally demanding even for a simple runs statistic such as the total number of successes (i.e. N_s). For example, if a sequence of L dichotomous trials $\{Y_t : t \in \Gamma_L\}$ is first order non-homogeneous Markov dependent with the state space $\Omega_Y = \{S, F\}$ and the transition probability matrices

$$\mathcal{A}_Y(t) = \begin{array}{c c} Y_{t-1}\backslash Y_t & \begin{array}{cc} \quad F \quad & \quad S \quad \end{array} \\ \begin{array}{c} F \\ S \end{array} & \left[\begin{array}{cc} P_{FF}(t) & P_{FS}(t) \\ P_{SF}(t) & P_{SS}(t) \end{array}\right] \end{array};$$

then a combinatorial method for the distribution of N_s suggested by Lou (Lou, 1996 [97]) is as follows:

$$\begin{aligned}
Pr(N_s = n) &= Pr\Big((N_s = n) \cap \big((Y_1 = S) \cup (Y_1 = F)\big)\Big) \\
&= Pr\big((N_s = n) \cap (Y_1 = S)\big) + Pr\big((N_s = n) \cap (Y_1 = F)\big) \\
&= Pr(Y_1 = S)Pr(N_s = n|Y_1 = S) + Pr(Y_1 = F)Pr(N_s = n|Y_1 = F)
\end{aligned}$$

(Using an indicator variable 1_{y_t}; where $1_{y_t} = 1$ if $Y_t = S$, otherwise $1_{y_t} = 0$.)

$$\begin{aligned}
&= Pr(Y_1 = S) \sum_{\substack{\text{All possible } y_{[2,L]}\text{'s} \\ \text{s.t. } 1_{y_2}+\ldots+1_{y_L}=n-1}} \left(\prod_{t=2}^{L} P_{FF}(t)^{(1-1_{y_{t-1}})(1-1_{y_t})} \, P_{FS}(t)^{(1-1_{y_{t-1}})1_{y_t}} \, P_{SF}(t)^{1_{y_{t-1}}(1-1_{y_t})} \, P_{SS}(t)^{1_{y_{t-1}}1_{y_t}} \right) \\
&\quad + Pr(Y_1 = F) \sum_{\substack{\text{All possible } y_{[2,L]}\text{'s} \\ \text{s.t. } 1_{y_2}+\ldots+1_{y_L}=n}} \left(\prod_{t=2}^{L} P_{FF}(t)^{(1-1_{y_{t-1}})(1-1_{y_t})} \, P_{FS}(t)^{(1-1_{y_{t-1}})1_{y_t}} \, P_{SF}(t)^{1_{y_{t-1}}(1-1_{y_t})} \, P_{SS}(t)^{1_{y_{t-1}}1_{y_t}} \right).
\end{aligned} \tag{4.6}$$

In the above combinatorial formula (4.6), the conditions $1_{y_2} + \ldots + 1_{y_L} = n - 1$ and $1_{y_2} + \ldots + 1_{y_L} = n$ for the first and second summations imply that there are $\binom{L-1}{n-1}$ and $\binom{L-1}{n}$ possible $y_{[2,L]}$ arrangements respectively at a particular value of n. I.e. In total, there are $\binom{L-1}{n-1} + \binom{L-1}{n} = \binom{L}{n}$ possible arrangements needed to go through at a particular value of n. E.g. For $L = 500$, at $n = 250$ there are $\binom{500}{250} \doteq 1.167 \times 10^{149}$ arrangements. It is clearly very computationally demanding, especially when L is large, to obtain the whole distribution of N_s through the above combinatorial approach.

In 1994, an approach called the finite Markov chain imbedding (FMCI) was introduced for studying distributions of runs and patterns by Fu and Koutras (Fu & Koutras, 1994 [58]). The key idea behind this approach is simple, yet profound. We believe that it is certainly an attractive alternative, especially when we are dealing with non-trivial runs statistics and/or non-IID trials, for the distribution theory of runs. In the following subsection, a review of the FMCI approach is given with an illustrative example.

4.1.3 Finite Markov Chain Imbedding (FMCI) Approach

The FMCI technique was first invented by professor James C. Fu* for studying the reliability problems of linearly connected engineering systems in the mid-1980's (Fu, 1986 [55]). It has then been successfully adopted as a unified approach for finding the exact distributions of the (previously mentioned) five most commonly used runs statistics — $R_l, R_{l\uparrow}, R_{l)(}, R_{l\between}$, and $\underset{\leftrightsquigarrow}{R}$ — in a sequence of independent identically (or non-identically) distributed dichotomous trials (Fu & Koutras, 1994 [58]). Later, the concept of using FMCI has also been extended to study the exact distributions of runs and patterns in a Markov dependent sequence of polychotomous trials (Fu, 1996 [56]). Although there is much interest in the case of polychotomous trials, further investigation in this topic will be left for future research (especially for protein sequence analysis). In the case of dichotomous trials, the distribution theory of runs through the FMCI technique has been studied in the following four probabilistic frameworks. I.e. Given L and a finite index set $\Gamma_L = \{1, \ldots, L\}$, a sequence of L dichotomous trials $\{Y_t : t \in \Gamma_L\}$, with each random variable Y_t is either a success S or a failure F for all $t \in \Gamma_L$, is considered to be:

- independent and identically distributed with $Pr(Y_t = S) = P_S$ and $Pr(Y_t = F) = P_F$, $\forall t$. Or,
- independent but non-identically distributed with $Pr(Y_t = S) = P_S(t)$ and $Pr(Y_t = F) = P_F(t)$, $\forall t$. Or,

*Dr. James C. Fu (Ph.D. in Statistics) is a professor of the Department of Statistics at the University of Manitoba in Winnipeg, Canada.

- first order homogeneous Markov dependent with the transition probability matrix:

$$\mathcal{A}_Y = \begin{array}{c|cc} Y_{t-1}\backslash Y_t & F & S \\ \hline F & p_{FF} & p_{FS} \\ S & p_{SF} & p_{SS} \end{array}, \quad \forall t. \qquad \text{Or,}$$

- first order non-homogeneous Markov dependent with the transition probability matrices:

$$\mathcal{A}_Y(t) = \begin{array}{c|cc} Y_{t-1}\backslash Y_t & F & S \\ \hline F & p_{FF}(t) & p_{FS}(t) \\ S & p_{SF}(t) & p_{SS}(t) \end{array}, \quad \forall t.$$

NOTE: The FMCI technique can also be extended to the cases of higher order (i.e. order > 1) homogeneous or nonhomogeneous Markov dependent trials, but they will not be pursued here.

The fundamental concepts of the FMCI technique are as follows:

FMCI DEFINITION (Fu & Koutras, 1994 [58] and Fu, 1996 [56])

A nonnegative integer random variable R_{stats}, defined from a sequence of L dichotomous (or polychotomous) trials, is finite Markov chain imbeddable if

- there exists a finite Markov chain $\{Z_t : t \in \Gamma_L\}$ defined on a finite state space $\Omega = \{s_1, \ldots, s_h\}$ with its transition probability matrix $\Lambda(t)$ at time t, where $t \in \Gamma_L$, and
- there exists a finite partition $\{C_r, r = 0, 1, \ldots, r_{max}\}$ on the state space Ω (where C_r and r_{max} may depend on L), such that

$$Pr(R_{stats} = r) = Pr(Z_L \in C_r), \quad \text{for every} \quad r = 0, 1, \ldots, r_{max}. \qquad (4.7)$$

FMCI THEOREM (Fu & Koutras, 1994 [58])

For a given L, if a statistic R_{stats} can be imbedded into a finite Markov chain $\{Z_t, \Omega, \Lambda(t) : t \in$

$\Gamma_L\}$, then

$$Pr(R_{stats} = r) = Pr(Z_L \in C_r) = \pi^{FMCI}\Big(\prod_{t=2}^{L} \Lambda(t)\Big)U(C_r), \quad \text{for every} \quad r = 0, 1, \ldots, r_{max}; \tag{4.8}$$

where $\pi^{FMCI} = (Pr(Z_1 = s_1), Pr(Z_1 = s_2), \ldots, Pr(Z_1 = s_h))$ is the initial probability of the imbedded Markov chain; and $U(C_r) = \sum_{s_g \in C_r} U(s_g)$ with $U(s_g)$ being a $h \times 1$ unit column vector having a value of 1 at the g-th coordinate and zero otherwise.

Essentially, the FMCI approach is a reformulation of the problem of finding the exact distribution of a runs statistic, which has been traditionally dealt with from a combinatorial viewpoint. The complexity and the computational burden of the combinatorial approach have been the major hurdles for formulating and computing the distributions of many runs statistics, even for some of the basic runs statistics such as the number of success runs of length exactly l (i.e. R_l) in the simplest case of independent and identically distributed (IID) trials. The combinatorial approach often requires tremendous effort (if not intractable) when dealing with non-trivial runs statistics and/or non-IID trials. On the other hand, the idea of the FMCI approach is to imbed the runs statistic of interest into a finite Markov chain so that the exact distribution of the statistic can then be expressed in terms of transition probabilities of the imbedded Markov chain. The above formula (4.8) is indeed a matrix version of the *Chapman-Kolmogorov Theorem.* When the imbedded Markov chain is assumed to be *homogeneous* (i.e. $\Lambda(t) = \Lambda, \forall t \in \Gamma_L$), then $(\prod_{t=2}^{L} \Lambda(t)) = \Lambda^{L-1}$ and the formula (4.8) becomes the following (Fu & Koutras, 1994 [58]):

$$Pr(R_{stats} = r) = \pi^{FMCI}\Lambda^{L-1}\, U(C_r), \quad \text{for every} \quad r = 0, 1, \ldots, r_{max}. \tag{4.9}$$

Since the key difference between the non-homogeneous FMCI and the homogeneous FMCI is their underlying transition probabilities (i.e. the former one with $(L-1)$ time-dependent $\Lambda(t)$'s, whereas the latter one with a single time-independent Λ), without loss of generality, this thesis work has been mainly concerned with the homogeneous Markov chain imbedding in order to avoid extreme multi-dimensionality of the parameter space. It is worth mentioning that results from the non-homogeneous FMCI can be obtained by replacing Λ by $\Lambda(t)$. To illustrate the FMCI technique, an example is provided below in detail.

FMCI Example: Distribution of R

For a sequence of L first order homogeneous Markov dependent dichotomous trials $\{Y_t, \Omega_Y, \mathcal{A}_Y : t \in \Gamma_L\}$, the number of success runs — R — can be imbedded into a finite Markov chain by constructing the following:

- A homogeneous Markov chain $\{Z_t = (R_{s_t}, Y_t) : t \in \Gamma_L\}$, where R_{s_t} is the number of success runs in the first t trials, and Y_t is the outcome of the t-th trial.

- A finite state space $\Omega = \{(0,F)\} \cup \Big\{(r,y) : r \in \{1,\dots,r_{max}-1\}; y \in \{S,F\}\Big\} \cup Set_{r_{max}}$; where $Set_{r_{max}} = \begin{cases} \{(r_{max},S),(r_{max},F)\} & \text{if } L \text{ is even,} \\ \{(r_{max},S)\} & \text{if } L \text{ is odd;} \end{cases}$ and $r_{max} = \begin{cases} \frac{L}{2} & \text{if } L \text{ is even,} \\ \frac{L+1}{2} & \text{if } L \text{ is odd.} \end{cases}$
  NOTE: $\Omega = \{s_1,\dots,s_h\}$ is a set of 2-tuple (r,y) states, and the number of elements in $\Omega = size(\Omega) = h = L+1$.

- The transition probability matrix $\Lambda = [Pr(Z_t = z_t | Z_{t-1} = z_{t-1})]$ of the imbedded Markov chain:

  For $t = 2,\dots,L$;

  - $Pr(Z_t = (r,F)|Z_{t-1} = (r,S)) = \begin{cases} \text{Invalid} & \text{if } r = r_{max} \text{ \& } L \text{ is odd,} \\ Pr(Y_t = F|Y_{t-1} = S) = P_{SF} & \text{otherwise.} \end{cases};$
  - $Pr(Z_t = (r,F)|Z_{t-1} = (r,F)) = \begin{cases} 1 & \text{if } r = r_{max} \text{ \& } L \text{ is even,} \\ Pr(Y_t = F|Y_{t-1} = F) = P_{FF} & \text{otherwise.} \end{cases};$
  - $Pr(Z_t = (r,S)|Z_{t-1} = (r,S)) = \begin{cases} 1 & \text{if } r = r_{max} \text{ \& } L \text{ is odd,} \\ Pr(Y_t = S|Y_{t-1} = S) = P_{SS} & \text{otherwise.} \end{cases};$
  - $Pr(Z_t = (r+1,S)|Z_{t-1} = (r,F)) = Pr(Y_t = S|Y_{t-1} = F) = P_{FS}$ if $r < r_{max}$.
  - Otherwise, the transition probabilities are zero.

  NOTE: If L is an odd integer, then Λ remains almost the same except that the state (r_{max}, F) is no longer valid, and the state (r_{max}, S) becomes the absorbing state.

- A finite partition $C_r = \begin{cases} \{(0,F)\} & \text{if } r = 0, \\ \{(r,S),(r,F)\} & \text{if } r \in \{1,\dots,r_{max}-1\}, \\ \{(r_{max},S),(r_{max},F)\} & \text{if } r = r_{max} \text{ \& } L \text{ is even}, \\ \{(r_{max},S)\} & \text{if } r = r_{max} \text{ \& } L \text{ is odd}, \end{cases}$ on the state space Ω; such that $Pr(R=r) = Pr(Z_L \in C_r)$, for every r.

Then, given the initial probability of the imbedded Markov chain, $\pi^{FMCI} = (Pr(Z_1 = s_1), Pr(Z_1 = s_2), \dots, Pr(Z_1 = s_h)) = (Pr(Z_1 = (0,F)), Pr(Z_1 = (1,S)), 0, \dots, 0)$, the FMCI theorem can be applied and we have:

$$Pr(R=r) = Pr(Z_L \in C_r) = \pi^{FMCI}\Lambda^{L-1}U(C_r), \quad \text{for every } r. \tag{4.10}$$

For example, if $L = 5$, then the state space $\Omega = \{(0,F),(1,S),(1,F),(2,S),(2,F),(3,S)\}$, $size(\Omega) = h = 5+1 = 6$, and the transition probability matrix:

$$\Lambda = \begin{array}{c} Z_{t-1}\backslash Z_t \\ (0,F) \\ (1,S) \\ (1,F) \\ (2,S) \\ (2,F) \\ (3,S) \end{array} \begin{array}{c} \begin{array}{cccccc} (0,F) & (1,S) & (1,F) & (2,S) & (2,F) & (3,S) \end{array} \\ \begin{bmatrix} p_{FF} & p_{FS} & 0 & 0 & 0 & 0 \\ 0 & p_{SS} & p_{SF} & 0 & 0 & 0 \\ 0 & 0 & p_{FF} & p_{FS} & 0 & 0 \\ 0 & 0 & 0 & p_{SS} & p_{SF} & 0 \\ 0 & 0 & 0 & 0 & p_{FF} & p_{FS} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}.$$

REMARK: If the trials $\{Y_t : t \in \Gamma_L\}$ are indeed independent and identically distributed rather than first order homogeneous Markov dependent, then we can simply adjust the transition probability matrix of the imbedded Markov chain and apply the FMCI theorem. I.e.

$$\left.\begin{array}{r} \text{By setting } P_{FF} = P_{SF} = P_F \\ \text{and } P_{FS} = P_{SS} = P_S \end{array}\right\} \overset{\text{leads to}}{\Longrightarrow} \Lambda_{(IID)} = \begin{array}{c|cccccc|} Z_{t-1}\backslash Z_t & (0,F) & (1,S) & (1,F) & (2,S) & (2,F) & (3,S) \\ \hline (0,F) & P_F & P_S & 0 & 0 & 0 & 0 \\ (1,S) & 0 & P_S & P_F & 0 & 0 & 0 \\ (1,F) & 0 & 0 & P_F & P_S & 0 & 0 \\ (2,S) & 0 & 0 & 0 & P_S & P_F & 0 \\ (2,F) & 0 & 0 & 0 & 0 & P_F & P_S \\ (3,S) & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array},$$

and

$$Pr(R = r) = \pi^{FMCI} \Lambda_{(IID)}^{L-1} U(C_r), \quad \text{for every } r. \tag{4.11}$$

As mentioned early, if the trials are non-homogeneous first order Markov dependent, then the distribution of R can be obtained by replacing the transition probabilities P_{FF}, P_{FS}, P_{SF}, and P_{SS} in Λ with time-dependent transition probabilities $P_{FF}(t), P_{FS}(t), P_{SF}(t)$, and $P_{SS}(t) \overset{\text{leads to}}{\Longrightarrow} \Lambda(t)$. And,

$$Pr(R = r) = Pr(Z_L \in C_r) = \pi^{FMCI} \Big(\prod_{t=2}^{L} \Lambda(t)\Big) U(C_r), \quad \text{for every } r. \tag{4.12}$$

Furthermore, if the trials are independent but non-identically distributed, then we can replace the probabilities P_F, and P_S in $\Lambda_{(IID)}$ with time-dependent probabilities $P_F(t)$, and $P_S(t) \overset{\text{leads to}}{\Longrightarrow} \Lambda_{(INID)}(t)$. And,

$$Pr(R = r) = Pr(Z_L \in C_r) = \pi^{FMCI} \Big(\prod_{t=2}^{L} \Lambda_{(INID)}(t)\Big) U(C_r), \quad \text{for every } r. \tag{4.13}$$

4.2 Studying Runs and Patterns in DNA Via FMCI: From an Independent Identically Distributed (IID) to a Markov Chain (MC) to an HMM Framework

It is perhaps because of the strong influence of the combinatorial approach to the distribution theory of runs that most research on runs and patterns in DNA (or RNA or protein) sequences has been mainly restricted to the independent identically distributed (IID) framework (e.g. Karlin et al.,1992, 1996, 1997 [75, 79, 80]; Waterman, 1995 [144]; Leung & Yamashita, 1999 [93]). Although studies on specific runs-related statistics and their distributions under an IID framework have been demonstrated to be useful for detecting non-random patterns in DNA, the IID assumption is often violated in general practice. With the invention of the finite Markov chain imbedding (FMCI) technique, Fu and his colleagues have started to adopt their FMCI work for DNA pattern recognition under an assumption that nucleotides of a DNA sequence have Markov dependent nitrogenous bases (e.g. Fu et al., 1999 [59]; Lou, 2000 [99]).

Having learned that carefully designed hidden Markov models (HMMs) are capable of capturing certain mosaic structures in DNA (e.g. Churchill, 1992 [32]; Krogh et al., 1994 [85]; Burge & Karlin, 1997 [27]; Baldi & Brunak, 1998 [4]; Durbin et al., 1998 [45]; Balding et al., 2001 [10]), a novel idea of using the FMCI technique to study runs and patterns in DNA under an HMM has emerged. In essence, this work coalesces research on runs-related statistics and research on HMMs for DNA pattern recognition through the use of the FMCI technique. With a vision of studying elaborate multiple runs-related statistics simultaneously under an HMM through the FMCI technique, this work establishes an investigation of the double runs statistic (N_s, R) , i.e. the total number of successes and the number of success runs, in a binary sequence under a binary HMM for DNA pattern recognition. Before proceeding to the investigation, a general recursive algorithm based on the FMCI technique is derived and implemented for the determination of the exact distribution of this double runs statistic under an IID framework, an MC framework, and a binary HMM framework.

4.2.1 Distribution of a Double Runs Statistic Under an IID or an MC Framework

In 1996, Lou developed a numerical method based on the FMCI technique for the determination of the exact distribution of the double runs statistic (N_s, R) under not only an independent identically distributed (IID) framework, but also a Markov chain (MC) framework (Lou, 1996 [97]). Since Lou simply applied the FMCI formula (4.8) directly to her work, her numerical method suffers from a lot of *non-productive operations* (i.e. $0+0$ or 0×0 operations) in the matrix multiplications $\prod_{t=2}^{L} \Lambda(t)$ (or Λ^{L-1} when homogeneous FMCI is employed) of the FMCI formula. Hence, her method easily becomes computationally difficult when the sequence length L is large. A later suggestion by Fu was that, depending on the definition of the runs statistic of interest, a recursive relationship may be established for the FMCI formula in order to have more efficient computations. For example, the FMCI formulae for the single runs statistics $R_{l\uparrow}$, $R_{l_{)(}}$, and $R_{l_{\between}}$ can be improved computationally with an appropriate recursive implementation (Discussed in Fu, 1996 [56]).

Having reviewed optimization techniques for matrix computations, I have discovered that the non-productive operations in the matrix multiplications of the FMCI formula can often be avoided by adopting various dynamic programming principles and/or constructing well-designed dynamic data structures. Fu's suggestion of a recursive implementation is indeed a special case of making use of an inductive dynamic programming principle. We shall revise and improve Lou's work on the exact distribution of the double runs statistic (N_s, R) by introducing a more computationally efficient dynamic programming principle. Specifically, a different FMCI construction is achieved, and a recursive algorithm is derived and implemented for the computations of the probabilities $Pr((N_s, R) = (n, r))$ under an IID framework and an MC framework.

Since an IID framework is nested within an MC framework, the FMCI work and the recursive algorithm for computing $Pr(N_s, R)$ under an MC framework are presented in detail. For a sequence of L first-order homogeneous Markov dependent dichotomous trials $\{Y_t, \Omega_Y, \mathcal{A}_\mathcal{Y} : t \in \Gamma_L\}$, the double runs statistic (N_s, R) is imbedded into a finite MC by constructing the following:

- A homogeneous MC $\{Z_t = (N_{s_t}, R_{s_t}, Y_t) : t \in \Gamma_L\}$, where N_{s_t} is the number of successes in the first t trials, R_{s_t} is the number of success runs in the first t trials, and Y_t is the outcome of the t-th trial.

- A finite state space $\Omega = \{0, 0, F\} \cup \Big\{(n, r, y) : n \in \{1, \dots, L\};\ r \in \{1, \dots, r_{max}\};\ y \in \{S, F\};$ with $r \leq min(n, r_{max})$ and $(n + r) \leq (L + 1)\Big\}$, where $r_{max} = \begin{cases} \frac{L}{2} & \text{if } L \text{ is even,} \\ \frac{L+1}{2} & \text{if } L \text{ is odd.} \end{cases}$

  NOTE: $\Omega = \{s_1, \dots, s_h\}$ is a set of 3-tuple (n, r, y) states, and the number of elements in $\Omega = size(\Omega) = h = 1 + \sum_{n=1}^{L} n$.

- The transition probability matrix $\Lambda = [Pr(Z_t = z_t | Z_{t-1} = z_{t-1})]$ of the imbedded Markov chain:

  For $t = 2, \dots, L$;

  - $Pr(Z_t = (n, r, F) | Z_{t-1} = (n, r, S)) = Pr(Y_t = F | Y_{t-1} = S) = P_{SF}$.
  - $Pr(Z_t = (n, r, F) | Z_{t-1} = (n, r, F)) = \begin{cases} Pr(Y_t = F | Y_{t-1} = F) = P_{FF} & \text{if } (n + r) < L, \\ 1 & \text{if } (n + r) = L. \end{cases}$
  - $Pr(Z_t = (n + 1, r + 1, S) | Z_{t-1} = (n, r, F)) = Pr(Y_t = S | Y_{t-1} = F) = P_{FS}$.
  - $Pr(Z_t = (n + 1, r, S) | Z_{t-1} = (n, r, S)) = Pr(Y_t = S | Y_{t-1} = S) = P_{SS}$.
  - $Pr(Z_t = (n, r, S) | Z_{t-1} = (n, r, S)) = 1 \qquad \text{if } (n + r) = (L + 1)$.
  - Otherwise, the transition probabilities are zero.

- A finite partition $C_{(n,r)} = \begin{cases} \{(0, 0, F)\} & \text{if } n = r = 0, \\ \{(n, r, S)\} & \text{if } (n + r) = (L + 1), \\ \{(n, r, S), (n, r, F)\} & \text{otherwise,} \end{cases}$ on the state space Ω; such that $Pr((N_s, R) = (n, r)) = Pr(Z_L \in C_{(n,r)})$, for every (n, r) pair.

Then, with the initial probability of the imbedded Markov chain, $\pi^{FMCI} = (Pr(Z_1 = (0, 0, F)), Pr(Z_1 = (1, 1, S)), 0, \dots, 0)$, we have the following for every (n, r) pair:

$$Pr((N_s, R) = (n, r)) = Pr(Z_L \in C_{(n,r)}) = \pi^{FMCI} \Lambda^{L-1} U(C_{(n,r)}), \tag{4.14}$$

where $U(C_{(n,r)}) = \sum_{s_g \in C_{(n,r)}} U(s_g)$ with $U(s_g)$ being a $h \times 1$ unit column vector having a value of 1 at the g-th coordinate and zero otherwise.

For example, if $L = 5$, the state space $\Omega = \{(0,0,F),(1,1,S),(1,1,F),(2,1,S),(2,1,F),(2,2,S),$ $(2,2,F),(3,1,S),(3,1,F),(3,2,S),(3,2,F),(3,3,S),(4,1,S),(4,1,F),(4,2,S),(5,1,S)\}$, $size(\Omega) =$ $h = 1+\sum_{n=1}^{L=5} n = 1+(1+2+3+4+5) = 16$, and the transition probability matrix of the imbedded Markov chain is:

$$
\Lambda = \begin{array}{c|cccccccccccccccc}
Z_{t-1}\backslash Z_t & (0,0,F) & (1,1,S) & (1,1,F) & (2,1,S) & (2,1,F) & (2,2,S) & (2,2,F) & (3,1,S) & (3,1,F) & (3,2,S) & (3,2,F) & (3,3,S) & (4,1,S) & (4,1,F) & (4,2,S) & (5,1,S) \\
\hline
(0,0,F) & P_{FF} & P_{FS} & & & & & & & & & & & & & & \\
(1,1,S) & & & P_{SF} & P_{SS} & & & & & & & & & 0 & & & \\
(1,1,F) & & & P_{FF} & 0 & 0 & P_{FS} & & & & & & & & & & \\
(2,1,S) & & & & & P_{SF} & 0 & 0 & P_{SS} & & & & & & & & \\
(2,1,F) & & & & & P_{FF} & 0 & 0 & 0 & 0 & P_{FS} & & & & & & \\
(2,2,S) & & & & & & & P_{SF} & 0 & 0 & P_{SS} & & & & & & \\
(2,2,F) & & & & & & & P_{FF} & 0 & 0 & 0 & 0 & P_{FS} & & & & \\
(3,1,S) & & & & & & & & & P_{SF} & 0 & 0 & 0 & P_{SS} & & & \\
(3,1,F) & & & & & & & & & P_{FF} & 0 & 0 & 0 & 0 & 0 & P_{FS} & \\
(3,2,S) & & & & & & & & & & & P_{SF} & 0 & 0 & 0 & P_{SS} & \\
(3,2,F) & & & & & & & & & & & 1 & 0 & 0 & 0 & 0 & \\
(3,3,S) & & & & & & & & & & & & 1 & 0 & 0 & 0 & \\
(4,1,S) & & & & & & & & & & & & & 0 & P_{SF} & 0 & P_{SS} \\
(4,1,F) & & & & & & & & & & & & & & 1 & 0 & 0 \\
(4,2,S) & & & & 0 & & & & & & & & & & & 1 & 0 \\
(5,1,S) & & & & & & & & & & & & & & & & 1
\end{array}
$$

Viewing Λ carefully, the following special matrix structure is revealed:

$$\Lambda = \begin{bmatrix}
P_{FF} & P_{FS} & 0 & & & & & & & & & & & & & \\
 & 0 & P_{SF} & P_{SS} & 0 & 0 & 0 & & & & & & 0 & & & \\
 & 0 & P_{FF} & 0 & 0 & P_{FS} & 0 & & & & & & & & & \\
 & & & 0 & P_{SF} & 0 & 0 & P_{SS} & 0 & 0 & 0 & 0 & & & & \\
 & & & 0 & P_{FF} & 0 & 0 & 0 & 0 & P_{FS} & 0 & 0 & & & & \\
 & & & 0 & 0 & 0 & P_{SF} & 0 & 0 & P_{SS} & 0 & 0 & & & & \\
 & & & 0 & 0 & 0 & P_{FF} & 0 & 0 & 0 & 0 & P_{FS} & & & & \\
 & & & & & & & 0 & P_{SF} & 0 & 0 & 0 & P_{SS} & 0 & 0 & \\
 & & & & & & & 0 & P_{FF} & 0 & 0 & 0 & 0 & 0 & P_{FS} & \\
 & & & & & & & 0 & 0 & 0 & P_{SF} & 0 & 0 & 0 & P_{SS} & \\
 & & & & & & & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \\
 & & & & & & & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 & & & & & & & & & & & & 0 & P_{SF} & 0 & P_{SS} \\
 & & & 0 & & & & & & & & & 0 & 1 & 0 & 0 \\
 & & & & & & & & & & & & 0 & 0 & 1 & 0 \\
 & & & & & & & & & & & & & & & 1
\end{bmatrix}$$

In fact, the transition probability matrix Λ of this imbedded Markov chain has a form which can be viewed as a sum of two special matices: a *block-diagonal matrix* $\mathcal{D}$ with block-components $\mathcal{D}_n$'s being block-diagonal matrices themselves, where $n = 0, 1, \ldots, L$; and an *upper-block-step matrix* $\mathcal{U}$ with block-components $\mathcal{U}_n$'s, where $n = 0, 1, \ldots, L-1$. I.e.

$$\Lambda = \mathcal{D} + \mathcal{U} = \left[\begin{array}{ccccccccc}
\mathcal{D}_0 & \vdots & \mathcal{U}_0 & & & & & & \\
\cdots & \cdots & \cdots & & & & & & \\
 & & \mathcal{D}_1 & \vdots & \mathcal{U}_1 & & 0 & & \\
 & & & \vdots & & & & & \\
 & & \cdots & \cdots & \cdots & & & & \\
 & & & & \ddots & \ddots & & & \\
 & & & & \mathcal{D}_n & \mathcal{U}_n & & & \\
 & & & & & \ddots & \ddots & & \\
 & & & & & \cdots & \cdots & \cdots & \\
 & & & & & & & \vdots & \\
 & 0 & & & & & \mathcal{D}_{L-1} & \vdots & \mathcal{U}_{L-1} \\
 & & & & & & & \vdots & \\
 & & & & & & \cdots & \cdots & \cdots \\
 & & & & & & & & \mathcal{D}_L
\end{array}\right]; \tag{4.15}$$

where

$$\mathcal{D}_0 = P_{FF}, \quad \mathcal{D}_1 = \begin{bmatrix} 0 & P_{SF} \\ 0 & P_{FF} \end{bmatrix}, \quad \mathcal{D}_{L-1} = \begin{bmatrix} 0 & P_{SF} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathcal{D}_L = 1, \quad \text{and}$$

$$\mathcal{D}_n = \begin{cases} \begin{bmatrix} \mathcal{D}_1 & & 0 \\ & \ddots & \\ 0 & & \mathcal{D}_1 \end{bmatrix} & \text{if } 2 \le n < r_{max}\text{; (NOTE: This is a block-diagonal matrix with } n\ \mathcal{D}_1\text{'s.)} \\ \begin{bmatrix} \mathcal{D}_1 & & 0 & \vdots & \\ & \ddots & & \vdots & 0 \\ 0 & & \mathcal{D}_1 & \vdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & 0 & & \vdots & \mathcal{D}_{\blacktriangleleft} \end{bmatrix} & \text{if } r_{max} \le n < L-1\text{. (NOTE: This matrix has } (L-1-n)\ \mathcal{D}_1\text{'s \& an } \mathcal{D}_{\blacktriangleleft}\text{.)} \end{cases}$$

$$\text{NOTE: } \mathcal{D}_{\blacktriangleleft} = \begin{cases} \begin{bmatrix} 0 & P_{SF} \\ 0 & 1 \end{bmatrix} & \text{if } L \text{ is even,} \\ \mathcal{D}_{L-1} & \text{if } L \text{ is odd.} \end{cases}$$

together with

$$\mathcal{U}_0 = \begin{bmatrix} P_{FS} & 0 \end{bmatrix}, \quad \mathcal{U}_{\blacktriangleright} = \begin{bmatrix} P_{SS} & 0 \end{bmatrix}, \quad \mathcal{U}_{\blacksquare} = \begin{bmatrix} \mathcal{U}_0 \\ \mathcal{U}_{\blacktriangleright} \end{bmatrix}, \quad \mathcal{U}_1 = \begin{bmatrix} \mathcal{U}_{\blacktriangleright} & 0 \\ 0 & \mathcal{U}_0 \end{bmatrix}, \quad \mathcal{U}_{L-1} = \begin{bmatrix} P_{SS} \\ 0 \\ 0 \end{bmatrix}, \quad \text{and}$$

$$\mathcal{U}_n = \begin{bmatrix} \mathcal{U}_{\blacktriangleright} & \vdots & & & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ & \vdots & \mathcal{U}_{\blacksquare} & & \vdots & \\ & \vdots & & \ddots & \vdots & \\ & \vdots & & \mathcal{U}_{\blacksquare} & \vdots & \\ & & \cdots & \cdots & \cdots & \cdots \\ 0 & & & & \vdots & \mathcal{U}_{\blacktriangleleft} \end{bmatrix} \quad \text{if } 2 \le n \le L-2\text{. (NOTE: This matrix has an } \mathcal{U}_{\blacktriangleright}\text{, } num_{\blacksquare}\ \mathcal{U}_{\blacksquare}\text{'s, \& an } \mathcal{U}_{\blacktriangleleft}\text{.)}$$

$$\text{NOTE: } num_{\blacksquare} = \begin{cases} n-1 & \text{if } 2 \le n < r_{max}, \\ L-2-n & \text{if } r_{max} \le n \le L-2. \end{cases}$$

$$\text{NOTE: } \mathcal{U}_{\blacktriangleleft} = \begin{cases} \mathcal{U}_0 & \text{if } 2 \le n < r_{max}\text{, except when } L \text{ is odd and } n = r_{max}-1 \text{ then } \mathcal{U}_{\blacktriangleleft} = P_{FS}, \\ \begin{bmatrix} P_{FS} \\ p_{SS} \\ 0 \\ 0 \end{bmatrix} & \text{if } r_{max} \le n \le L-2. \end{cases}$$

Owing to the special structure of Λ and the fact that $\pi^{FMCI}\Lambda^{L-1} = \pi^{FMCI}\Lambda^{L-2}\Lambda$, a recursion can be established for the computation of the exact distribution of the imbedded double runs statistic (N_s, R). The recursive algorithm is derived as follows:

DEFINITION

For $t = 2, \ldots, L$ and $(n, r) \in \{0, \ldots, L\} \times \{0, \ldots, r_{max}\}$ with $r \le min(n, r_{max})$ and $(n + r) \le (L + 1)$;

(NOTE: $n = 0 \leftrightarrow r = 0 \leftrightarrow (n, r) = (0, 0)$.)

$$\zeta_t(n,r) = \begin{cases} Pr(Z_t = (0,0,F)) & \text{if } (n,r) = (0,0), \\ Pr(Z_t = (n,r,S)) & \text{if } (n+r) = (L+1), \\ \Big(Pr(Z_t = (n,r,S)), Pr(Z_t = (n,r,F))\Big) & \text{otherwise.} \end{cases}$$

and

$$\zeta_t(n) = \begin{cases} Pr(Z_t = (0,0,F)) & \text{if } n = 0, \\ \zeta_t(1,1) = \Big(Pr(Z_t = (1,1,S)), Pr(Z_t = (1,1,F))\Big) & \text{if } n = 1, \\ \Big(\zeta_t(n,1), \ldots, \zeta_t(n,r)\Big) & \text{if } 1 < n < L, \\ Pr(Z_t = (L,1,S)) & \text{if } n = L. \end{cases}$$

$\boxed{\text{DERIVATION}}$

For every (n,r) pair;

$$\begin{aligned} Pr((N_s,R)=(n,r)) &\overset{\text{FMCI def.}}{=} Pr(Z_L \in C_{(n,r)}) \\ &\qquad \text{(BY FMCI THEOREM)} \\ &= \pi^{FMCI}\Lambda^{L-1}U(C_{(n,r)}) \\ &= \pi^{FMCI}\Lambda^{L-2}\,\Lambda U(C_{(n,r)}) \\ &\qquad (\because \Lambda = \mathcal{D}+\mathcal{U}, \rightarrow \text{RECURSION}) \\ &= \zeta_L(n)\mathbf{1}^{(n,r)} \end{aligned} \tag{4.16}$$

NOTE: $\mathbf{1}^{(n,r)}$ is a $size(\zeta_L(n)) \times 1$ column vector having 1's at the coordinates corresponding to $\zeta_L(n,r)$.

The recursive algorithm consists of the following three steps.

1. **Initialization Step:**

   For $t=1$; (NOTE: This implies that $n=0$ or 1.)

$$\zeta_1(n) = \begin{cases} \zeta_1(0) = Pr(Z_1 = (0,0,F)) & \text{if } n=0, \\ \zeta_1(1) = \Big(Pr(Z_1=(1,1,S)),0\Big) & \text{if } n=1. \end{cases} \tag{4.17}$$

   REMARK: $\zeta_1(n)$'s are in fact coordinates of π^{FMCI}.

2. **Recursion/Induction Step:**

   For $t=2,\dots,L$ and $n \in \{0,\dots,L\}$;

$$\zeta_t(n) = \begin{cases} \zeta_{t-1}(0)\mathcal{D}_0 & \text{if } n=0, \\ \zeta_{t-1}(n-1)\mathcal{U}_{n-1} + \zeta_{t-1}(n)\mathcal{D}_n & \text{if } 1 \le n \le L. \end{cases} \tag{4.18}$$

3. **Termination Step:**

   For every (n,r) pair;

$$Pr((N_s,R)=(n,r)) = Pr(Z_L \in C_{(n,r)}) = \zeta_L(n)\mathbf{1}^{(n,r)} \tag{4.19}$$

NOTE: When $\{Y_t : t \in \Gamma_L\}$ are IID, we have $\pi^{FMCI} = (P_F, P_S, 0, \dots, 0)$, $P_{FF} = P_{SF} = P_F$, and $P_{FS} = P_{SS} = P_S$ in the transition probability matrix Λ of the imbedded MC. The same algorithm can be applied directly.

4.2.2 Using Conditional Runs Statistic to Test for Randomness Against Clustering

One of the most commonly used statistics for testing randomness in a sequence of dichotomous (e.g. success/failure) outcomes is the number of success runs given the number of successes, i.e. $(R|N_s)$. Since N_s is a sufficient statistic when we are dealing with IID dichotomous (or Bernoulli) trials, the distribution $Pr(R|N_s = n)$ does not depend on P_S and P_F for any given value n. The conditional runs statistic $(R|N_s)$ becomes an attractive test statistic for testing randomness in binary sequence data. Once the (joint) distribution of the double runs statistic (N_s, R) is available, the associated marginal and conditional distributions can be easily obtained. In particular, with the FMCI-based algorithm, we can compute the exact distribution $Pr(R|N_s = n)$ under an IID framework as follows:

$$Pr(R = r|N_s = n) = \frac{Pr(N_s = n, R = r)}{Pr(N_s = n)} = \frac{\zeta_L(n)\mathbf{1}^{(n,r)}}{\sum_{r'=0}^{r_{max}} \zeta_L(n)\mathbf{1}^{(n,r')}}. \tag{4.20}$$

Hence, for a given n, we can set up a success runs test for randomness (i.e. the null hypothesis H_0: Y_t's are IID) by finding the critical/rejection region(s) of the test at a desired level of significance. As an illustration, if $L = 51$, the conditional probabilities $Pr(R = r|N_s = n)$ under H_0 and the critical regions of the two-sided success runs tests at the individual 5%¶ significance level are displayed graphically in Figure 4.1. For example, when there are 20 successes in a sequence of 51 outcomes,

¶NOTE: Since all tests are based on discrete probability distributions, the tail probabilities may not be equal to the assigned significance level. An additional (e.g. a randomized test) procedure may be desired to correct for the continuity issue.

we will reject the hypothesis of random scattering of successes at the 5% significance level if the number of success runs is either smaller than 9 or greater than 15.

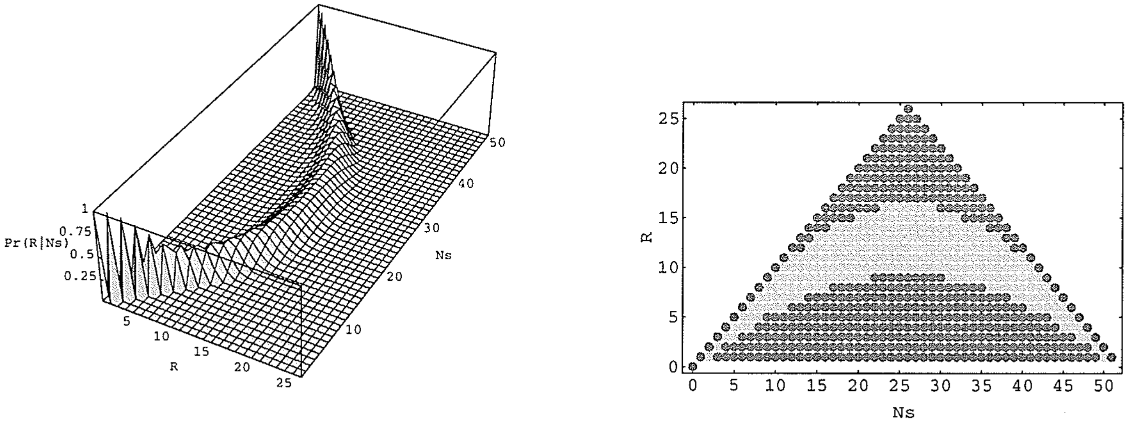


Figure 4.1: $Pr(R|N_s)$ for all possible N_s (Left) & the critical regions of the two-sided success runs tests at the individual 5% significance level (Right). (NOTE: Dark points correspond to the critical regions)

4.2.3 Distribution of a Double Runs Statistic Under an HMM Framework

Since there is a strong connection between an HMM structure and an MC structure, the FMCI technique can be adopted to study runs and patterns in a sequence of hidden Markov dependent outcomes $\{Y_t : t \in \Gamma_L\}$. As a demonstration, the connection between an HMM structure and an MC structure in its simplest form is explicitly established in the following lemma:

LEMMA

If a dichotomous outcome process $\{Y_t : t \in \Gamma_L\}$ has a hidden Markov model structure with its hidden state process $\{X_t, \Omega_X, \mathcal{A}_X : t \in \Gamma_L\}$ following a first-order homogeneous 2-state Markov chain, then the stochastic process $\{(X_t, Y_t) : t \in \Gamma_L\}$ can be represented as a first-order homogeneous Markov chain with its state space $\Omega_{XY} = \{(F,F), (F,S), (S,F), (S,S)\}$ and its transition probability matrix $\mathcal{A}_{XY}$.

NOTE:

$$\mathcal{A}_X = \begin{array}{c} X_{t-1}\backslash X_t \\ F \\ S \end{array} \begin{array}{c} \begin{array}{cc} F & S \end{array} \\ \left[\begin{array}{cc} a_{FF} & a_{FS} \\ a_{FS} & a_{SS} \end{array} \right] \end{array}, \quad \text{and}$$

$$\mathcal{A}_{XY} = \begin{array}{c|cccc} (X_{t-1}, Y_{t-1})\backslash(X_t, Y_t) & (F,F) & (F,S) & (S,F) & (S,S) \\ \hline (F,F) & p_{FFFF} & p_{FFFS} & p_{FFSF} & p_{FFSS} \\ (F,S) & p_{FSFF} & p_{FSFS} & p_{FSSF} & p_{FSSS} \\ (S,F) & p_{SFFF} & p_{SFFS} & p_{SFSF} & p_{SFSS} \\ (S,S) & p_{SSFF} & p_{SSFS} & p_{SSSF} & p_{SSSS} \end{array}.$$

PROOF

$$Pr((X_t, Y_t) = (x_t, y_t)|(X_{t-1}, Y_{t-1}) = (x_{t-1}, y_{t-1}), \dots, (X_1, Y_1) = (x_1, y_1))$$

$$= Pr(X_t = x_t, Y_t = y_t | X_{[1,t-1]} = x_{[1,t-1]}, Y_{[1,t-1]} = y_{[1,t-1]})$$

$$= Pr(Y_t = y_t | X_{[1,t]} = x_{[1,t]}, Y_{[1,t-1]} = y_{[1,t-1]})\, Pr(X_t = x_t | X_{[1,t-1]} = x_{[1,t-1]}, Y_{[1,t-1]} = y_{[1,t-1]})$$

($\because$ $\{X_t : t \in \Gamma_L\}$ is a 1st-order homogeneous MC, and

X_t & $Y_{[1,t-1]}$ are conditionally independent given X_{t-1}.)

$$= Pr(Y_t = y_t | X_t = x_t, X_{[1,t-1]} = x_{[1,t-1]}, Y_{[1,t-1]} = y_{[1,t-1]})\, Pr(X_t = x_t | X_{t-1} = x_{t-1})$$

($\because$ Y_t & $(X_{[1,t-1]}, Y_{[1,t-1]})$ are conditionally independent given X_t.)

$$= Pr(Y_t = y_t | X_t = x_t)\, Pr(X_t = x_t | X_{t-1} = x_{t-1})$$

($\because$ Y_t & (X_{t-1}, Y_{t-1}) are conditionally independent given X_t, and

X_t & Y_{t-1} are conditionally independent given X_{t-1}.)

$$= Pr(Y_t = y_t | X_t = x_t, X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1})\, Pr(X_t = x_t | X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1})$$

$$= Pr(X_t = x_t, Y_t = y_t | X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1}) \overset{\text{denoted as}}{\Longrightarrow} P_{x_{t-1}y_{t-1}x_t y_t}$$

$\therefore$ $\{(X_t, Y_t)\}$ is a 1st-order homogeneous MC. $\square$

NOTE: In terms of the binary HMM parameters a_{ij}'s and $b_{j(k)}$'s where $i, j, k \in \{S, F\}$, we have:

$$\mathcal{A}_{XY} = \begin{array}{c} (X_{t-1}, Y_{t-1})\backslash(X_t, Y_t) \\ (F,F) \\ (F,S) \\ (S,F) \\ (S,S) \end{array} \begin{array}{c} \begin{array}{cccc} (F,F) & (F,S) & (S,F) & (S,S) \end{array} \\ \left[\begin{array}{cccc} a_{FF}\, b_{F(F)} & a_{FF}\, b_{F(S)} & a_{FS}\, b_{S(F)} & a_{FS}\, b_{S(S)} \\ a_{FF}\, b_{F(F)} & a_{FF}\, b_{F(S)} & a_{FS}\, b_{S(F)} & a_{FS}\, b_{S(S)} \\ a_{SF}\, b_{F(F)} & a_{SF}\, b_{F(S)} & a_{SS}\, b_{S(F)} & a_{SS}\, b_{S(S)} \\ a_{SF}\, b_{F(F)} & a_{SF}\, b_{F(S)} & a_{SS}\, b_{S(F)} & a_{SS}\, b_{S(S)} \end{array}\right] \end{array}.$$

With this lemma, we can use the FMCI technique to obtain distributions of various runs-related statistics under a binary HMM framework by working with $\{(X_t, Y_t), \Omega_{XY}, \mathcal{A}_{XY} : t \in \Gamma_L\}$. For instance, by replacing the univariate outcome random variable Y_t in Subsection §4.2.1 with the bivariate random variable (X_t, Y_t) for all t, we can simply adjust the FMCI construction work in Subsection §4.2.1 and obtain $Pr(N_s, R)$ under a binary HMM framework. The states of the imbedded MC become 4-tuple (n, r, x, y) states. The recursive algorithm presented earlier can also be used with appropriate modifications.

For example, if $L = 5$, the state space $\Omega = \{(0,0,F,F), (0,0,S,F), (1,1,F,S), (1,1,S,S), (1,1,F,F), (1,1,S,F), (2,1,F,S), (2,1,S,S), (2,1,F,F), (2,1,S,F), (2,2,F,S), (2,2,S,S), (2,2,F,F), (2,2,S,F), (3,1,F,S), (3,1,S,S), (3,1,F,F), (3,1,S,F), (3,2,F,S), (3,2,S,S), (3,2,F,F), (3,2,S,F), (3,3,F,S), (3,3,S,S), (4,1,F,S), (4,1,S,S), (4,1,F,F), (4,1,S,F), (4,2,F,S), (4,2,S,S), (5,1,F,S), (5,1,S,S)\}$, $size(\Omega) = 2(1 + \sum_{n=1}^{L=5} n) = 32$, and

$$\Lambda = \left[\begin{array}{cccccccccccccccc}
\Lambda_{FF} & \Lambda_{FS} & & & & & & & & & & & & & & \\
 & & \Lambda_{SF} & \Lambda_{SS} & & & & & & & & & \mathcal{O}_2 & & & \\
 & & \Lambda_{FF} & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{FS} & & & & & & & & & & \\
 & & & & \Lambda_{SF} & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{SS} & & & & & & & & \\
 & & & & \Lambda_{FF} & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{FS} & & & & & & \\
 & & & & & & \Lambda_{SF} & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{SS} & & & & & & \\
 & & & & & & \Lambda_{FF} & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{FS} & & & & \\
 & & & & & & & & \Lambda_{SF} & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{SS} & & & \\
 & & & & & & & & \Lambda_{FF} & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{FS} & \\
 & & & & & & & & & & \Lambda_{SF} & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \Lambda_{SS} & \\
 & & & & & & & & & & \mathcal{I}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \\
 & & & & & & & & & & & \mathcal{I}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \mathcal{O}_2 & \\
 & & & & & & & & & & & & \mathcal{O}_2 & \Lambda_{SF} & \mathcal{O}_2 & \Lambda_{SS} \\
 & & & & & & & & & & & & & \mathcal{I}_2 & \mathcal{O}_2 & \mathcal{O}_2 \\
 & & & \mathcal{O}_2 & & & & & & & & & & & \mathcal{I}_2 & \mathcal{O}_2 \\
 & & & & & & & & & & & & & & & \mathcal{I}_2
\end{array}\right]$$

where

$$\Lambda_{FF} = \begin{bmatrix} p_{FFFFF} & p_{FFFSF} \\ p_{SFFFF} & p_{SFFSF} \end{bmatrix}, \quad \Lambda_{FS} = \begin{bmatrix} p_{FFFFS} & p_{FFFSS} \\ p_{SFFFS} & p_{SFFSS} \end{bmatrix}, \quad \Lambda_{SF} = \begin{bmatrix} p_{FSFFF} & p_{FSSSF} \\ p_{SSSFF} & p_{SSSSF} \end{bmatrix}, \quad \Lambda_{SS} = \begin{bmatrix} p_{FSFFS} & p_{FSSSS} \\ p_{SSSFS} & p_{SSSSS} \end{bmatrix},$$

and

$$\mathcal{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathcal{O}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Being able to obtain the exact $Pr(N_s, R)$ under a binary HMM framework, this work establishes an investigation of the double runs statistic (N_s, R) in a binary sequence under a binary HMM. Having studied the distributions of the double runs statistic under different binary HMM parameter sets, probabilistic profiles of (N_s, R) are created. With these probabilistic profiles, issues of HMM parameter estimation are addressed and examined in the following section.

4.3 Examining HMM Parameter Estimation with Probabilistic Profiles of Runs

4.3.1 Identifiability of HMMs

Strictly speaking, the parameters of a hidden Markov model are not identifiable. The likelihood function $\mathcal{L}_Y(\boldsymbol{\theta})$ can have the same value with different values of $\boldsymbol{\theta}$, i.e. $\mathcal{L}_Y(\boldsymbol{\theta}_1) = \mathcal{L}_Y(\boldsymbol{\theta}_2)$ where $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$. This implies that maximum likelihood estimator of the parameter set $\boldsymbol{\theta}$ of an HMM may not be unique. Well-known HMM identifiability problems are due to label-switching of the hidden states or having non-distinct outcome-emission distributions from different hidden states. As reported by Churchill, HMM identifiability issues will also arise when $a_{01} + a_{10} = 1$ in a binary HMM (Churchill, 1998 [34]). Identifiability problems will make statistical inferences unstable and unpredictable. Hence, sensible constraints must be placed on the parameters to avoid identifiability problems in practice.

From an interpretative point of view, it is worth pointing out that the hidden state process of a binary HMM will no longer be Markovian when $a_{01} + a_{10} = 1$ (or equivalently $a_{00} + a_{11} = 1$). Although the first key HMM assumption (i.e. $X_{[1,L]}$ is a Markov chain) implicitly prevents $a_{01} + a_{10} = 1$ in a binary HMM from happening, we may explicitly ensure the hidden state process to be strictly Markovian by imposing constraints on the transition probability matrix $\mathcal{A}_X$, e.g. $a_{01} + a_{10} \neq 1$ (or equivalently $a_{00} + a_{11} \neq 1$) in a binary HMM. In general, constraining the stochastic $N \times N$ matrix $\mathcal{A}_X$ to have N distinct rows will ensure the hidden state process of a general HMM to be strictly Markovian, and will help avoid label-switching of the hidden states. In addition, by imposing an increasing (or decreasing) order on the outcome-emission probabilities, e.g. $b_{0(0)} > b_{1(0)}$ or equivalently $b_{1(1)} > b_{0(1)}$ in a binary HMM, we can avoid having non-distinct outcome-emission distributions from different hidden states.

As a demonstration, HMM identifiability problems are illustrated through probabilistic profiling of the double runs statistic (N_s, R) under an unidentifiable binary HMM with $\pi_0 = a_{00} = a_{11} = 0.5$ being fixed. Specifically, probabilistic profiles of (N_s, R) are graphically shown as contour plots of different probability distributions of (N_s, R) under different $(b_{0(0)}, b_{1(1)})$ sets with $\pi_0 = a_{00} = a_{11} = 0.5$ of a binary HMM (Refer to the first five panels of Figure 4.2). Since the hidden states of a binary HMM can switch labels freely when $\pi_0 = a_{00} = a_{11} = 0.5$, the likelihood function $\mathcal{L}_Y(\boldsymbol{\theta})$ can always have the exact same value with different values of $\boldsymbol{\theta}$, e.g. $\mathcal{L}_Y(\pi_0 = a_{00} = a_{11} = 0.5, b_{0(0)} = 0.1, b_{1(1)} = 0.1)$ is exactly the same as $\mathcal{L}_Y(\pi_0 = a_{00} = a_{11} = 0.5, b_{0(0)} = 0.9, b_{1(1)} = 0.9)$. Subsequently, as we can see in the sixth panel of Figure 4.2 and Tables 4.1, & 4.2, parameters $b_{0(0)}$ and $b_{1(1)}$ are not identifiable, and the unidentifiableness is reflected in the probabilistic profiles of the double runs statistic (N_s, R).

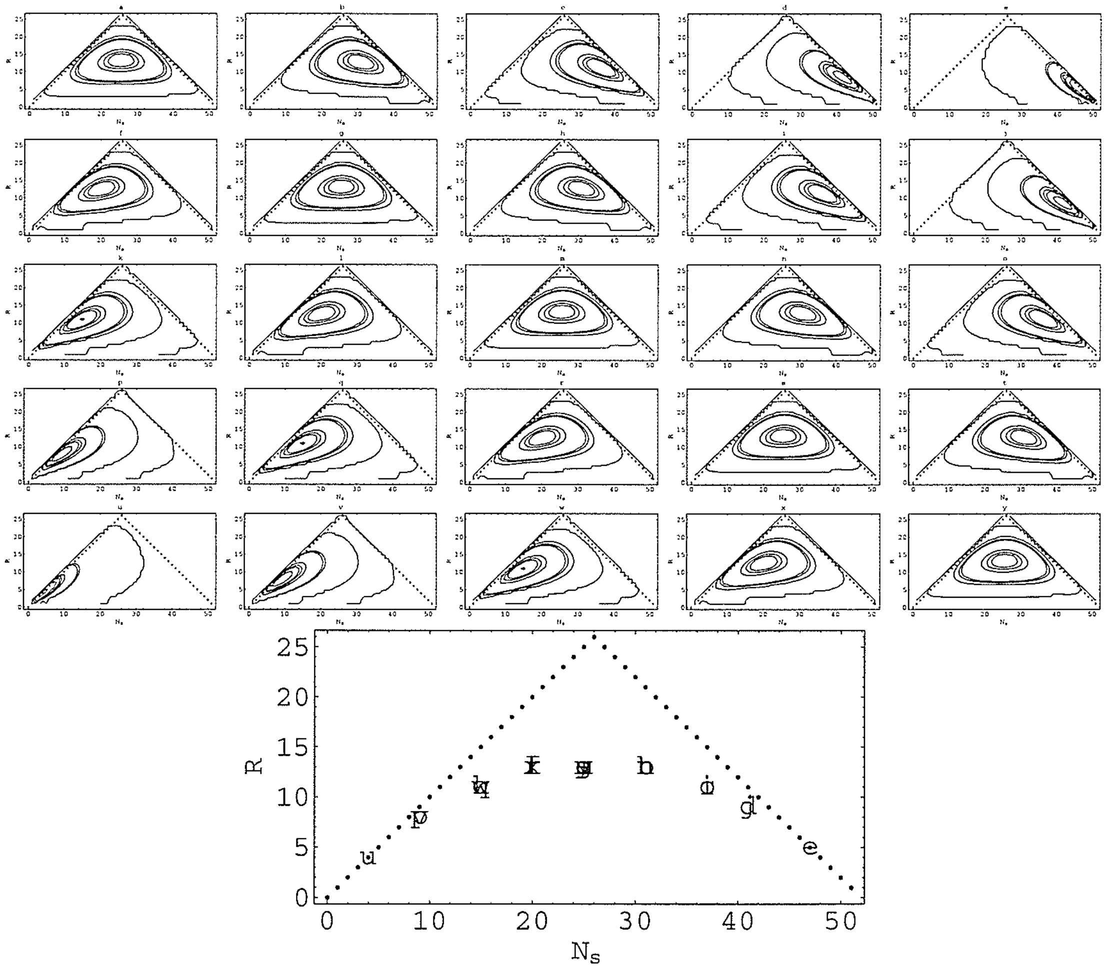


Figure 4.2: Probability Profiles of (N_s, R) under an Unidentifiable HMM: Contour plots of the probability distributions of (N_s, R) under different $(b_{0(0)}, b_{1(1)})$ sets with $\pi_0 = a_{00} = a_{11} = 0.5$ of a binary HMM. **First Panel** — $b_{0(0)} = 0.1$ with $b_{1(1)} = 0.1$ (a), 0.3 (b), 0.5 (c), 0.7 (d) or 0.9 (e). **Second Panel** — $b_{0(0)} = 0.3$ with $b_{1(1)} = 0.1$ (f), 0.3 (g), 0.5 (h), 0.7 (i) or 0.9 (j). **Third Panel** — $b_{0(0)} = 0.5$ with $b_{1(1)} = 0.1$ (k), 0.3 (l), 0.5 (m), 0.7 (n) or 0.9 (o). **Fourth Panel** — $b_{0(0)} = 0.7$ with $b_{1(1)} = 0.1$ (p), 0.3 (q), 0.5 (r), 0.7 (s) or 0.9 (t). **Fifth Panel** — $b_{0(0)} = 0.9$ with $b_{1(1)} = 0.1$ (u), 0.3 (v), 0.5 (w), 0.7 (x) or 0.9 (y). **Sixth Panel** — Overlaying maxima (labeled as a-y) of the contour plots. (NOTE: Length of sequence = $L = 51$).

Table 4.1: Illustration of an Unidentifiable Binary HMM with $\pi_0 = a_{00} = a_{11} = 0.5$ and $b_{0(0)} = b_{1(1)}$ or $b_{0(0)} \pm 0.2 = b_{1(1)}$ ($L = 51$)

Contour Label	$b_{0(0)}$	$b_{1(1)}$	Most likely (n_s, r)	Corresponding $Pr(n_s, r)$
(a)	0.1	0.1	(25, 13)	0.0240877417100549
(g)	0.3	0.3	(25, 13)	0.0240877417100549
(m)	0.5	0.5	(25, 13)	0.0240877417100549
(s)	0.7	0.7	(25, 13)	0.0240877417100549
(y)	0.9	0.9	(25, 13)	0.0240877417100549
(b)	0.1	0.3	(31, 13)	0.0256692837557371
(h)	0.3	0.5	(31, 13)	0.0256692837557371
(n)	0.5	0.7	(31, 13)	0.0256692837557371
(t)	0.7	0.9	(31, 13)	0.0256692837557371
(f)	0.3	0.1	(20, 13)	0.0255277488578358
(l)	0.5	0.3	(20, 13)	0.0255277488578358
(r)	0.7	0.5	(20, 13)	0.0255277488578358
(x)	0.9	0.7	(20, 13)	0.0255277488578358

Table 4.2: Illustration of an Unidentifiable Binary HMM with $\pi_0 = a_{00} = a_{11} = 0.5$ and $b_{0(0)} \pm 0.4 = b_{1(1)}$ or $b_{0(0)} \pm 0.6 = b_{1(1)}$ ($L = 51$)

Contour Label	$b_{0(0)}$	$b_{1(1)}$	Most likely (n_s, r)	Corresponding $Pr(n_s, r)$
(c)	0.1	0.5	(37, 11)	0.0308042605010935
(i)	0.3	0.7	(37, 11)	0.0308042605010935
(o)	0.5	0.9	(37, 11)	0.0308042605010935
(k)	0.5	0.1	(15, 11)	0.0325645039582989
(q)	0.7	0.3	(15, 11)	0.0325645039582989
(w)	0.9	0.5	(15, 11)	0.0325645039582989
(d)	0.1	0.7	(41, 9)	0.0460579825008322
(j)	0.3	0.9	(41, 9)	0.0460579825008322
(p)	0.7	0.1	(9, 8)	0.0505280921879815
(v)	0.9	0.3	(9, 8)	0.0505280921879815

4.3.2 Trapping HMM Parameter Estimates and DNA Decoding

The maximum likelihood approach has been widely used to deal with the hidden Markov model training (i.e. estimation of $\boldsymbol{\theta}$) in different forms of the DNA decoding problem or the problem of locating functional domains and/or genes in DNA. As discussed earlier in §3.2.4 (p.43–), the Expectation-Maximization algorithm has been a common solution for finding the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ of the parameter set of an HMM. Since the rate of convergence of the EM algorithm is only linear* in the vicinity of the MLE and the likelihood surface of an HMM is generally multimodal, the EM algorithm is quite vulnerable from its slow convergence and the problem of landing on a local maximum or even a saddle point in HMM training. Many variants of the EM algorithm have been recently proposed to help speed up the convergence (e.g. Liu & Rubin, 1994 [95]; Jamshidian & Jennrich, 1997 [71]; Meng & van Dyk, 1997 [102]; McLachlan & Krishnan, 1997 [101]; Liu, Rubin & Wu, 1998 [96]), but the problem of landing on a local maximum or a saddle point still highly depends upon the initial estimate which starts the algorithm. Although senselessly selecting initial estimates over the entire parameter space can be a brute-force way to find $\widehat{\boldsymbol{\theta}}$, the number of initial estimates can easily be prohibitively high. In this subsection, a novel idea of trapping the MLE of an HMM based on probabilistic profiles of runs is introduced as an enhancement for the EM algorithm in the HMM parameter estimation. As an establishment, an MLE-trapping scheme based on probabilistic profiles of the double runs statistic (N_s, R) is developed to enhance the EM algorithm in training a binary HMM.

Since the HMM parameters of interest are often a_{ij}'s and $b_{j(k)}$'s where $i, j \in \{1, \dots, N\} \& k \in \{1, \dots, M\}$, a slightly different implementation of the EM algorithm suggested by Leroux and Puterman is adopted in this work. Following the proposal of Leroux and Puterman, the maximization task of the EM algorithm is completed by solving the N individual lower-dimensional maximization problems defined by starting from a fixed initial state. In other words, we take the initial probability

*NOTE: The rate of convergence or equivalently the speed of convergence of the EM algorithm depends on the proportion of "missing data/information" in the prescribed EM framework.

of each of the N states of the hidden Markov chain in turn to be one, then solve the corresponding N lower-dimensional maximization problems respectively, and choose the initial state which gives the largest maximized $\log \mathcal{L}_{YX}(\boldsymbol{\theta})$ in the end (Leroux & Puterman, 1992 [90]). Hence, the parameter set $\boldsymbol{\theta} = (\pi_i$'s, a_{ij}'s, $b_{j(k)}$'s) is indeed treated as $\boldsymbol{\theta} = (a_{ij}$'s, $b_{j(k)}$'s) in the HMM estimation procedure.

To avoid HMM identifiability problems and to prepare for our applications of binary HMMs to DNA pattern recognition, we impose constraints $a_{00} > 0.5$ and $a_{11} > 0.5$ on the state-transition probabilities and the constraint $b_{0(0)} > b_{1(0)}$ (or equivalently $b_{0(0)} > 1 - b_{1(1)} \Leftrightarrow b_{1(1)} > b_{0(1)}$) on the outcome-emission probabilities in this work. These constraints are sensible for our applications of binary HMMs to recognize the start sites of transcription of RNA polymerase II transcribed genes (Details in section §4.4 p.102–). With these identifiability constraints, the machinery developed in §4.2.3 (p.76–) is used to calculate the exact probability distributions of the double runs statistic (N_s, R) under different binary HMM parameter sets. Having studied $Pr(N_s, R)$ under various combinations of different values of $a_{00}, a_{11}, b_{0(0)}$, & $b_{1(1)}$, extensive probabilistic profiles of (N_s, R) are created and used to form the basis of the MLE-trapping scheme. For example, we can create probabilistic profiles of (N_s, R) under $2 \times 25 \times 36 = 1800$ binary HMM sets (i.e. $\pi_0 = 0.0$ or 1.0 with a_{00} & a_{11} equal to either $0.55, 0.65, 0.75, 0.85$, or 0.95 and $b_{0(0)}$ & $b_{1(1)}$ equal to any one of the 36 combinations in Table 4.3).

Essentially, the MLE-trapping scheme is to use a trapping grid (or trapping grids), which is (are) built from probabilistic profiles of a runs statistic, to locate the neighbourhood of the MLE of $\boldsymbol{\theta}$ (i.e. $\widehat{\boldsymbol{\theta}}$). It is worth mentioning that, ideally, we would have to have an at least 4-tuple statistic (e.g. a well-designed quadruple[†] runs statistic) to sufficiently correspond to the 4-tuple parameter $\boldsymbol{\theta}$ (i.e. $a_{00}, a_{11}, b_{0(0)}, b_{1(1)}$) of a binary HMM. If probabilistic distributions/profiles of such a sufficient 4-tuple statistic under different binary HMM parameter sets are available, we could then build a four-dimensional trapping grid to locate the neighbourhood of $\widehat{\boldsymbol{\theta}}$. With probabilistic profiles of the double runs statistic (N_s, R), a "less-than-ideal" MLE-trapping scheme is developed through the use

[†] NOTE: Studies of elaborate multiple runs-related statistics under an HMM through the FMCI technique are left for future research.

Table 4.3: Examples of Binary HMM Parameter Sets with $\pi_0 = 0.0$ or 1.0 under Identifiability Constraints $a_{00} > 0.5$ and $a_{11} > 0.5$ on the State-Transition Probabilities and the Constraint $b_{0(0)} > b_{1(0)}$ (or Equivalently $b_{0(0)} > 1 - b_{1(1)}$) on the Outcome-Emission Probabilities

a_{00}	a_{11}	$b_{0(0)}$	$b_{1(1)}$
0.55	0.55	0.2	0.9
		0.3	0.8 or 0.9
		0.4	0.7, 0.8 or 0.9
		0.5	0.6, 0.7, 0.8 or 0.9
		0.6	0.5, 0.6, 0.7, 0.8 or 0.9
		0.7	0.4, 0.5, 0.6, 0.7, 0.8 or 0.9
		0.8	0.3, 0.4, 0.5, 0.6, 0.7, 0.8 or 0.9
		0.9	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 or 0.9
0.55	0.65	(same 36 combinations of $b_{0(0)}$ & $b_{1(1)}$)	
⋮	⋮		⋮
⋮	⋮		⋮
⋮	⋮		⋮
0.95	0.95	(same 36 combinations of $b_{0(0)}$ & $b_{1(1)}$)	

of many pieces of different two-dimensional (2-D) trapping grids instead. Specifically, we suppose that $\widehat{a_{00}}$ and $\widehat{a_{11}}$ (or any two of the four components in $\widehat{\boldsymbol{\theta}}$) are known to be $\widetilde{a_{00}}$ and $\widetilde{a_{11}}$, then a 2-D trapping grid conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}})$ is built to locate the neighbourhood of $\widehat{b_{0(0)}}$ and $\widehat{b_{1(1)}}$ (or the two unknown components in $\widehat{\boldsymbol{\theta}}$). For any sequence length L, the grid points of a 2-D trapping grid are defined with respect to the double runs statistic (N_s, R) as follows:

$\boxed{\textbf{DEFINITION}}$

- The horizontal-axis of a 2-D trapping grid represents the number of successes (i.e. N_s), and the vertical-axis of the grid represents the number of success runs (i.e. R).

- The center point of a trapping grid is $(N_s, R) = (N_{s(\frac{1}{2})}, R_{(\frac{1}{2})})$, where $N_{s(\frac{1}{2})} = \mathit{Floor}\left(\frac{L+1}{2}\right)$ and $R_{(\frac{1}{2})} = \mathit{Floor}\left(\frac{N_{s(\frac{1}{2})}}{2}\right)$.

- More generally, grid points are defined with $N_s = N_{s(\frac{1}{\sharp_n})}, N_{s(\frac{2}{\sharp_n})}, \ldots, N_{s(\bigstar)}, \ldots, N_{s(\frac{\sharp_n - 1}{\sharp_n})}$ and $R = R_{(\frac{1}{\sharp_r})}, R_{(\frac{2}{\sharp_r})}, \ldots, R_{(\bigstar)}, \ldots, R_{(\frac{\sharp_r - 1}{\sharp_r})}$, where $\sharp_n$ & $\sharp_r$ are positive integers and $N_{s(\bigstar)} = \mathit{Floor}\left[(L+1)\bigstar\right]$ & $R_{(\bigstar)} = \mathit{Floor}\left[\mathit{min}\left(N_s, L+1-N_s\right)\bigstar\right]$.

  NOTE: $r \leq \mathit{min}(n_s, r_{max})$ and $(n_s + r) \leq (L+1)$, where $r_{max} = \begin{cases} \frac{L}{2} & \text{if } L \text{ is even,} \\ \frac{L+1}{2} & \text{if } L \text{ is odd.} \end{cases}$

  The $\bigstar$ symbol denotes a positive fraction and $\mathit{Floor}\,(\cdot)$ is an operator that rounds its operand to the largest integer not exceeding the operand.

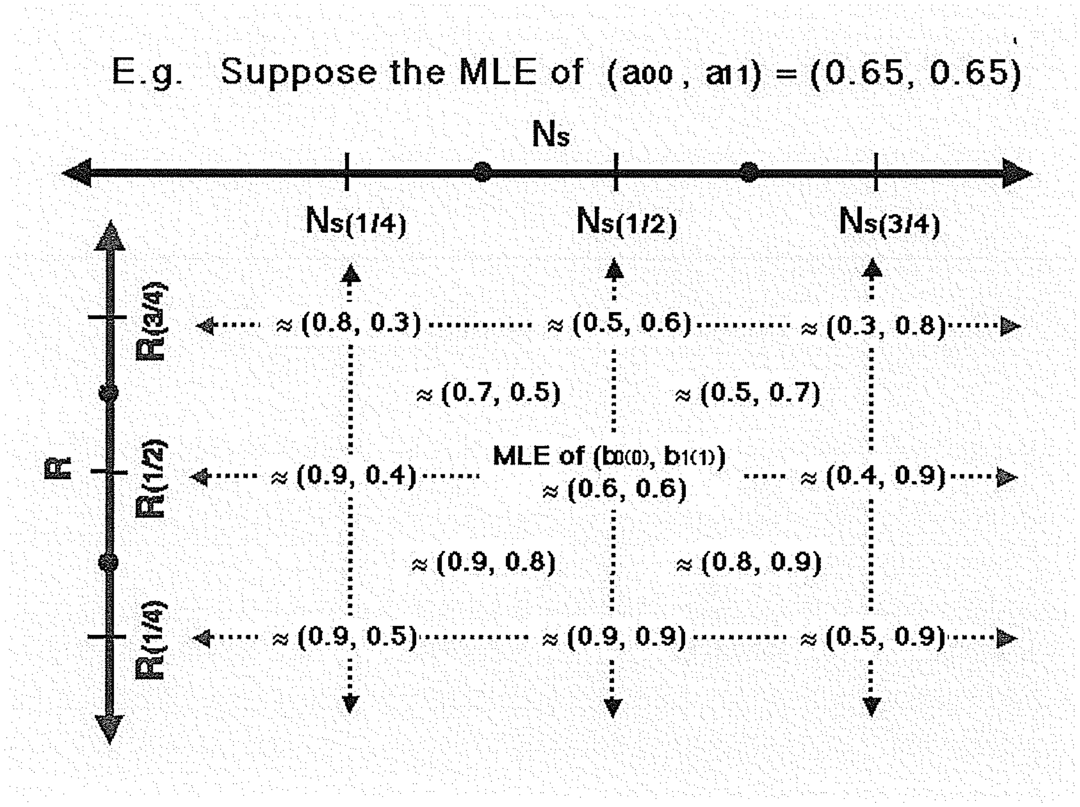
In a 2-D trapping grid conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}})$, each specific grid point $(N_s = n_s, R = r)$ is linked to the pair of $b_{0(0)}$ and $b_{1(1)}$, say $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$, which results the highest value of $Pr(N_s = n_s, R = r)$ among the probabilistic profiles of (N_s, R) created under various combinations of different values of $b_{0(0)}$ and $b_{1(1)}$ with the same $(\widetilde{a_{00}}, \widetilde{a_{11}})$. As expected, at least in the intuitive sense, we have found that $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$ is indeed close to $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$ conditioning on a fixed $(\widetilde{a_{00}}, \widetilde{a_{11}})$. This finding is supported by our investigations based on simulated sequences. Binary sequences are simulated under different binary HMMs with specific $\boldsymbol{\theta}$'s, then $\widehat{\boldsymbol{\theta}}$'s are obtained by using a large number of different initial estimates (e.g. using the $25 \times 36 = 900$ combinations of different values of $a_{00}, a_{11}, b_{0(0)}$, & $b_{1(1)}$ in Table 4.3 to form a set of initial estimates) for the EM algorithm to avoid landing on local maxima. By comparing $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$ from the 2-D trapping grid conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}})$ with $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$ from the EM algorithm, we find the difference is about ± 0.1. This difference can also be decreased if the 2-D trapping grid is built from even more extensive probabilistic profiles of (N_s, R).

Hence, once the double runs statistic of a sequence of outcomes is observed, the 2-D trapping grid can be used as a compass to locate the neighbourhood of $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$ conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}})$. As an illustration, a 2-D trapping grid conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}}) = (0.65, 0.65)$ is shown in Figure 4.3. If we observe the double runs statistic equals to the center point of the grid, i.e. $(N_s = N_{s(1/2)}, R = R_{(1/2)})$, then the neighbourhood of $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$ conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}}) = (0.65, 0.65)$ is around $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}}) = (0.6, 0.6)$.

Since the grid points (N_s, R) are expressed as different fractions of the sequence length L, a trapping grid can be re-used (or partially re-used) for different values of L. For example, the trapping grid in Figure 4.3 can be used for $L = 51, 101, 501$ etc. (i.e. if we observe the double runs statistic equals to the center point[‡] $(N_s = N_{s(1/2)}, R = R_{(1/2)})$, then the neighbourhood of $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$ conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}}) = (0.65, 0.65)$ is around $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}}) = (0.6, 0.6)$ for any L). Similarly, conditioning on a different $(\widetilde{a_{00}}, \widetilde{a_{11}})$ pair, we can build a different 2-D trapping grid. Having studied different trapping grids, a general linking relationship between (N_s, R) and $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$ is found. That is, for any particular value of $N_s = n_s$, if R increases then at least one of the two components in $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$ decreases. In addition, a special "symmetric" linking relationship is revealed when $\widetilde{a_{00}} = \widetilde{a_{11}}$. That is, for any particular value of $R = r$, if $(N_s = N_{s(\bigstar)}, R = r)$ is linked to $(\widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$ then $(N_s = N_{s(1-\bigstar)}, R = r)$ is linked to the pair $(b_{0(0)} = \widetilde{b_{1(1)}}, b_{1(1)} = \widetilde{b_{0(0)}})$. For example, $(N_s = N_{s(1/4)}, R = R_{(1/4)})$ is linked to $(0.9, 0.5)$ and $(N_s = N_{s(3/4)}, R = R_{(1/4)})$ is linked to $(0.5, 0.9)$ in Figure 4.3.

Since a trapping grid can be used as a compass to locate the neighbourhood of $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$

[‡] NOTE: The center grid points for $L = 51, 101, 501$ are $(N_s = 26, R = 13)$, $(N_s = 51, R = 25)$, and $(N_s = 251, R = 125)$ respectively.



E.g. Suppose the MLE of (a00 , a11) = (0.65, 0.65)

Ns

Ns(1/4)

Ns(1/2)

Ns(3/4)

R

R(3/4)

R(1/2)

R(1/4)

≈ (0.8, 0.3) ≈ (0.5, 0.6) ≈ (0.3, 0.8)

≈ (0.7, 0.5) ≈ (0.5, 0.7)

≈ (0.9, 0.4) MLE of (b0(0), b1(1)) ≈ (0.6, 0.6) ≈ (0.4, 0.9)

≈ (0.9, 0.8) ≈ (0.8, 0.9)

≈ (0.9, 0.5) ≈ (0.9, 0.9) ≈ (0.5, 0.9)

Figure 4.3: Trapping Grid for $(\widehat{b_{0(0)}}, \widehat{b_{1(1)}})$ Conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}}) = (0.65, 0.65)$. (NOTE: $N_{s(1/4)} = Floor\left(\frac{L+1}{4}\right), N_{s(1/2)} = Floor\left(\frac{L+1}{2}\right), N_{s(3/4)} = Floor\left(3\frac{L+1}{4}\right), R_{(1/4)} = Floor\left[\frac{min(N_s, L+1-N_s)}{4}\right], R_{(1/2)} = Floor\left[\frac{min(N_s, L+1-N_s)}{2}\right]$, and $R_{(3/4)} = Floor\left[3\frac{min(N_s, L+1-N_s)}{4}\right]$. Probabilistic profiles of (N_s, R) for this trapping grid are created under the 36 combinations of different values of $b_{0(0)}$ and $b_{1(1)}$ in Table 4.3 with $(a_{00}, a_{11}) = (0.65, 0.65)$ and the sequence length $L = 501$.)

conditioning on $(\widetilde{a_{00}}, \widetilde{a_{11}})$, it is sound to collect the 4-tuple $(\widetilde{a_{00}}, \widetilde{a_{11}}, \widetilde{b_{0(0)}}, \widetilde{b_{1(1)}})$ from each trapping grid to form a set of initial estimates for the EM algorithm in training a binary HMM. In other words, these trapping grids indeed serve as screening tools for good initial estimates for the EM algorithm. Hence, instead of using senseless combinations of different values of $a_{00}, a_{11}, b_{0(0)}$, & $b_{1(1)}$ as initial estimates for the EM algorithm, the MLE-trapping scheme offers screened initial estimates that can substantially push the EM algorithm to jump-start in the neighbourhood of $\widehat{\boldsymbol{\theta}}$. For example, as mentioned earlier, the set of initial EM estimates senselessly selected from 25 × 36 = 900 combinations of different values of $a_{00}, a_{11}, b_{0(0)}$, & $b_{1(1)}$ in Table 4.3 can be shrunk dramatically (from 900 of them to only 25 or by a 36-fold decrease) with the MLE-trapping scheme. In summary, the MLE-trapping scheme based on probabilistic profiles of the double runs statistic (N_s, R) enhances the EM algorithm by providing fewer and better initial estimates which can jump-start the EM algorithm in the neighbourhood of $\widehat{\boldsymbol{\theta}}$. As a result, two difficult issues associated with the EM algorithm (i.e. having a slow rate of convergence and landing on a local maximum or saddle point of the likelihood surface) in the HMM parameter estimation procedure are tackled simultaneously.

4.3.3 Constructing Confidence Intervals by Parametric Bootstrapping

With assumptions to avoid identifiability problems, there are some vigorous proofs on asymptotic properties of the maximum likelihood estimators $\widehat{\boldsymbol{\theta}}$ for general hidden Markov models in recent literature. In 1992, Leroux proved that the consistency of the maximum likelihood estimators for general hidden Markov models (Leroux, 1992 [89]). Then later, Bickel and his colleagues proved that the maximum likelihood estimators for general hidden Markov models are asymptotically normal under mild regularity conditions (Bickel et al., 1996, 1998 [20, 21]). In addition to the theoretical insights*

*NOTE: The mathematical and statistical proofs on the consistency and asymptotic normality of the maximum likelihood estimators for HMMs certainly shed some new light on the development of more sophisticated bootstrap-based methods to construct better confidence intervals/regions for the HMM parameter $\boldsymbol{\theta}$. This is an interesting branch of research to be explored in the future.

gained from these mathematical and statistical proofs, we employ a computer-based resampling approach — the parametric bootstrap — to construct simple confidence intervals for the individual components of the HMM parameter set $\boldsymbol{\theta}$. Since the maximum likelihood approach is the approach that we use for the HMM parameter estimation, our first task is to get bootstrap replications of the MLE $\widehat{\boldsymbol{\theta}}$.

Specifically, we first incorporate the MLE-trapping scheme developed in subsection §4.3.2 into the parametric bootstrap paradigm and build an "HMM-MLE bootstrap engine" to generate bootstrap replications of $\widehat{\boldsymbol{\theta}}$. In our notation, $y_{[1,L]}^{(1)\diamond}, y_{[1,L]}^{(2)\diamond}, \cdots, y_{[1,L]}^{(b_s)\diamond}, \cdots, y_{[1,L]}^{(B_s)\diamond}$ denote the B_s independent bootstrap samples (or realizations) simulated from a fitted HMM. The fitted HMM is the HMM that fitted to the actual sequence data $y_{[1,L]}$, hence, we resulted with the MLE $\widehat{\boldsymbol{\theta}}$ corresponding to the data. Having developed an HMM simulator, bootstrap samples are obtained from setting the parameters of the HMM simulator equal to the $\widehat{\boldsymbol{\theta}}$. Then, corresponding to each bootstrap sample, we fit an HMM of the same type with the same identifiability constraints and get the corresponding MLE through the MLE-trapping scheme and the EM algorithm. As a result, we have B_s bootstrap replications of $\widehat{\boldsymbol{\theta}}$, and they are denoted as $\widehat{\boldsymbol{\theta}}_1^{\diamond}, \widehat{\boldsymbol{\theta}}_2^{\diamond}, \cdots, \widehat{\boldsymbol{\theta}}_{b_s}^{\diamond}, \cdots, \widehat{\boldsymbol{\theta}}_{B_s}^{\diamond}$ (Refer to Figure 4.4 for a schematic illustration).

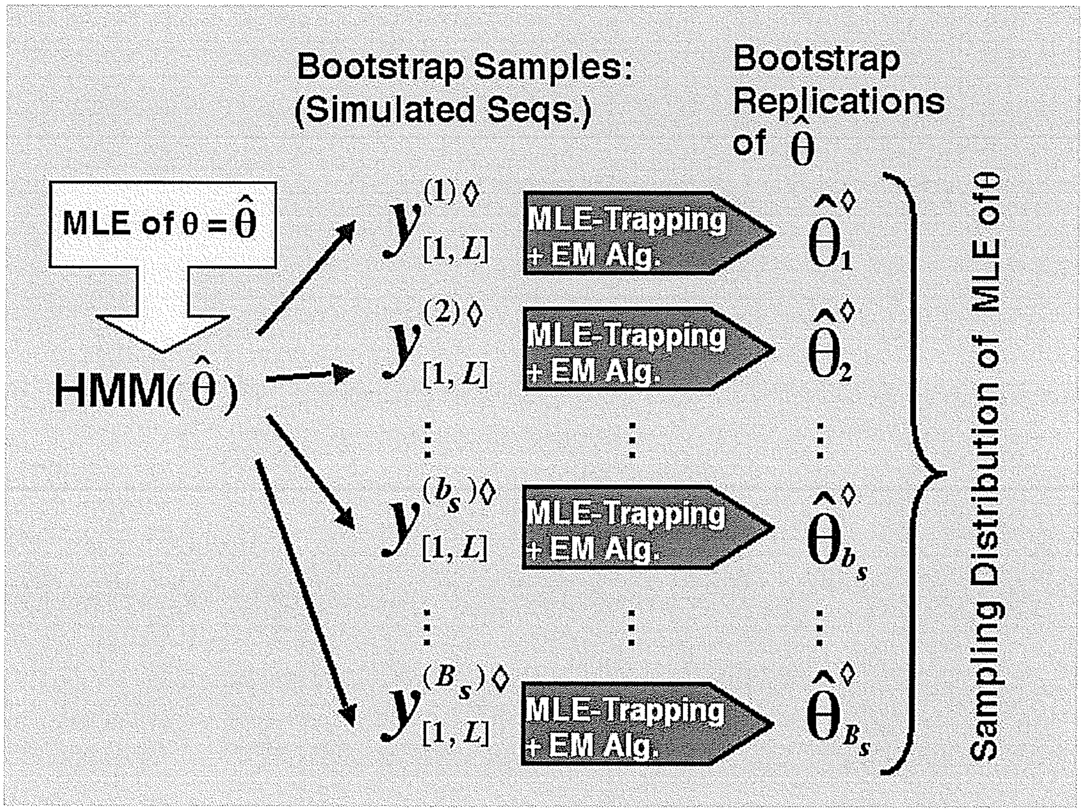

Bootstrap Samples: (Simulated Seqs.)
Bootstrap Replications of θ̂
MLE of θ = θ̂
HMM(θ̂)
$y^{(1)\diamond}_{[1,L]}$
$y^{(2)\diamond}_{[1,L]}$
$y^{(b_s)\diamond}_{[1,L]}$
$y^{(B_s)\diamond}_{[1,L]}$
MLE-Trapping + EM Alg.
MLE-Trapping + EM Alg.
MLE-Trapping + EM Alg.
MLE-Trapping + EM Alg.
$\hat{\theta}^{\diamond}_1$
$\hat{\theta}^{\diamond}_2$
$\hat{\theta}^{\diamond}_{b_s}$
$\hat{\theta}^{\diamond}_{B_s}$
Sampling Distribution of MLE of θ

Figure 4.4: Illustration of an HMM-MLE Bootstrap Engine (NOTE: Each simulated sequence is of the same length L)

Two simple bootstrap-based methods (generally described in Efron & Tibshirani, 1993 [50]) are used to construct confidence intervals for the individual components of the HMM parameter set $\boldsymbol{\theta}$. For a fitted binary HMM, an individual component of the MLE $\widehat{\boldsymbol{\theta}}$ is also denoted as $\widehat{\theta}$, i.e. $\widehat{\theta}$ can be $\widehat{a_{00}}, \widehat{a_{11}}, \widehat{b_{0(0)}}$, or $\widehat{b_{1(1)}}$. Similarly, an individual component of a bootstrap replication of $\widehat{\boldsymbol{\theta}}$ is denoted as $\widehat{\theta}^{\diamond}$, i.e. $\widehat{\theta}^{\diamond}$ can be $\widehat{a_{00}}^{\diamond}, \widehat{a_{11}}^{\diamond}, \widehat{b_{0(0)}}^{\diamond}$, or $\widehat{b_{1(1)}}^{\diamond}$.

1. **Percentile Method:**

   The $(1-2\alpha)$ percentile interval for an individual θ is defined by

$$[\widehat{\theta}^{\diamond}_{(\alpha)}, \quad \widehat{\theta}^{\diamond}_{(1-\alpha)}], \tag{4.21}$$

   where $\widehat{\theta}^{\diamond}_{(\alpha)}$ and $\widehat{\theta}^{\diamond}_{(1-\alpha)}$ are the $B_s \times \alpha$-th and the $B_s \times (1-\alpha)$-th values in the ordered list of the B_s $\widehat{\theta}^{\diamond}$'s.

2. **Student's t Method:**

The $(1-2\alpha)$ Student's t interval for an individual θ is defined by

$$[\widehat{\theta} - t_{L-1}^{(1-\alpha)} \times sd(\widehat{\theta}^{\diamond}), \quad \widehat{\theta} - t_{L-1}^{(\alpha)} \times sd(\widehat{\theta}^{\diamond})], \tag{4.22}$$

where $t_{L-1}^{(1-\alpha)} = -t_{L-1}^{(\alpha)}$, e.g. $t_{51-1}^{0.95} = -t_{51-1}^{0.05} = 1.68$ when $L = 51$ and $\alpha = 0.05$, and $sd(\widehat{\theta}^{\diamond})$ is the standard deviation of the B_s $\widehat{\theta}^{\diamond}$'s. The lower bound of a Student's t interval is set to the minimum possible value of θ, say θ_{min}, if we have $\widehat{\theta} - t_{L-1}^{(1-\alpha)} \times sd(\widehat{\theta}^{\diamond}) < \theta_{min}$. Similarly, the upper bound of a Student's t interval is set to one if we have $\widehat{\theta} - t_{L-1}^{(\alpha)} \times sd(\widehat{\theta}^{\diamond}) > 1$.

NOTE: The variance-covariance matrix of $\boldsymbol{\theta}$ is denoted as:

$$\mathcal{V}(\widehat{\boldsymbol{\theta}}) = \begin{bmatrix} Var(\widehat{a_{00}}, \widehat{a_{00}}) & Cov(\widehat{a_{00}}, \widehat{a_{11}}) & Cov(\widehat{a_{00}}, \widehat{b_{0(0)}}) & Cov(\widehat{a_{00}}, \widehat{b_{1(1)}}) \\ Cov(\widehat{a_{11}}, \widehat{a_{00}}) & Var(\widehat{a_{11}}, \widehat{a_{11}}) & Cov(\widehat{a_{11}}, \widehat{b_{0(0)}}) & Cov(\widehat{a_{11}}, \widehat{b_{1(1)}}) \\ Cov(\widehat{b_{0(0)}}, \widehat{a_{00}}) & Cov(\widehat{b_{0(0)}}, \widehat{a_{11}}) & Var(\widehat{b_{0(0)}}, \widehat{b_{0(0)}}) & Cov(\widehat{b_{0(0)}}, \widehat{b_{1(1)}}) \\ Cov(\widehat{b_{1(1)}}, \widehat{a_{00}}) & Cov(\widehat{b_{1(1)}}, \widehat{a_{11}}) & Cov(\widehat{b_{1(1)}}, \widehat{b_{0(0)}}) & Var(\widehat{b_{1(1)}}, \widehat{b_{1(1)}}) \end{bmatrix},$$

and the bootstrap estimate for the $\mathcal{V}(\widehat{\boldsymbol{\theta}})$ is calculated by:

$$\mathcal{V}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \frac{1}{B_s - 1} \sum_{b_s=1}^{B_s} \left(\widehat{\boldsymbol{\theta}}_{b_s}^{\diamond} - \overline{\widehat{\boldsymbol{\theta}}^{\diamond}} \right)' \left(\widehat{\boldsymbol{\theta}}_{b_s}^{\diamond} - \overline{\widehat{\boldsymbol{\theta}}^{\diamond}} \right), \tag{4.23}$$

where

$$\overline{\widehat{\boldsymbol{\theta}}^{\diamond}} = \frac{1}{B_s} \sum_{b_s=1}^{B_s} \widehat{\boldsymbol{\theta}}_{b_s}^{\diamond}. \tag{4.24}$$

With the MLE-trapping scheme based on the previously mentioned 2 × 25 × 36 = 1800 probabilistic profiles of the double runs statistic (N_s, R), we use the HMM-MLE bootstrap engine to generate bootstrap replications of $\widehat{\boldsymbol{\theta}}$ of a constrained binary HMM. As in §4.3.2, the identifiability constraints are $a_{00} > 0.5$, $a_{11} > 0.5$ and $b_{0(0)} > b_{1(0)}$. For an illustrative study, we suppose that a constrained binary HMM is fitted to an actual binary sequence, and the MLE $\widehat{\boldsymbol{\theta}} = (\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8, \widehat{b_{1(1)}} = 0.4)$. Based on 100 bootstrap replications, we study the effects of having a different sequence length L (e.g. $L = 51, 101, 501, 1001$, or 5001) and compare the corresponding 90% percentile interval and the corresponding 90% Student's t interval for each individual component of $\boldsymbol{\theta}$. As L increases from 51 to 5001, we observe the following five rough trends[†]:

- Each entry of $\mathcal{V}_L(\widehat{\boldsymbol{\theta}}^{\diamond})$ tends to decrease in absolute value. (Refer to $\mathcal{V}_{L=51}(\widehat{\boldsymbol{\theta}}^{\diamond})$, $\mathcal{V}_{L=101}(\widehat{\boldsymbol{\theta}}^{\diamond})$, $\mathcal{V}_{L=501}(\widehat{\boldsymbol{\theta}}^{\diamond})$, $\mathcal{V}_{L=1001}(\widehat{\boldsymbol{\theta}}^{\diamond})$, and $\mathcal{V}_{L=5001}(\widehat{\boldsymbol{\theta}}^{\diamond})$)

- Since the parametric bootstrap standard deviation estimate $sd(\widehat{\theta}^{\diamond})$ is the square root of the corresponding diagonal element of $\mathcal{V}_L(\widehat{\boldsymbol{\theta}}^{\diamond})$, it is generally getting smaller and smaller (Refer to Table 4.4).

- Both the median $\widehat{\theta}^{\diamond}_{(50\%)}$ and the mean $\overline{\widehat{\theta}^{\diamond}}$ are approaching to the MLE $\widehat{\theta}$ (Refer to Table 4.4).

- The bootstrap sampling distribution of $\widehat{\theta}$ is becoming more and more symmetric and looking more normal-shaped (Refer to Table 4.4 and Figures 4.5, 4.6, 4.7, & 4.8).

- The discrepancy between the percentile interval and the Student's t interval is diminishing (Refer to Table 4.5 and Figures 4.5, 4.6, 4.7, & 4.8).

The above findings are not at all surprising, especially with the theoretical proofs on the consistency and asymptotic normality of the MLE $\widehat{\boldsymbol{\theta}}$. Since the coverage probability of the percentile method or the Student's t method will usually not exactly equal the desired $100(1-2\alpha)\%$, either the percentile interval or the Student's t interval is only an approximate confidence interval for θ. As recommended by Efron and Tibshirani, in general, one should prefer the percentile interval over the Student's t interval if the bootstrap sampling distribution is quite asymmetric (Efron & Tibshirani, 1993 [50]). The precision of the coverage probability of these simple methods and the development

[†] NOTE: Based on only 100 bootstrap replications, the five trends may be affected by the presence of outliers.

of more sophisticated bootstrap-based methods to construct confidence intervals/regions with more precise coverage probability for the HMM parameter $\boldsymbol{\theta}$ are to be explored in the future.

$$\mathcal{V}_{L=51}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 0.0142 & -0.0040 & -0.0067 & 0.0013 \\ -0.0040 & 0.0266 & 0.0095 & -0.0011 \\ -0.0067 & 0.0095 & 0.0105 & 5.385\times10^{-5} \\ 0.0013 & -0.0011 & 5.385\times10^{-5} & 0.03671 \end{bmatrix},$$

$$\mathcal{V}_{L=101}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 0.0062 & -0.0012 & -0.0040 & 0.0040 \\ -0.0012 & 0.0175 & 0.0067 & -0.0048 \\ -0.0040 & 0.0067 & 0.0097 & 0.0016 \\ 0.0040 & -0.0048 & 0.0016 & 0.0426 \end{bmatrix},$$

$$\mathcal{V}_{L=501}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 0.0011 & 0.0008 & -0.0004 & 0.0021 \\ 0.0008 & 0.0084 & 0.0016 & -0.0010 \\ -0.0004 & 0.0016 & 0.0017 & 0.0007 \\ 0.0021 & -0.0010 & 0.0007 & 0.0211 \end{bmatrix},$$

$$\mathcal{V}_{L=1001}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 0.0006 & 0.0010 & -0.0001 & 0.0002 \\ 0.0010 & 0.0060 & 0.0008 & -0.0019 \\ -0.0001 & 0.0008 & 0.0010 & 0.0013 \\ 0.0002 & -0.0019 & 0.0013 & 0.0116 \end{bmatrix}, \quad \text{and}$$

$$\mathcal{V}_{L=5001}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 1.571\times10^{-5} & 2.492\times10^{-5} & 6.033\times10^{-6} & -5.008\times10^{-6} \\ 2.492\times10^{-5} & 4.862\times10^{-4} & 1.492\times10^{-4} & 2.964\times10^{-4} \\ 6.033\times10^{-6} & 1.492\times10^{-4} & 1.611\times10^{-4} & 3.116\times10^{-4} \\ -5.008\times10^{-6} & 2.964\times10^{-4} & 3.116\times10^{-4} & 9.805\times10^{-4} \end{bmatrix}.$$

(NOTE: Many covariance entries of $\mathcal{V}_{L=51}(\widehat{\boldsymbol{\theta}}^{\diamond})$ do not follow the first trend.)

Table 4.4: Percentiles, Means and Standard Deviations of $\widehat{\theta}^{\diamond}$ Based on 100 Bootstrap Replications with $L = 51, 101, 501, 1001, 5001$ (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8$, and $\widehat{b_{1(1)}} = 0.4$ are fed into the HMM-MLE bootstrap engine)

HMM		Percentile							Mean	Std. Dev.
$\widehat{\theta}^{\diamond}$	L	5%	10%	25%	50%	75%	90%	95%	$\overline{\widehat{\theta}^{\diamond}}$	$sd(\widehat{\theta}^{\diamond})$
$\widehat{a_{00}}^{\diamond}$	51	0.561	0.659	0.799	0.893	0.915	0.948	0.961	0.843	0.1192
	101	0.715	0.785	0.871	0.894	0.911	0.968	0.982	0.878	0.0789
	501	0.869	0.880	0.896	0.901	0.908	0.945	0.976	0.905	0.0334
	1001	0.876	0.886	0.898	0.901	0.906	0.920	0.932	0.902	0.0253
	5001	0.897	0.898	0.899	0.900	0.901	0.903	0.904	0.901	0.0040
$\widehat{a_{11}}^{\diamond}$	51	0.586	0.612	0.643	0.738	0.994	0.999	1.000	0.796	0.1632
	101	0.557	0.626	0.664	0.704	0.857	0.955	0.999	0.753	0.1322
	501	0.614	0.645	0.665	0.697	0.745	0.885	0.922	0.720	0.0916
	1001	0.591	0.656	0.671	0.695	0.711	0.772	0.883	0.702	0.0774
	5001	0.679	0.683	0.695	0.700	0.708	0.718	0.721	0.703	0.0221
$\widehat{b_{0(0)}}^{\diamond}$	51	0.728	0.748	0.802	0.903	0.999	1.000	1.000	0.892	0.1025
	101	0.714	0.726	0.764	0.820	0.937	0.997	0.999	0.846	0.0984
	501	0.753	0.758	0.777	0.797	0.823	0.847	0.868	0.804	0.0414
	1001	0.756	0.764	0.781	0.798	0.820	0.837	0.845	0.802	0.0314
	5001	0.780	0.783	0.791	0.800	0.808	0.814	0.820	0.800	0.0127
$\widehat{b_{1(1)}}^{\diamond}$	51	0.157	0.190	0.247	0.318	0.426	0.642	0.772	0.367	0.1916
	101	0.221	0.250	0.297	0.359	0.471	0.740	0.994	0.429	0.2065
	501	0.245	0.264	0.316	0.384	0.447	0.547	0.721	0.407	0.1452
	1001	0.264	0.274	0.326	0.387	0.432	0.474	0.559	0.394	0.1078
	5001	0.339	0.351	0.377	0.397	0.415	0.433	0.437	0.394	0.0313

Table 4.5: The 90% Percentile Intervals and the Student's t Intervals for θ Based on 100 Bootstrap Replications of with $L = 51, 101, 501, 1001, 5001$ (NOTE: $t_{50}^{0.95} = 1.68$, $t_{100}^{0.95} = 1.66$, $t_{L \geq 500}^{0.95} = 1.645$. $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8$, and $\widehat{b_{1(1)}} = 0.4$ are fed into the HMM-MLE bootstrap engine)

HMM		90% Percentile Interval		90% Student's t Interval	
θ	L	Lower Bound	Upper Bound	Lower Bound	Upper Bound
a_{00}	51	0.561	0.961	0.699	1.000
	101	0.715	0.982	0.769	1.000
	501	0.869	0.976	0.845	0.955
	1001	0.876	0.932	0.858	0.942
	5001	0.897	0.904	0.893	0.907
a_{11}	51	0.586	1.000	0.426	0.974
	101	0.557	0.999	0.481	0.919
	501	0.614	0.922	0.549	0.851
	1001	0.591	0.883	0.573	0.827
	5001	0.679	0.721	0.664	0.736
$b_{0(0)}$	51	0.728	1.000	0.628	0.972
	101	0.714	0.999	0.637	0.963
	501	0.753	0.868	0.732	0.868
	1001	0.756	0.845	0.748	0.852
	5001	0.780	0.820	0.779	0.821
$b_{1(1)}$	51	0.157	0.772	0.078	0.722
	101	0.221	0.994	0.057	0.743
	501	0.245	0.721	0.161	0.639
	1001	0.264	0.559	0.223	0.577
	5001	0.339	0.437	0.348	0.452

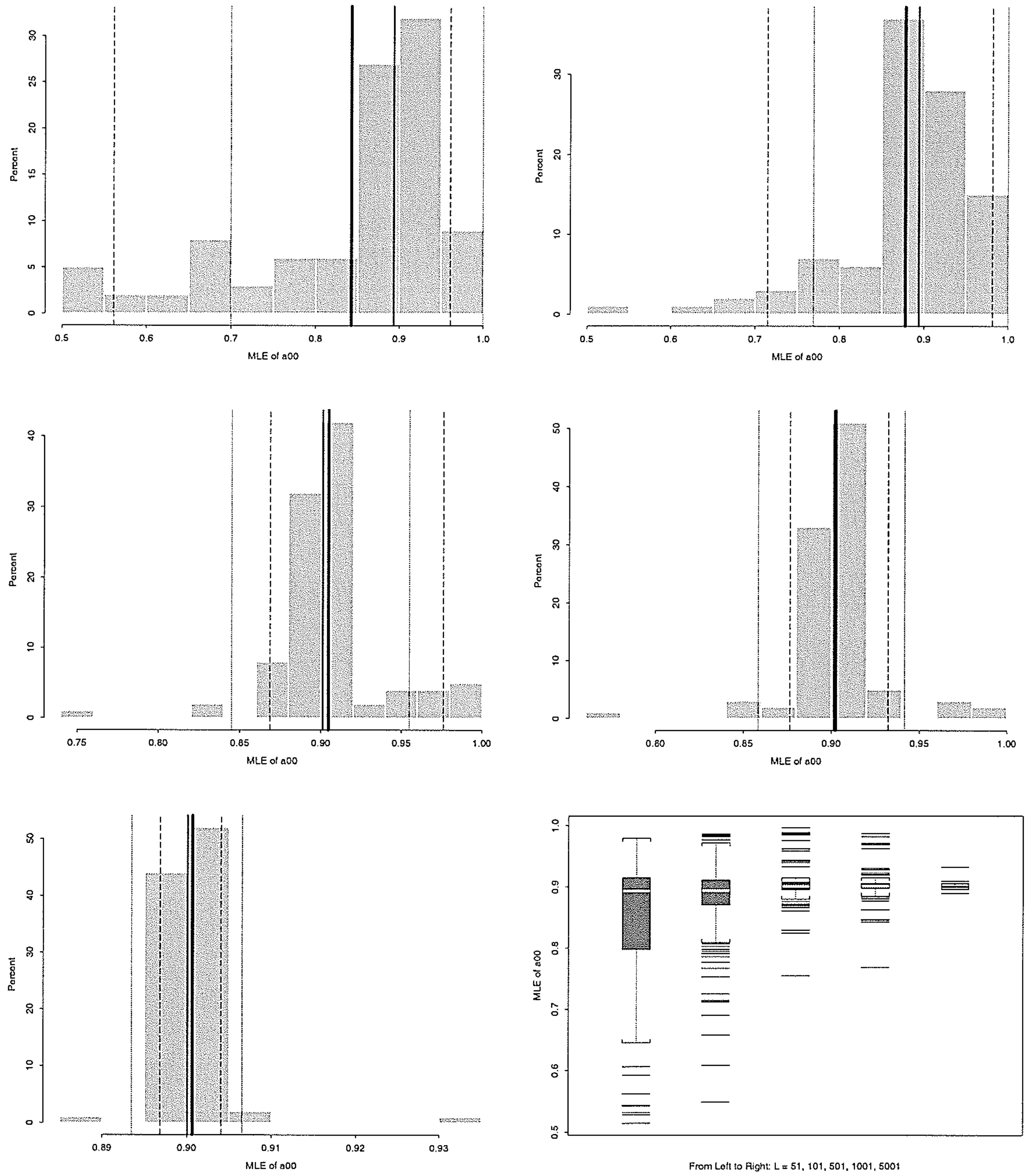


Figure 4.5: Bootstrap Sampling Distributions of $\widehat{a_{00}}$ of a Constrained Binary HMM. **Top Panel** — $L = 51$ (Left), $L = 101$ (Right). **Middle Panel** — $L = 501$ (Left), $L = 1001$ (Right). **Bottom Panel** — $L = 5001$ (Left), Side-by-side Box-plot (Right). (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8$, and $\widehat{b_{1(1)}} = 0.4$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{a_{00}}^{\diamond}}$ (thick) & $\widehat{a_{00}}^{\diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{a_{00}}^{\diamond}_{(5\%)}$ & $\widehat{a_{00}}^{\diamond}_{(95\%)}$)

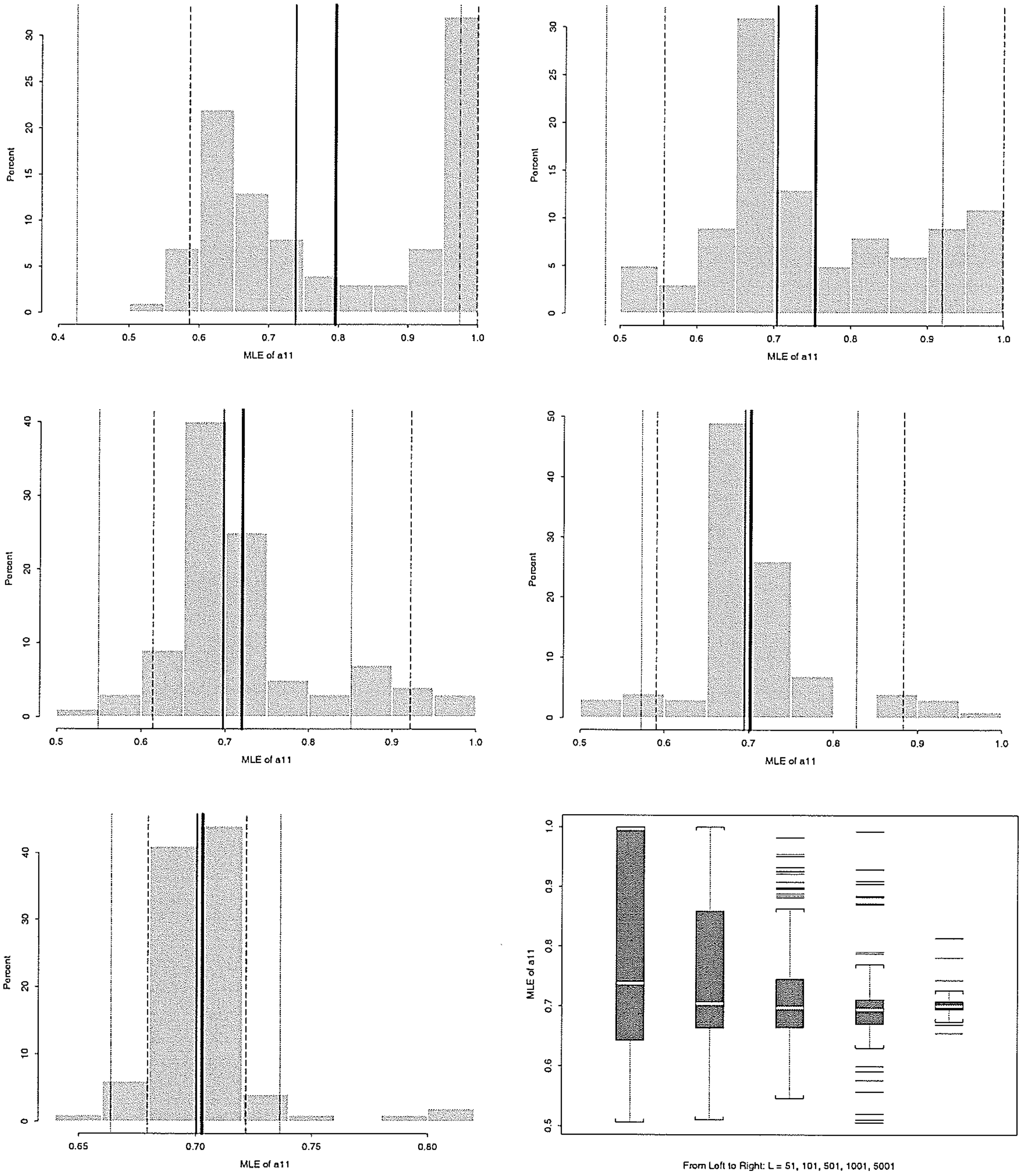


Figure 4.6: Bootstrap Sampling Distributions of $\widehat{a_{11}}$ of a Constrained Binary HMM. **Top Panel** — $L = 51$ (Left), $L = 101$ (Right). **Middle Panel** — $L = 501$ (Left), $L = 1001$ (Right). **Bottom Panel** — $L = 5001$ (Left), Side-by-side Box-plot (Right). (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8$, and $\widehat{b_{1(1)}} = 0.4$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{a_{11}}^{\Diamond}}$ (thick) & $\widehat{a_{11}}^{\Diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{a_{11}}^{\Diamond}_{(5\%)}$ & $\widehat{a_{11}}^{\Diamond}_{(95\%)}$)

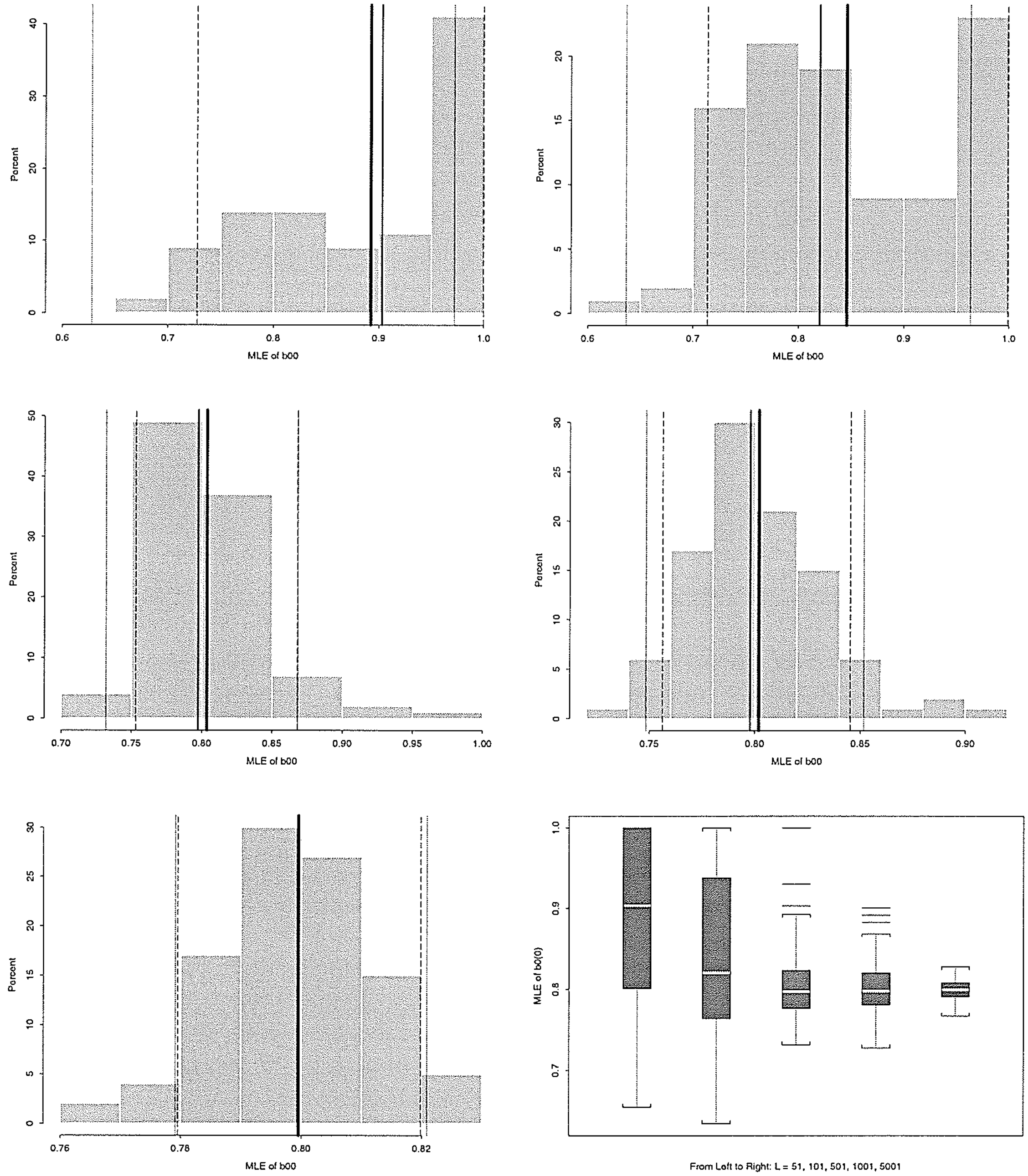


Figure 4.7: Bootstrap Sampling Distributions of $\widehat{b_{0(0)}}$ of a Constrained Binary HMM. **Top Panel** — $L = 51$ (Left), $L = 101$ (Right). **Middle Panel** — $L = 501$ (Left), $L = 1001$ (Right). **Bottom Panel** — $L = 5001$ (Left), Side-by-side Box-plot (Right). (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8$, and $\widehat{b_{1(1)}} = 0.4$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{b_{0(0)}}^{\Diamond}}$ (thick) & $\widehat{b_{0(0)}}^{\Diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{b_{0(0)}}^{\Diamond}_{(5\%)}$ & $\widehat{b_{0(0)}}^{\Diamond}_{(95\%)}$)

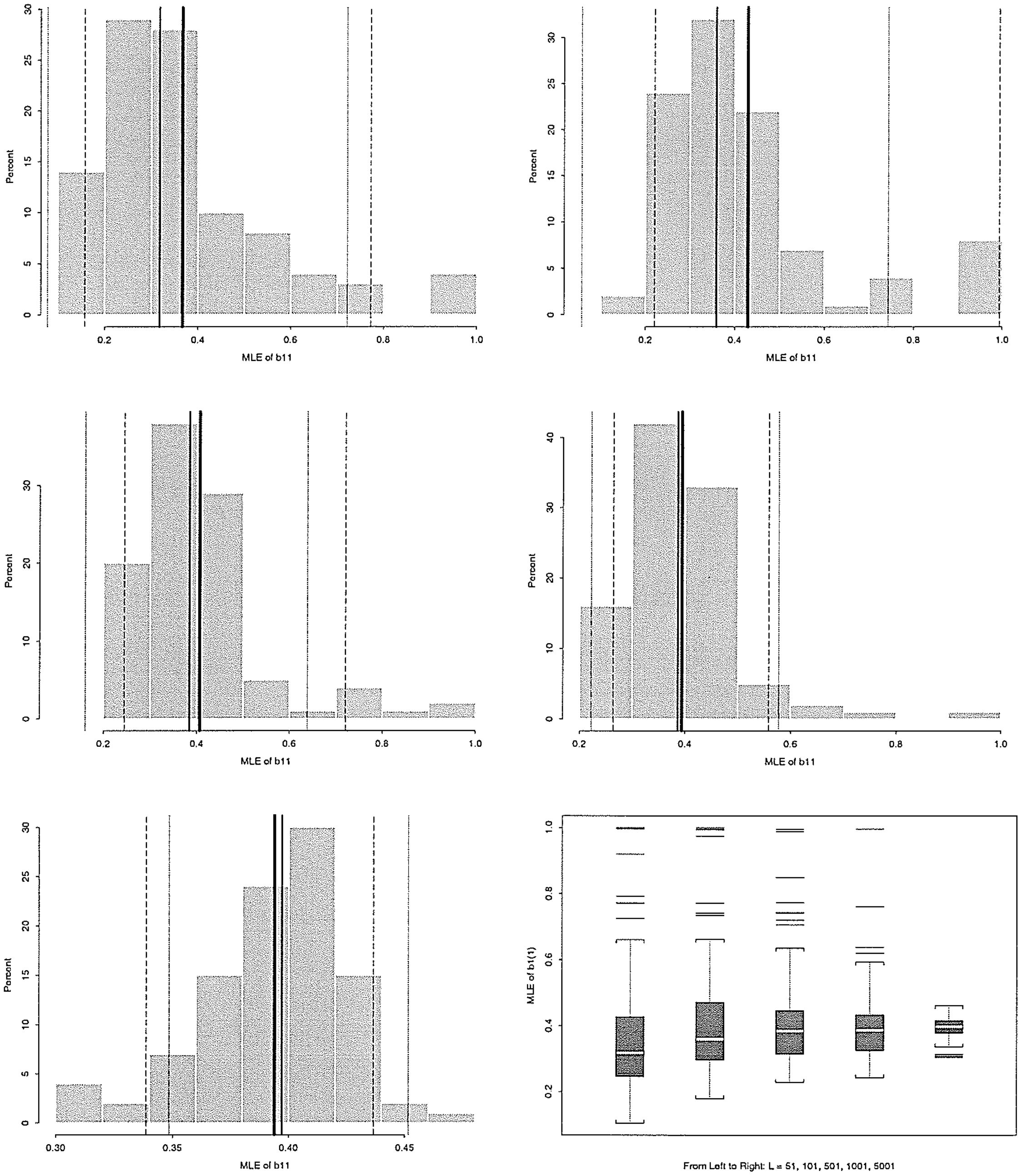


Figure 4.8: Bootstrap Sampling Distributions of $\widehat{b_{1(1)}}$ of a Constrained Binary HMM. **Top Panel** — $L = 51$ (Left), $L = 101$ (Right). **Middle Panel** — $L = 501$ (Left), $L = 1001$ (Right). **Bottom Panel** — $L = 5001$ (Left), Side-by-side Box-plot (Right). (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.9, \widehat{a_{11}} = 0.7, \widehat{b_{0(0)}} = 0.8$, and $\widehat{b_{1(1)}} = 0.4$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{b_{1(1)}}}^{\diamond}$ (thick) & $\widehat{b_{1(1)}}^{\diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{b_{1(1)}}^{\diamond}_{(5\%)}$ & $\widehat{b_{1(1)}}^{\diamond}_{(95\%)}$)

4.4 Applications: Analysis of Human Genomic DNA Sequences

In this section, we demonstrate the use of the conditional runs statistic $(R|N_s)$, the double runs statistic (N_s, R), and the probabilistic profiles in conjunction with a binary HMM for DNA pattern recognition. The objectives are to detect non-random clustering of selected "signals" and to help reveal the start sites of transcription of RNA polymerase II transcribed genes. Based on general biological findings, we define appropriate signals with respect to DNA bendability in our studies. We treat the DNA sequence data as experimentally uncharacterized DNA sequence data during our pattern recognition analysis.

4.4.1 Descriptions of DNA Sequence Data Sets

After very thorough reduction of data redundancy*, data of 624 human genomic DNA sequences were provided by Dr. Anders Pedersen at the Center for Biological Sequence Analysis of the Technical University of Denmark. These data were originally extracted from the GenBank database release 95. Sequences contain ambiguous nucleotide symbols and/or non-experimentally determined transcriptional start sites were discarded during their data revision process. Each sequence is of length 501 nucleotides (nt), and has 250 nt upstream of an experimentally determined start site of transcription of a RNA polymerase II transcribed gene. The data were subdivided into two data sets: "DcSet" and "DncSet". DcSet contains 262 sequences with coding subsequences and DncSet contains 362 sequences without any coding subsequences† in the 250 nt downstream region of the start site of transcription (Pedersen et al., 1998 [109]). The GenBank LOCUS names and accession numbers of the sequences in the first data set DcSet (Refer to Appendix A) and the second data set DncSet (Refer to Appendix B) are appended at the end of the thesis.

*NOTE: Sequence data in databases are known to be redundant due to the presence of identical sequences being submitted to the database more than once or genes belonging to gene families (Pedersen et al., 1998 [109]).

†NOTE: Each sequence in DncSet is indeed having a 5′ untranslated region of 250 nt or longer downstream.

4.4.2 Analysis of DNA Bendability Signals

Since Brukner and his colleagues derived the sequence-dependent DNA bendability scales (or bending propensity) of trinucleotides directly from DNase I digestion data (Brukner et al., 1995 [25]), a number of studies has been conducted to analyze bendability scales along a DNA sequence (e.g. Baldi et al., 1998, 1999 [8, 6]; Pedersen et al., 1998 [109]). In particular, studies on DNase I-derived bendability scales by Pedersen and his colleagues indicated that the upstream region of the start site of transcription is generally less bendable or flexible than the downstream region (Pedersen et al., 1998 [109]). Their analysis of the two data sets showed that the downstream region being generally more bendable or flexible is not due to codon usage bias. They found that the bendability difference between upstream and downstream regions could be explained by the preferential usage of a number of certain low-bendability trinucleotides upstream and certain high-bendability trinucleotides downstream. Specifically, they found that 10 out of the 13 most over-represented trinucleotides in the downstream region have high DNase I-derived bendability scales, and the 6 most over-represented trinucleotides in the upstream region all have low DNase I-derived bendability scales (Pedersen et al., 1998 [109]).

Using the DNase I-derived trinucleotide bendability scales (as in Pedersen et al., 1998 [109]), we view a DNA sequence as a sequence of overlapping trinucleotides, and we first obtain a sequence of corresponding bendability scales for each DNA sequence in both data sets. Simple plots of the average bendability scales are shown in Figure 4.9 (Top panel). In terms of the average bendability scales, the upstream region indeed appears to be less bendable or flexible than the downstream region. However, when we look at the bendability scales along an individual sequence, it becomes rather difficult to see the bendability difference between upstream and downstream regions even a significant difference exists. For example, plots of individual bendability scales of a sequence from GenBank locus HUMMETIPG in DcSet and a sequence from GenBank locus HSDYSE51 in DncSet are shown in Figure 4.9 (Bottom panel).

Based on the premise that the bendability difference between upstream and downstream regions could be explained by the preferential usage of a number of certain low-bendability trinucleotides

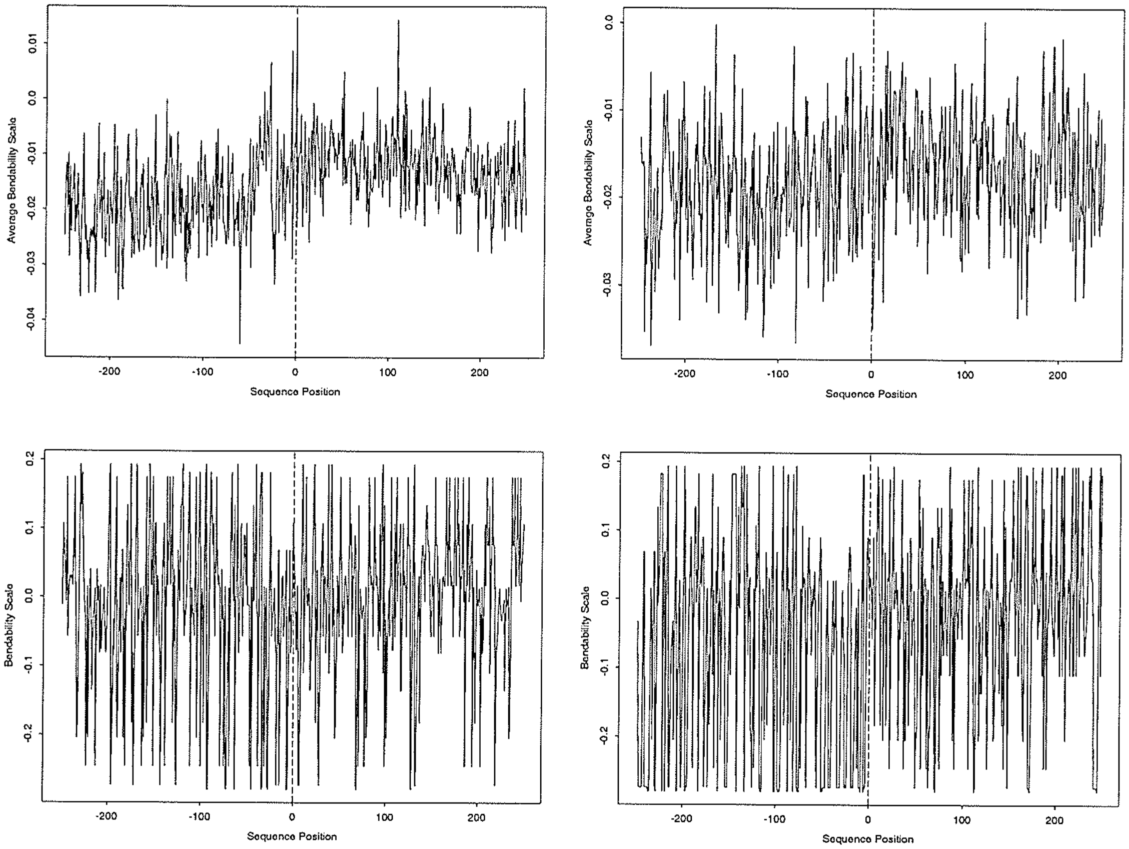


Figure 4.9: **Top Panel** — Average bendability scales of the DcSet (Left) & the DncSet (Right). **Bottom Panel** — Bendability scales of a sequence from GenBank locus HUMMETIPG (Left) & a sequence from GenBank locus HSDYSE51 (Right). (NOTE: Dashed line indicates the true start site of transcription)

upstream and certain high-bendability trinucleotides downstream, we analyze runs of "bendability signals" to detect clustering patterns. According to the findings reported by Pedersen and his colleagues (Pedersen et al.,1998 [109]), we have decided to dichotomize bendability scales in two ways for two separate studies: "H10" and "L6" studies. In the H10 study, we dichotomize bendability scales by treating the 10 trinucleotides TCT, GCA, GCC, CAG, CTG, GAG, CTC, AGC, GCT, and TGC, mentioned in Pedersen et al., 1998 [109], as selected high "H10 bendability signals". In other words, a DNA sequence is represented as a sequence of binary (high "1" or low "0") H10 bendability signals. Similarly, we dichotomize bendability scales by treating the 6 trinucleotides

GGG, CCC, AAA, TTT, AAT, and ATT, mentioned in Pedersen et al., 1998 [109], as selected low "L6 bendability signals" in the L6 study. Having observed runs statistics N_s and R in terms of the H10 (or L6) bendability signals in each DNA sequence, we have found almost all DNA sequences* fall into the lower critical/rejection regions of the conditional runs tests for randomness at the individual significant level of 5% (Refer to Figure 4.10).

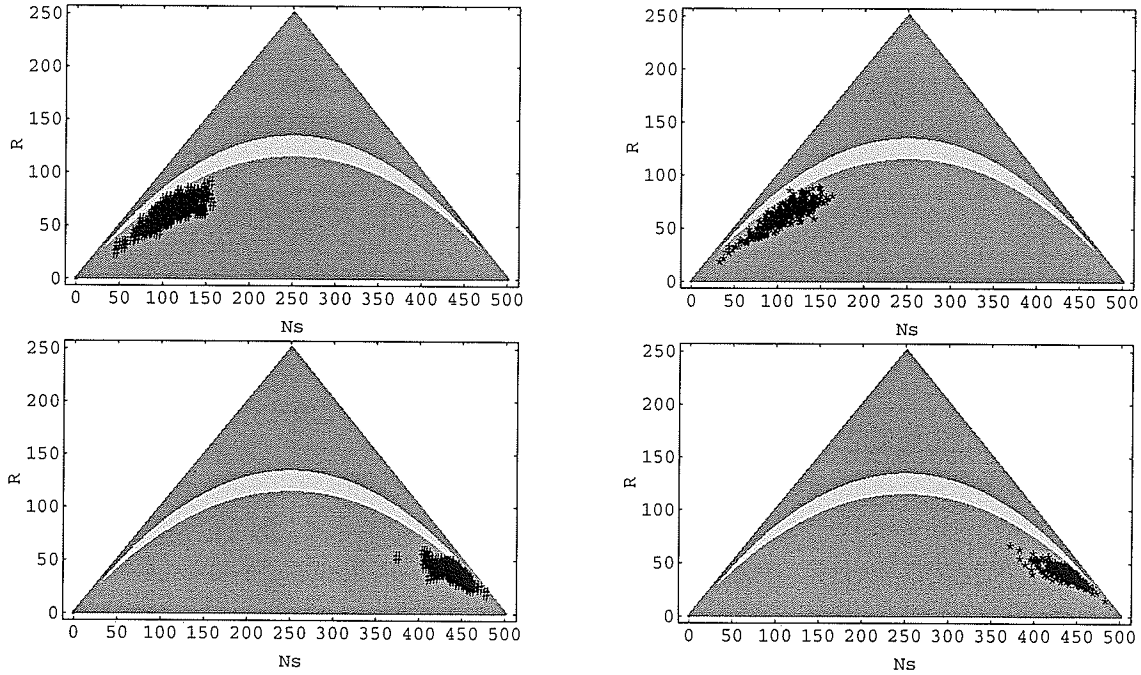


Figure 4.10: Conditional Runs Test. Runs tests based on R given $N_s = n$ for testing randomness of selected high bendability signals of individual DNA sequences in DcSet (Left) & DncSet (Right). **Top Panel** — Use of binary H10 signal representation. **Bottom Panel** — Use of binary L6 signal representation. (NOTE: Significance level = 5 %, dark regions are critical regions, and an observed (N_s, R) is denoted as a # for a sequence in DcSet & as a * for a sequence in DncSet)

*NOTE: Two sequences from the same GenBank locus HSU52111, the *Homo sapiens* X28 region which nears the adrenoleukodystrophy (ALD) protein locus, appear to have random H10 bendability signals.

Hence, we reject the null hypothesis of randomness and conclude non-random clustering of H10 (or L6) bendability signals in most sequences in the data sets. In other words, the individual H10 (or L6) signals in a DNA sequence tend to stick together to form non-random blocks/clusters of signals. These blocks/clusters of signals are reflected by the conditional runs statistic $(R|N_s)$. With the premise of the preferential usage of certain bendability signals in the upstream and downstream regions, the conclusion of non-random clustering of bendability signals from the runs tests supports the notion of viewing a DNA sequence as having distinct regions which are homogeneous in bendability signal composition. This is our justification to employ an HMM to capture this mosaic structure for the prediction of the start site of transcription of a RNA polymerase II transcribed gene. Specifically, a DNA sequence in terms of binary H10 (or L6) bendability signals is modeled by a binary HMM as follows. (Refer to Figure 4.11)

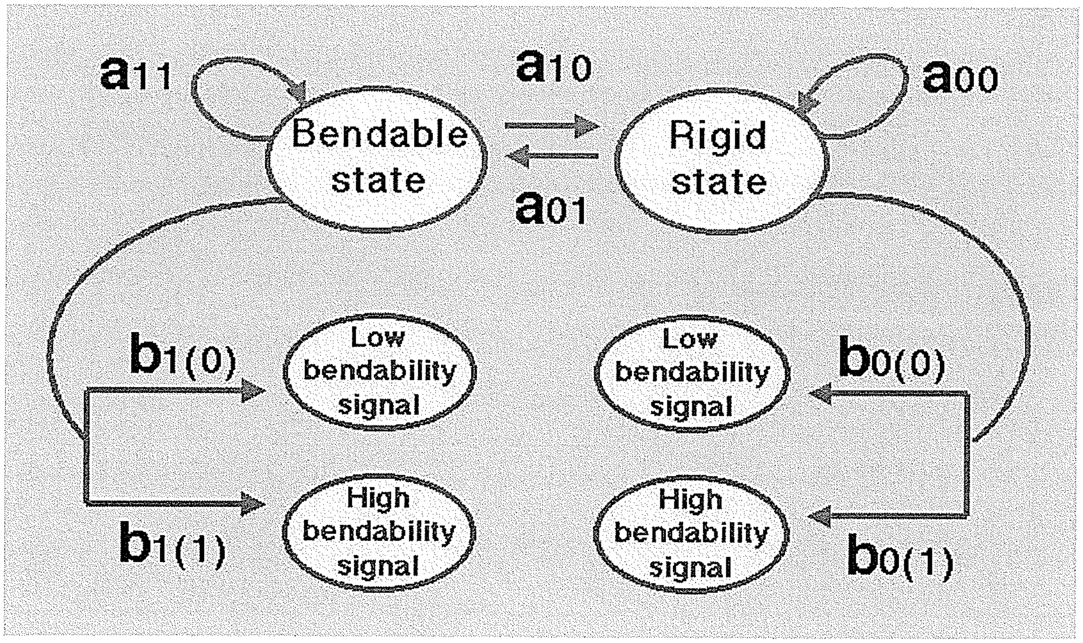


Figure 4.11: A Binary HMM for the Analysis of DNA Bendability Pattern. (NOTE: Bendability scales are dichotomized as binary H10 (or L6) bendability signals for the H10 (or L6) study)

The binary HMM has two hidden states, i.e. "rigid-state" (or state 0) and "bendable-state" (or state 1), and two observable outcomes, i.e the binary H10 (or L6) bendability signals. As discussed in §4.3.1 (p.79–), sensible constraints must be placed on the parameter space to avoid HMM identifiability problems in practice. Having been inspired by Churchill's work (especially Churchill, 1989 & 1992 [31, 32]), we decide to follow his logic to avoid HMM identifiability problems. First, we explicitly ensure the hidden binary state process to be strictly Markovian by imposing constraints $a_{00} > 0.5$ and $a_{11} > 0.5$ on the transition probability matrix $\mathcal{A}_X$. These constraints are stronger than the general constraint $a_{01} + a_{10} \neq 1$ (or equivalently $a_{00} + a_{11} \neq 1$), but weaker than the ones such as $a_{00} > 0.999$ and $a_{11} > 0.999$ used by Churchill (Churchill, 1989 & 1992 [31, 32]). With constraints $a_{00} > 0.5$ and $a_{11} > 0.5$, we are essentially casting the hidden states of a binary HMM to have at least a slight tendency to persist. Secondly, since the rigid-state should emit a low bendability signal more likely than the bendable-state, the HMM identifiability constraint $b_{0(0)} > b_{1(0)}$ on the outcome-emission probabilities makes sense. Equivalently, $b_{1(1)} > b_{0(1)}$ is a sensible constraint, and it is a consequence of $b_{0(0)} > b_{1(0)}$ (NOTE: $b_{0(0)} > b_{1(0)} \Leftrightarrow b_{0(0)} > 1 - b_{1(1)} \Leftrightarrow b_{1(1)} > b_{0(1)}$).

Now, we demonstrate the use of the double runs statistic (N_s, R) and the probabilistic profiles of (N_s, R) in conjunction with a binary HMM for the prediction of the start site of transcription in the two previously mentioned individual DNA sequences (i.e. a sequence from GenBank locus HUMMETIPG in DcSet and a DNA sequence from GenBank locus HSDYSE51 in DncSet). For the HUMMETIPG sequence in terms of binary H10 signals, the observed double runs statistic $(N_s, R) = (142, 62)$. Based on the $25 \times 36 = 900$ probabilistic profiles created in §4.3.2 (p.84–), the MLE-trapping scheme offers 25 screened initial estimates for the EM algorithm (Refer to Table 4.6).

Table 4.6: Initial EM Estimates, $\widehat{\boldsymbol{\theta}}^{(0)}$'s, Offered by the MLE-Trapping Scheme for Training a Binary HMM to Fit the H10 signals of the HUMMETIPG Sequence. (NOTE: 25 2-D trapping grids are built. Each trapping grid is conditioning on a unique (a_{00}, a_{11}) pair, where a_{00} & a_{11} equal to either $0.55, 0.65, 0.75, 0.85$, or 0.95. An individual trapping grid is built from probabilistic profiles of (N_s, R) under the 36 combinations of different values of $b_{0(0)}$ and $b_{1(1)}$ in Table 4.3. The double head arrow "↞" indicates the initial EM estimate which is closest to the MLE.)

$\widehat{a_{00}}^{(0)}$	$\widehat{a_{11}}^{(0)}$	$\widehat{b_{0(0)}}^{(0)}$	$\widehat{b_{1(1)}}^{(0)}$		$\widehat{a_{00}}^{(0)}$	$\widehat{a_{11}}^{(0)}$	$\widehat{b_{0(0)}}^{(0)}$	$\widehat{b_{1(1)}}^{(0)}$
0.55	0.55	0.9	0.4		0.65	0.55	0.9	0.5
	0.65	0.9	0.4			0.65	0.9	0.4
	0.75	0.9	0.4			0.75	0.9	0.4
	0.85	0.9	0.3			0.85	0.9	0.3
	0.95	0.9	0.3			0.95	0.9	0.3
0.75	0.55	0.9	0.6		0.85	0.55	0.9	0.7
	0.65	0.9	0.6			0.65	0.9	0.6
	0.75	0.9	0.5			0.75	0.9	0.5
	0.85	0.9	0.4			0.85	0.9	0.4
	0.95	0.9	0.3			0.95	0.9	0.3
0.95	0.55	0.9	0.9					
	0.65	0.9	0.8					
	0.75	0.9	0.7					
	0.85	0.9	0.6					
	0.95	0.9	0.4	↞				

Using these 25 initial estimates for the EM algorithm, we obtain the MLE $(\widehat{a_{00}}, \widehat{a_{11}}, \widehat{b_{0(0)}}, \widehat{b_{1(1)}}) = (0.983, 0.998, 0.888, 0.417)$. We then calculate the probability of the state being bendable at each position given the observed sequence $y_{[1,501]}$, i.e. the HMM decoding probability $Pr(X_t = 1|Y_{[1,501]})$, to reconstruct the underlying hidden process of the sequence. Similarly, for the HSDYSE51 sequence in terms of binary L6 signals, the observed double runs statistic $(N_s, R) = (408, 44)$. The MLE-trapping scheme offers another set of 25 screened initial estimates for the EM algorithm, and we obtain $(\widehat{a_{00}}, \widehat{a_{11}}, \widehat{b_{0(0)}}, \widehat{b_{1(1)}}) = (0.990, 0.992, 0.319, 0.940)$. Plots of $Pr(X_t = 1|Y_{[1,501]})$ for the HUMMETIPG sequence and the HSDYSE51 sequence are shown in Figure 4.12.

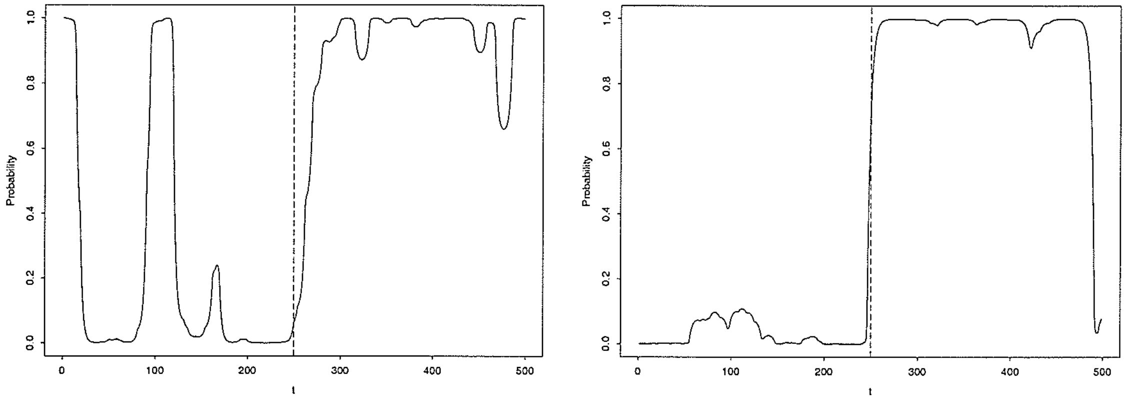


Figure 4.12: Plots of HMM Decoding Probabilities. $Pr(X_t = 1|Y_{[1,501]})$ for a sequence from the GenBank locus HUMMETIPEG (Left) & a sequence from the GenBank locus HSDYSE51 (Right). (NOTE: Dashed line indicates the true start site of transcription, t is the sequence position index)

Each plot in Figure 4.12 reveals a clear persistent bendable region of length about 250 nt. This bendable region starts sharply at about the mid-point of the sequence, so it suggests the location of the start site of transcription in the neighbourhood of position 251. We note that it coincides rather closely with the true start site of transcription of the sequence. Based on 100 bootstrap replications of the fitted binary HMM for the HUMMETIPG (or HSDYSE51) sequence, we obtain the bootstrap estimate for the $\mathcal{V}(\widehat{\boldsymbol{\theta}})$, the 90% percentile interval and the 90% Student's t interval.

Results from the parametric bootstrapping are summarized as follows:

- Almost all entries of $\mathcal{V}(\widehat{\boldsymbol{\theta}}^{\diamond})$ for the HUMMETIPG sequence are bigger (in absolute value) than the corresponding ones for the HSDYSE51 sequence.
- All bootstrap sampling distributions of $\widehat{a_{00}}$ and $\widehat{a_{11}}$ are extremely skewed for both sequences. However, the bootstrap sampling distributions of $\widehat{b_{0(0)}}$ and $\widehat{b_{1(1)}}$ are quite symmetric and normal-shaped for the HSDYSE51 sequence. (Refer to Tables 4.7 & 4.9 and Figures 4.13 & 4.14).
- The discrepancy between the percentile interval and the Student's t interval is generally bigger for the HUMMETIPG sequence. (Refer to Tables 4.8 & 4.10 and Figures 4.13 & 4.14).

All these bootstrap findings are pointing to the fact that the binary HMM is a more robust decoding tool for the HSDYSE51 sequence than for the HUMMETIPG sequence (although it seems performing well for either sequence). Actually, we also see this fact earlier in Figure 4.12. I.e. The plot of HMM decoding probabilities shows a more convincing indication of a clear persistent bendable region of length about 250 nt in the HSDYSE51 sequence than in the HUMMETIPG sequence.

NOTE: $$\mathcal{V}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 0.0191 & 0.0013 & 0.0006 & -0.0017 \\ 0.0013 & 0.0018 & -0.0005 & -0.0024 \\ 0.0006 & -0.0005 & 0.0171 & 0.0002 \\ -0.0017 & -0.0024 & 0.0002 & 0.0046 \end{bmatrix}$$ for the HUMMETIPG Sequence,

and

$$\mathcal{V}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 1.489\times10^{-4} & -6.630\times10^{-5} & -3.098\times10^{-5} & -3.237\times10^{-6} \\ -6.630\times10^{-5} & 1.991\times10^{-3} & 7.342\times10^{-6} & -3.306\times10^{-4} \\ -3.098\times10^{-5} & 7.342\times10^{-6} & 1.463\times10^{-3} & -3.198\times10^{-4} \\ -3.237\times10^{-6} & -3.306\times10^{-4} & -3.198\times10^{-4} & 8.121\times10^{-4} \end{bmatrix}$$

for the HSDYSE51 Sequence.

Table 4.7: Percentiles, Means and Standard Deviations of $\widehat{\theta}^{\diamond}$ Based on 100 Bootstrap Replications of the Fitted Binary HMM for the HUMMETIPG Sequence (NOTE: $\widehat{\pi_0} = 0.0, \widehat{a_{00}} = 0.983, \widehat{a_{11}} = 0.998, \widehat{b_{0(0)}} = 0.888$, and $\widehat{b_{1(1)}} = 0.417$ are fed into the HMM-MLE bootstrap engine)

HMM	Percentile							Mean	Std. Dev.
$\widehat{\theta}^{\diamond}$	5%	10%	25%	50%	75%	90%	95%	$\overline{\widehat{\theta}^{\diamond}}$	$sd(\widehat{\theta}^{\diamond})$
$\widehat{a_{00}}^{\diamond}$	0.600	0.640	0.778	0.911	0.979	0.991	0.994	0.859	0.1383
$\widehat{a_{11}}^{\diamond}$	0.934	0.976	0.990	0.996	0.998	0.999	1.000	0.985	0.0422
$\widehat{b_{0(0)}}^{\diamond}$	0.601	0.632	0.758	0.885	0.941	0.995	0.999	0.847	0.1307
$\widehat{b_{1(1)}}^{\diamond}$	0.375	0.381	0.403	0.422	0.447	0.479	0.503	0.434	0.0678

Table 4.8: The 90% Percentile Intervals & Student's t Intervals for θ Based on 100 Bootstrap Replications of the Fitted Binary HMM for the HUMMETIPG Sequence (NOTE: $\widehat{\pi_0} = 0.0, \widehat{a_{00}} = 0.983, \widehat{a_{11}} = 0.998, \widehat{b_{0(0)}} = 0.888$, and $\widehat{b_{1(1)}} = 0.417$ are fed into the HMM-MLE bootstrap engine)

HMM	90% Percentile Interval		90% Student's t Interval	
θ	Lower Bound	Upper Bound	Lower Bound	Upper Bound
a_{00}	0.600	0.994	0.756	1.000
a_{11}	0.934	1.000	0.929	1.000
$b_{0(0)}$	0.601	0.999	0.673	1.000
$b_{1(1)}$	0.375	0.503	0.305	0.529

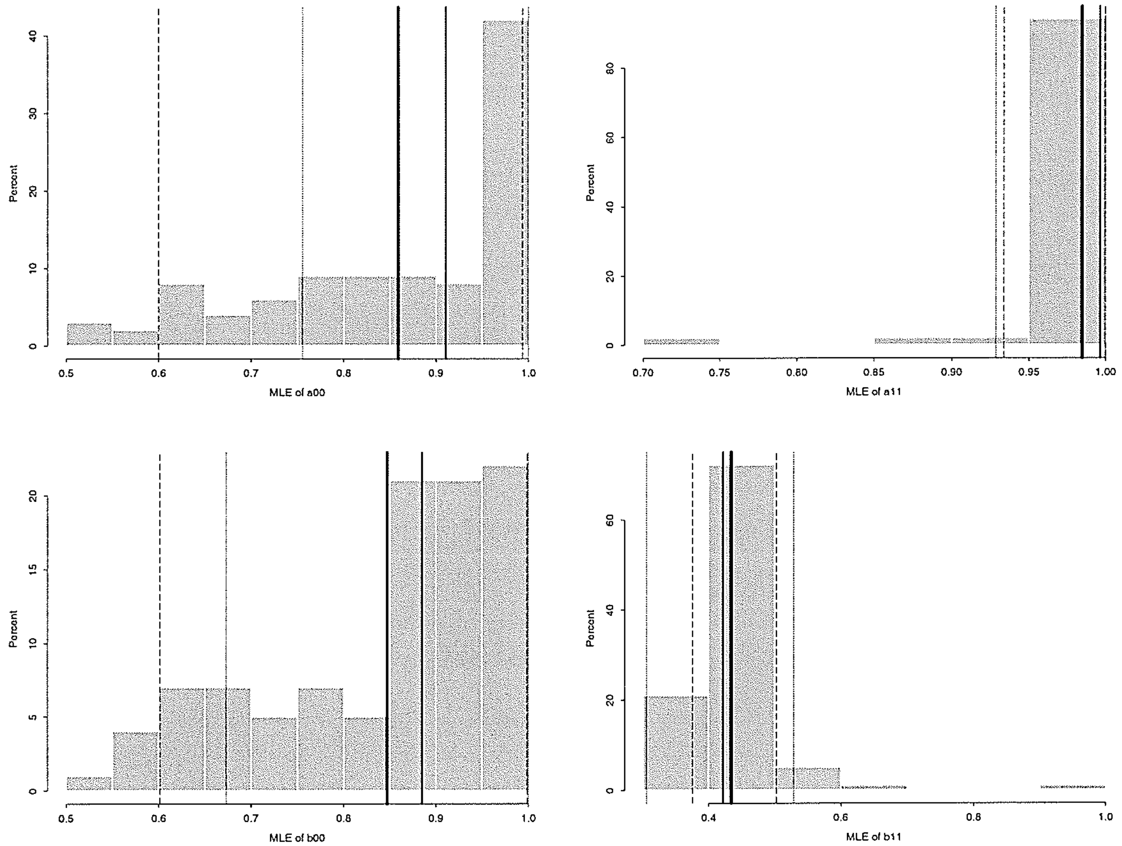


Figure 4.13: Bootstrap Sampling Distributions of $\widehat{\theta}$ of the Fitted Binary HMM for the HUMMETIPEG Sequence. **Top Panel** — Bootstrap Sampling Distribution of $\widehat{a_{00}}$ (Left), Bootstrap Sampling Distribution of $\widehat{a_{11}}$ (Right). **Bottom Panel** — Bootstrap Sampling Distribution of $\widehat{b_{0(0)}}$ (Left), Bootstrap Sampling Distribution of $\widehat{b_{1(1)}}$ (Right). (NOTE: $\widehat{\pi_0} = 0.0, \widehat{a_{00}} = 0.983, \widehat{a_{11}} = 0.998, \widehat{b_{0(0)}} = 0.888$, and $\widehat{b_{1(1)}} = 0.417$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{\theta}^{\diamond}}$ (thick) & $\widehat{\theta}^{\diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{\theta}^{\diamond}_{(5\%)}$ & $\widehat{\theta}^{\diamond}_{(95\%)}$)

Table 4.9: Percentiles, Means and Standard Deviations of $\widehat{\theta}^{\diamond}$ Based on 100 Bootstrap Replications of the Fitted Binary HMM for the HSDYSE51 Sequence (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.990, \widehat{a_{11}} = 0.992, \widehat{b_{0(0)}} = 0.319$, and $\widehat{b_{1(1)}} = 0.940$ are fed into the HMM-MLE bootstrap engine)

HMM	Percentile							Mean	Std. Dev.
$\widehat{\theta}^{\diamond}$	5%	10%	25%	50%	75%	90%	95%	$\overline{\widehat{\theta}^{\diamond}}$	$sd(\widehat{\theta}^{\diamond})$
$\widehat{a_{00}}^{\diamond}$	0.958	0.969	0.985	0.990	0.994	0.996	0.997	0.986	0.0122
$\widehat{a_{11}}^{\diamond}$	0.949	0.966	0.981	0.990	0.994	0.997	0.999	0.978	0.0446
$\widehat{b_{0(0)}}^{\diamond}$	0.262	0.279	0.300	0.319	0.342	0.368	0.381	0.321	0.0382
$\widehat{b_{1(1)}}^{\diamond}$	0.895	0.906	0.926	0.945	0.957	0.971	0.980	0.939	0.0285

Table 4.10: 90% Percentile Intervals & Student's t Intervals for θ Based on 100 Bootstrap Replications of the Fitted Binary HMM for the HSDYSE51 Sequence (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.990, \widehat{a_{11}} = 0.992, \widehat{b_{0(0)}} = 0.319$, and $\widehat{b_{1(1)}} = 0.940$ are fed into the HMM-MLE bootstrap engine)

HMM	90% Percentile Interval		90% Student's t Interval	
θ	Lower Bound	Upper Bound	Lower Bound	Upper Bound
a_{00}	0.958	0.997	0.970	1.000
a_{11}	0.949	0.999	0.919	1.000
$b_{0(0)}$	0.262	0.381	0.256	0.382
$b_{1(1)}$	0.895	0.980	0.893	0.987

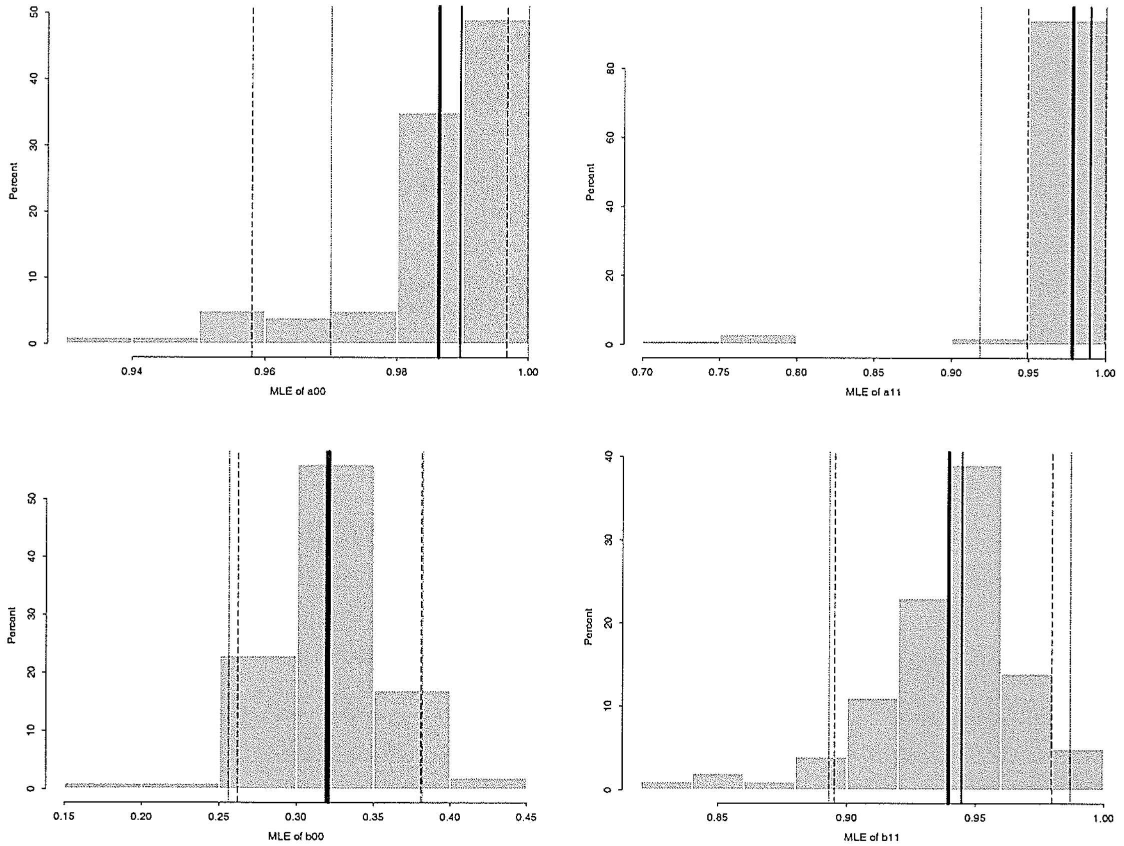


Figure 4.14: Bootstrap Sampling Distributions of $\widehat{\theta}$ of the Fitted Binary HMM for the HSDYSE51 Sequence. **Top Panel** — Bootstrap Sampling Distribution of $\widehat{a_{00}}$ (Left), Bootstrap Sampling Distribution of $\widehat{a_{11}}$ (Right). **Bottom Panel** — Bootstrap Sampling Distribution of $\widehat{b_{0(0)}}$ (Left), Bootstrap Sampling Distribution of $\widehat{b_{1(1)}}$ (Right). (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.990, \widehat{a_{11}} = 0.992, \widehat{b_{0(0)}} = 0.319$, and $\widehat{b_{1(1)}} = 0.940$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{\theta}^{\diamond}}$ (thick) & $\widehat{\theta}^{\diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{\theta}^{\diamond}_{(5\%)}$ & $\widehat{\theta}^{\diamond}_{(95\%)}$)

If we restrict only one change-point between the homogeneous regions (e.g. once the hidden state process is in the bendable-state, it persists and does not switch back to the rigid-state), the binary HMM becomes a single change-point model (as in Churchill, 1989 [31]). Repeating the analysis with a single change-point model to prevent transitions from the bendable-state to the rigid-state, i.e. the binary HMM with $a_{10} = 0$, the MLE-trapping scheme is now based on $5 \times 36 = 180$ probabilistic profiles with $a_{11} = 1$ and offers 5 initial estimates for the EM algorithm. As a result, $(\widehat{a_{00}}, \widehat{b_{0(0)}}, \widehat{b_{1(1)}}) = (0.992, 0.797, 0.430)$ for the HUMMETIPG sequence, and $(\widehat{a_{00}}, \widehat{b_{0(0)}}, \widehat{b_{1(1)}}) = (0.995, 0.293, 0.928)$ for the HSDYSE51 sequence. Again, we can calculate $Pr(X_t = 1|Y_{[1,501]})$ to reconstruct the underlying hidden process of the sequence. In addition, $Pr(X_t = 0, X_{t+1} = 1|Y_{[1,501]})$ can be interpreted as the posterior density of the change-point at position t under the single change-point model when the prior distribution for the location of the change-point is taken as geometric with parameter a_{01} (Churchill, 1989 [31]). Plots of $Pr(X_t = 1|Y_{[1,501]})$ and $Pr(X_t = 0, X_{t+1} = 1|Y_{[1,501]})$ are shown in Figure 4.15.

Plots in Figure 4.15 show a smooth distinction between a persistent bendable region and a persistent rigid region. Interpretations of the $Pr(X_t = 1|Y_{[1,501]})$ plots in Figure 4.15 are consistent with the ones in Figure 4.12. Moreover, the HUMMETIPG sequence has the posterior density of the change-point highly concentrated at position 270 with a 90% maximum posterior density region between position 246 and position 303. Similarly, the HSDYSE51 sequence has the posterior density of the change-point highly concentrated at position 247 with a 90% maximum posterior density region between position 245 and position 257. The location of the true start site of transcription is at position 251 and it is inside the 90% maximum posterior density region of either sequence respectively.

Similarly, we obtain the bootstrap estimate for the $\mathcal{V}(\widehat{\boldsymbol{\theta}})$, the 90% percentile interval and the 90% Student's t interval based on 100 bootstrap replications of the fitted single change-point model for the HUMMETIPG (or HSDYSE51) sequence. Results from the parametric bootstrapping are shown as follows. These results are again pointing to the similar fact stated earlier.

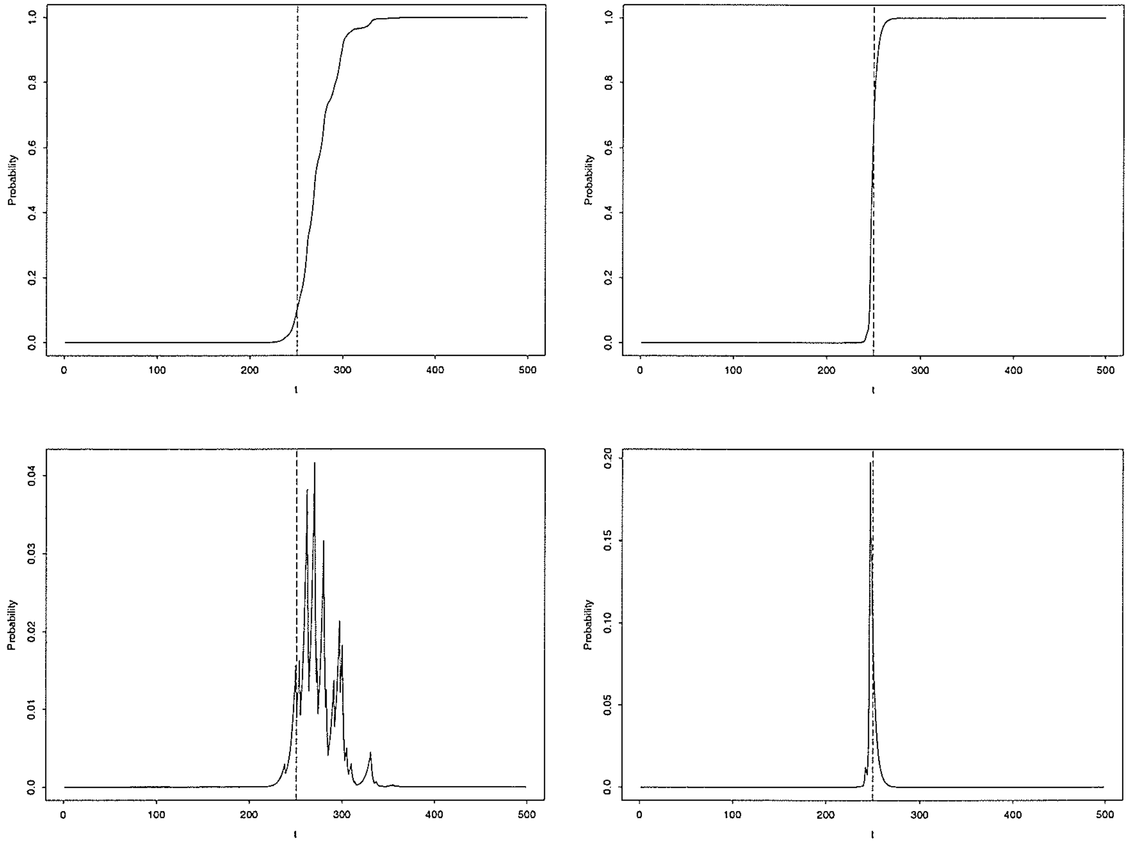


Figure 4.15: Plots of HMM Decoding Probabilities under Single Change-Point Assumption. Probabilities of hidden states of a sequence from the GenBank locus HUMMETIPEG (Left) & a sequence from the GenBank locus HSDYSE51 (Right). **Top Panel** — $Pr(X_t = 1|Y_{[1,501]})$. **Bottom Panel** — $Pr(X_t = 0, X_{t+1} = 1|Y_{[1,501]})$. (NOTE: Dashed line indicates the true start site of transcription, t is the sequence position index)

NOTE: Based on the fitted single change-point model for the HUMMETIPG Sequence,

$$\mathcal{V}(\widehat{\boldsymbol{\theta}}^{\diamond}) = \begin{bmatrix} 0.0066 & 0 & -4.411 \times 10^{-3} & -0.0005 \\ 0 & 0 & 0 & 0 \\ -4.411 \times 10^{-3} & 0 & 2.408 \times 10^{-2} & -4.323 \times 10^{-5} \\ -0.0005 & 0 & -4.323 \times 10^{-5} & 4.336 \times 10^{-4} \end{bmatrix}.$$

Table 4.11: Percentiles, Means and Standard Deviations of $\widehat{\theta}^{\diamond}$ Based on 100 Bootstrap Replications of the Fitted Single Change-Point Model for the HUMMETIPG Sequence (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.992, a_{11} = 1.000, \widehat{b_{0(0)}} = 0.797$, and $\widehat{b_{1(1)}} = 0.430$ are fed into the HMM-MLE bootstrap engine)

HMM	Percentile							Mean	Std. Dev.
$\widehat{\theta}^{\diamond}$	5%	10%	25%	50%	75%	90%	95%	$\overline{\widehat{\theta}^{\diamond}}$	$sd(\widehat{\theta}^{\diamond})$
$\widehat{a_{00}}^{\diamond}$	0.758	0.806	0.869	0.913	0.950	0.974	0.988	0.897	0.0815
$\widehat{b_{0(0)}}^{\diamond}$	0.562	0.575	0.604	0.671	0.871	0.993	0.999	0.735	0.1552
$\widehat{b_{1(1)}}^{\diamond}$	0.393	0.403	0.413	0.431	0.445	0.453	0.457	0.429	0.0208

Table 4.12: 90% Percentile Intervals & Student's t Intervals for θ Based on 100 Bootstrap Replications of the Fitted Single Change-Point Model for the HUMMETIPG Sequence (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.992, a_{11} = 1.000, \widehat{b_{0(0)}} = 0.797$, and $\widehat{b_{1(1)}} = 0.430$ are fed into the HMM-MLE bootstrap engine)

HMM	90% Percentile Interval		90% Student's t Interval	
$\boldsymbol{\theta}$	Lower Bound	Upper Bound	Lower Bound	Upper Bound
a_{00}	0.758	0.988	0.858	1.000
$b_{0(0)}$	0.562	0.999	0.542	1.000
$b_{1(1)}$	0.393	0.457	0.396	0.464

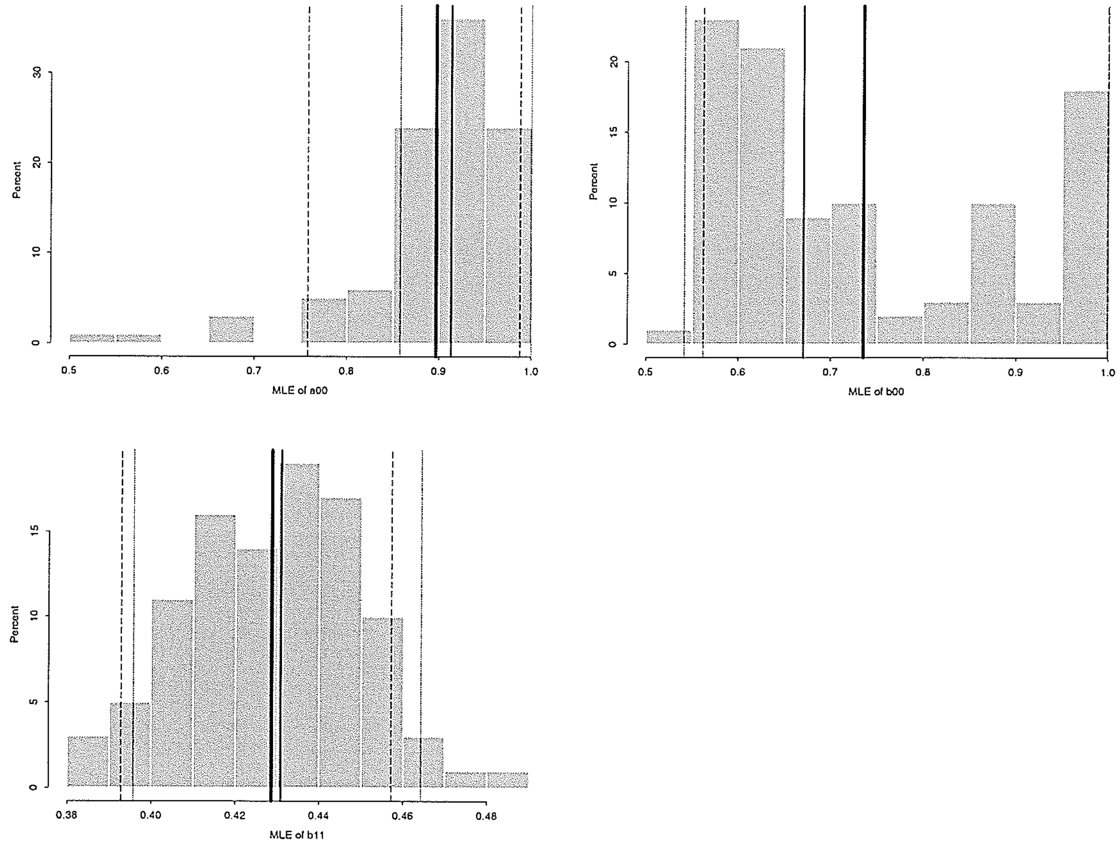


Figure 4.16: Bootstrap Sampling Distributions of $\widehat{\theta}$ of the Fitted Single Change-Point Model for the HUMMETIPG Sequence. **Top Panel** — Bootstrap Sampling Distribution of $\widehat{a_{00}}$ (Left), Bootstrap Sampling Distribution of $\widehat{b_{0(0)}}$ (Right). **Bottom Panel** — Bootstrap Sampling Distribution of $\widehat{b_{1(1)}}$. (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.992, a_{11} = 1.000, \widehat{b_{0(0)}} = 0.797$, and $\widehat{b_{1(1)}} = 0.430$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{\theta}^{\diamond}}$ (thick) & $\widehat{\theta}^{\diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{\theta}^{\diamond}_{(5\%)}$ & $\widehat{\theta}^{\diamond}_{(95\%)}$)

NOTE: Based on the fitted single change-point model for the HSDYSE51 Sequence,

$$\mathcal{V}(\widehat{\theta}^{\diamond}) = \begin{bmatrix} 1.003 \times 10^{-3} & 0 & 1.338 \times 10^{-4} & -1.717 \times 10^{-5} \\ 0 & 0 & 0 & 0 \\ 1.338 \times 10^{-4} & 0 & 0.0038 & -1.249 \times 10^{-4} \\ -1.717 \times 10^{-5} & 0 & -1.249 \times 10^{-4} & 1.369 \times 10^{-3} \end{bmatrix}.$$

Table 4.13: Percentiles, Means and Standard Deviations of $\widehat{\theta}^{\diamond}$ Based on 100 Bootstrap Replications of the Fitted Single Change-Point Model for the HSDYSE51 Sequence. (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.995, a_{11} = 1.000, \widehat{b_{0(0)}} = 0.293$, and $\widehat{b_{1(1)}} = 0.928$ are fed into the HMM-MLE bootstrap engine)

HMM	Percentile							Mean	Std. Dev.
$\widehat{\theta}^{\diamond}$	5%	10%	25%	50%	75%	90%	95%	$\overline{\widehat{\theta}^{\diamond}}$	$sd(\widehat{\theta}^{\diamond})$
$\widehat{a_{00}}^{\diamond}$	0.926	0.952	0.985	0.994	0.996	0.997	0.998	0.983	0.0317
$\widehat{b_{0(0)}}^{\diamond}$	0.219	0.233	0.268	0.293	0.317	0.350	0.406	0.294	0.0620
$\widehat{b_{1(1)}}^{\diamond}$	0.877	0.898	0.917	0.927	0.941	0.952	0.963	0.925	0.0370

Table 4.14: The 90% Percentile Intervals & Student's t Intervals for θ Based on 100 Bootstrap Replications of the Fitted Single Change-Point Model for the HSDYSE51 Sequence. (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.995, a_{11} = 1.000, \widehat{b_{0(0)}} = 0.293$, and $\widehat{b_{1(1)}} = 0.928$ are fed into the HMM-MLE bootstrap engine)

HMM	90% Percentile Interval		90% Student's t Interval	
$\boldsymbol{\theta}$	Lower Bound	Upper Bound	Lower Bound	Upper Bound
a_{00}	0.926	0.998	0.943	1.000
$b_{0(0)}$	0.219	0.406	0.191	0.395
$b_{1(1)}$	0.877	0.963	0.867	0.989

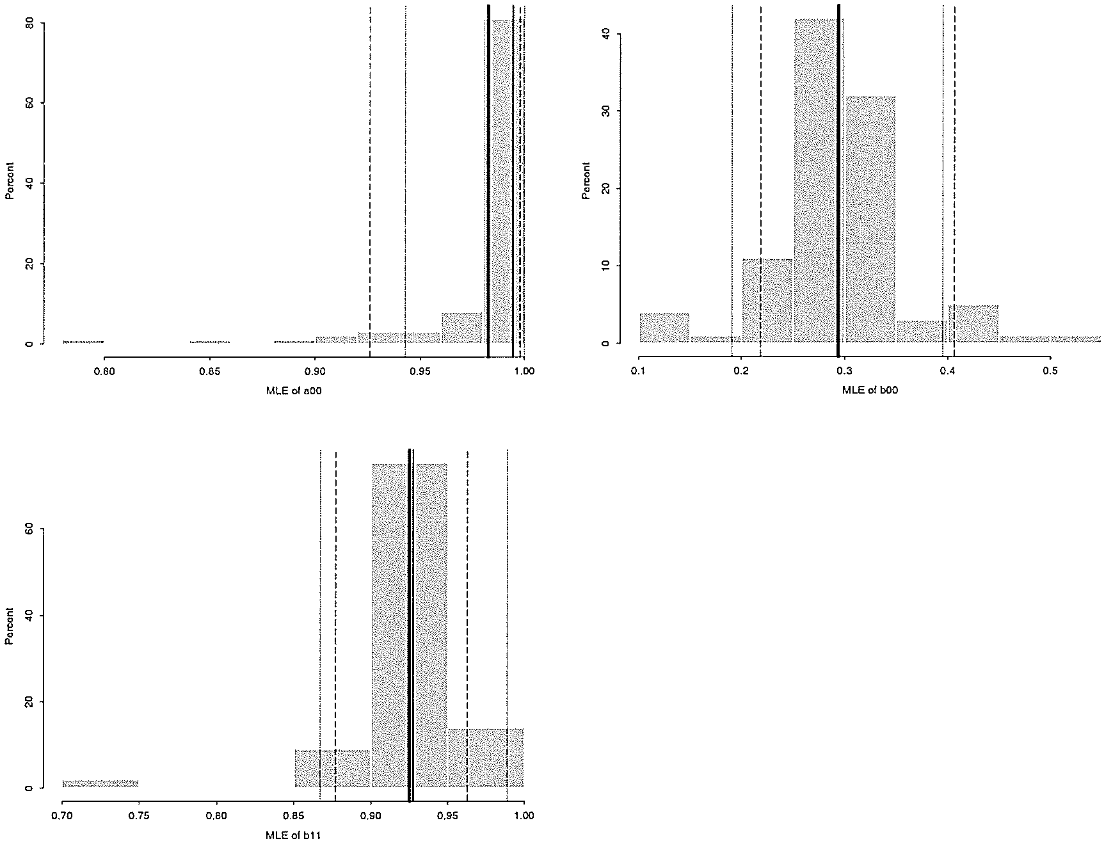


Figure 4.17: Bootstrap Sampling Distributions of $\widehat{\theta}$ of the Fitted Single Change-Point Model for the HSDYSE51 Sequence. **Top Panel** — Bootstrap Sampling Distribution of $\widehat{a_{00}}$ (Left), Bootstrap Sampling Distribution of $\widehat{b_{0(0)}}$ (Right). **Bottom Panel** — Bootstrap Sampling Distribution of $\widehat{b_{1(1)}}$. (NOTE: $\widehat{\pi_0} = 1.0, \widehat{a_{00}} = 0.995, a_{11} = 1.000, \widehat{b_{0(0)}} = 0.293$, and $\widehat{b_{1(1)}} = 0.928$ are fed into the HMM-MLE bootstrap engine. Solid lines indicate $\overline{\widehat{\theta^{\diamond}}}$ (thick) & $\widehat{\theta}^{\diamond}_{(50\%)}$ (thin), dotted line indicates the lower & upper bounds of the 90% Student's t interval, dashed line indicates the $\widehat{\theta}^{\diamond}_{(5\%)}$ & $\widehat{\theta}^{\diamond}_{(95\%)}$)

4.4.3 Concluding Remarks and Implications

Based on the conditional runs tests[†], no sequence appears to have random L6 bendability signals. However, two sequences from the same GenBank locus HSU52111, the *Homo sapiens* X28 region which nears the adrenoleukodystrophy (ALD) protein locus, appear to have random H10 bendability signals. On the surface, there are many different repeat families within this HSU52111 locus, so further investigations on this locus may be worthwhile. Although non-random clustering of H10 (or L6) signals is in most sequences in the data sets, only certain individual sequences (e.g. the HUMMETIPG sequence and the HSDYSE51 sequence) have more high H10 (or L6) bendability signals in the downstream region than in the upstream region. In fact, there are 183 out of 262 sequences (about 70%) in DcSet and 251 out of 362 sequences (about 70%) in DncSet have more high H10 signals in the downstream region. Similarly, 208 out of 262 sequences (about 80%) in DcSet and 235 out of 362 sequences (about 65%) in DncSet have more high L6 signals in the downstream region.

When a DNA sequence has distinct homogeneous bendability regions with the preferential usage of high H10 (or L6) bendability signals in the downstream region, the use of $Pr(X_t = 1|Y_{[1,501]})$ to reconstruct the hidden process based on a simple binary HMM indeed helps reveal the start site of transcription. The MLE-trapping scheme, based on probabilistic profiles of the double runs statistic (N_s, R), substantially enhances the EM algorithm in training an HMM. With this trapping strategy, the EM algorithm reaches convergence very rapidly (about 10 iterations with each initial estimate) in our analysis.

[†] NOTE: More formally, an adjustment (e.g. by the Bonferroni procedure) for multiple testing should be implemented to protect an overall desired level of significance when we are considering multiple sequences.

Chapter 5

PROBABILISTIC MODELS FOR DNA: A MULTIVARIATE PORTRAYAL

5.1 The Logic and Impetus

Having reviewed the literature on biochemical and biophysical features of DNA molecules and the literature on hidden Markov models, I have a vision of a novel *multivariate* class of HMMs, which I named hidden multivariate Markov models or abbreviated as HM^{3}s, for pattern recognition in genomic DNA sequences (especially for recognition of promoter regions of eukaryotic genes). As more and more biochemical and biophysical evidence indicates that DNA molecules possess many different aspects beyond their compositional content, creating probabilistic models from a *"multi-dimensional/multivariate perspective"* makes natural biological sense for modelling DNA. Based on the success of the typical HMMs, I believe that this new generation of probabilistic models, HM^{3}s, will help capture additional features and provide more insights among multiple aspects of DNA molecules. An obvious main advantage of these multivariate models over their univariate "parents" is the opening of the door for multivariate analyses of different *joint* behaviours of different aspects of DNA hidden in various biochemical and biophysical findings/data. These multivariate analyses would not be possible in the univariate setting.

Since DNA three-dimensional structure has been found to be influenced by the exact nucleotide sequence and is important for transcriptional control of gene expression (Hunter, 1993, 1996 [69, 70]; Goodsell & Dickerson, 1994 [62]; Brukner et al., 1995 [25]; Wolffe & Drew, 1995 [151]; Wolffe, 1998 [150]), researchers have been trying to capture additional features of DNA through structural analyses (Benham, 1996, 1999 [16, 17]; Karas et al., 1996 [72]; Baldi et al., 1998 [8]; Pedersen et al., 1998, 1999 [109, 110]). Among these researchers, Benham* introduced two interesting computational methods, which incorporate concepts in thermodynamics and DNA topology, for the prediction of DNA regulatory regions. His ambitious methods are reported as more sensitive than most of the other existing sequence-based computational methods for predicting DNA regulatory regions (Benham, 1996, 1999 [16, 17]). Although my research thinking on the development of hidden multivariate Markov models (HM^{3}s) is quite different from Benham's work, his work (especially his 1996 paper) has enlightened me to incorporate structural information of DNA molecules to hope to further improve existing sequence-based methods for pattern recognition in DNA sequences. In the light of the fact that two important aspects of genomic DNA sequences, a base-compositional aspect such as C+G richness and a structural aspect such as bendability, are now individually found to have a connection with promoter regions of eukaryotic genes (Antequera & Bird, 1993 [3]; Bernardi, 1993, 1995 [18, 19]; Cross & Bird, 1995 [37]; Pedersen et al., 1998, 1999 [109, 110]), a bivariate class of HM^{3}s is proposed.

*Dr. Craig J. Benham (Ph.D. in Mathematics) is a professor and the acting chair of the Department of Biomathematical Sciences at the Mount Sinai School of Medicine of the City University of New York in New York, U.S.A.

Developing the theory of a bivariate class of hidden multivariate Markov models is an attempt to first establish some groundwork for the theory of "full-blown" hidden multivariate Markov models HM^{3}s†. The bivariate HM^{3}s offer a new statistical framework within which the joint behaviour of the C+G richness pattern and the bendability pattern of DNA sequences can be explored. They may help shed new light on computational promoter prediction and gene identification for experimentally uncharacterized DNA Sequences. Since knowing the position of a promoter can delineate or approximate one end of the corresponding gene, improvements on the computational prediction of promoters will in turn help develop "smarter" computer systems for gene identification. Furthermore, they may aid in getting more insights about the transcriptional control of gene expression and genome organization. Since HM^{3}s could be considered as a multivariate version of the typical hidden Markov models (HMMs), it is possible to develop computer algorithms analogous to the ones used under HMMs for HM^{3}s.

5.2 Modelling DNA by Hidden Multivariate Markov Models (HM^{3}s): The Theory

5.2.1 Designs of Model Structures: Definition and Notation

In parallel with the "double layering" structure of a hidden Markov model (HMM), a bivariate hidden multivariate Markov model (HM^3) is defined as a double stochastic process of bivariate vectors. Since it is possible to design and construct many different model structures of an HM^3, three specific model structures have been carefully selected and proposed for the exploration of the joint behaviour of the C+G richness pattern and the bendability pattern of DNA sequences. The essence of this work is to consider/treat a DNA sequence as a sequence of bivariate outcomes, a base-compositional outcome

†NOTE: Although the original vision of HM^{3}s includes a much larger class of probabilistic models, this work is only concerned with the development of a bivariate class of HM^{3}s, so a more appropriate name for them may be hidden bivariate Markov models (HBMMs).

and a structural outcome, which is governed by an underlying unobservable/hidden bivariate process and a set of outcome distributions/densities.

The three selected model structures all share the same design of having the "first layer" of an HM^3 acted as an underlying unobservable/hidden discrete stochastic process of bivariate vectors which is assumed to follow a bivariate Markov chain, and it is denoted as a *bivariate state process* or $\{\mathbf{X}_t = (X_{I_t}, X_{II_t}), t = 1, 2, \dots\}$. However, a design of specific dependence structure among random variables/vectors in the "second layer" of an HM^3 and the random variables/vectors in the "first layer" provides a special overall model structure for the applications to DNA pattern recognition. The "second layer" of each of the three HM^3 model structures is an observable stochastic process of bivariate vectors, whose design will be described in detail for each selected model structure in the following subsections, and it is denoted as a *bivariate outcome process* or $\{\mathbf{Y}_t = (Y_{I_t}, Y_{II_t}), t = 1, 2, \dots\}$. In this subsection, x_{I_t} is a discrete *C+G richness state* (e.g. CG-rich state or CG-poor state), x_{II_t} is a discrete *bendability state* (e.g. bendable state or rigid state), y_{I_t} is a discrete DNA base-compositional outcome (e.g. a discrete value of a binary representation* of the bases such as a strong hydrogen bonding base (C or G = "1") or a weak hydrogen bonding base (A or T = "0")), and y_{II_t} is a continuous DNA structural outcome (e.g. bendability scale derived by Brukner et al., 1995 [25]) at the t-th position along a genomic DNA sequence.

Since recent research on characterizing the bending propensity of DNA has indicated that incorporating more sequence context information may improve models for derivation of sequence-dependent bending propensity parameters, tri- or tetranucleotide-based models† are considered to be important improvements than dinucleotide-based counterparts (Hunter, 1993 [69]; Brukner et al.,

*NOTE: Generally, there are three binary representations that could be considered. **1.** Purine (A or G) versus pyrimidine (C or T). **2.** Strong (C or G) versus weak (A or T) hydrogen bonding. And, **3.** Keto (G or T) versus amino (A or C) forms.

†NOTE: Extension to tetranucleotide-based models will require a much larger set of experimental data on DNase I digestion profiles than the one currently available. Models which incorporate tetranucleotide sequence context information or beyond will be of future research (Brukner et al., 1995 [25]).

1995 [25, 26]). In 1995, Brukner and his colleagues derived the sequence-dependent DNA bending propensity parameters of trinucleotides directly from DNase I digestion data (Refer to Table 5.1). These parameters were also compared with experimental X-ray crystallography data obtained on bent DNA, and they were found to be in good agreement (Brukner et al., 1995 [25]). Since regions of a DNA sequence that are flexible or inherently bent towards the major groove would be more accessible to DNase I digestion, high bendability scales indicate bending towards the major groove. Technically, these bendability scales may be due to either the dynamic property (flexibility) or the

Table 5.1: DNase I Derived Trinucleotide Bendability Scales (Brukner et al., 1995 [25])

Trinucleotide	**Bendability Scale**
AAA/TTT	-0.274
AAC/GTT	-0.205
AAG/CTT	-0.081
AAT/ATT	-0.280
ACA/TGT	-0.006
ACC/GGT	-0.032
ACG/CGT	-0.033
ACT/AGT	-0.183
AGA/TCT	0.027
AGC/GCT	0.017
AGG/CCT	-0.057
ATA/TAT	0.182
ATC/GAT	-0.110
ATG/CAT	0.134
CAA/TTG	0.015
CAC/GTG	0.040
CAG/CTG	0.175
CCA/TGG	-0.246
CCC/GGG	-0.012
CCG/CGG	-0.136
CGA/TCG	-0.003
CGC/GCG	-0.077
CTA/TAG	0.090
CTC/GAG	0.031
GAA/TTC	-0.037
GAC/GTC	-0.013
GCA/TGC	0.076
GCC/GGC	0.107
GGA/TCC	0.013
GTA/TAC	0.025
TAA/TTA	0.068
TCA/TGA	0.194

static property (intrinsic curvature) of the DNA molecule. The work by Brukner and his colleagues has stimulated a number of research studies on DNA bendability such as recent work by Baldi and Pedersen and their colleagues (Baldi et al., 1998, 1999 [8, 6]; Pedersen et al., 1998, 2000 [109, 112]). These recent papers, especially the 1998 paper by Pedersen and his colleagues, have certainly shed a lot of light on my research.

Using the DNA bending propensity parameters of trinucleotides derived by Brukner and his colleagues, it is possible to obtain a sequence of bendability scales for a given DNA sequence. Therefore, each position (except the first and last position) along a given DNA sequence can be represented by not only a discrete base-compositional outcome y_{I_t}, but also a continuous trinucleotide bendability outcome y_{II_t}. For example, we can represent a DNA sequence as a sequence of bivariate outcomes by using the following assignment procedure (Refer to Figure 5.1 for an illustration):

$$y_{I_t} = \begin{cases} 1 & \text{if the nucleotide at the } t\text{-th position has a } \textit{Cytosine} \text{ or a } \textit{Guanine} \\ 0 & \text{otherwise} \end{cases}$$

and

y_{II_t} = bendability scale of the trinucleotide with its middle nucleotide located at the t-th position.

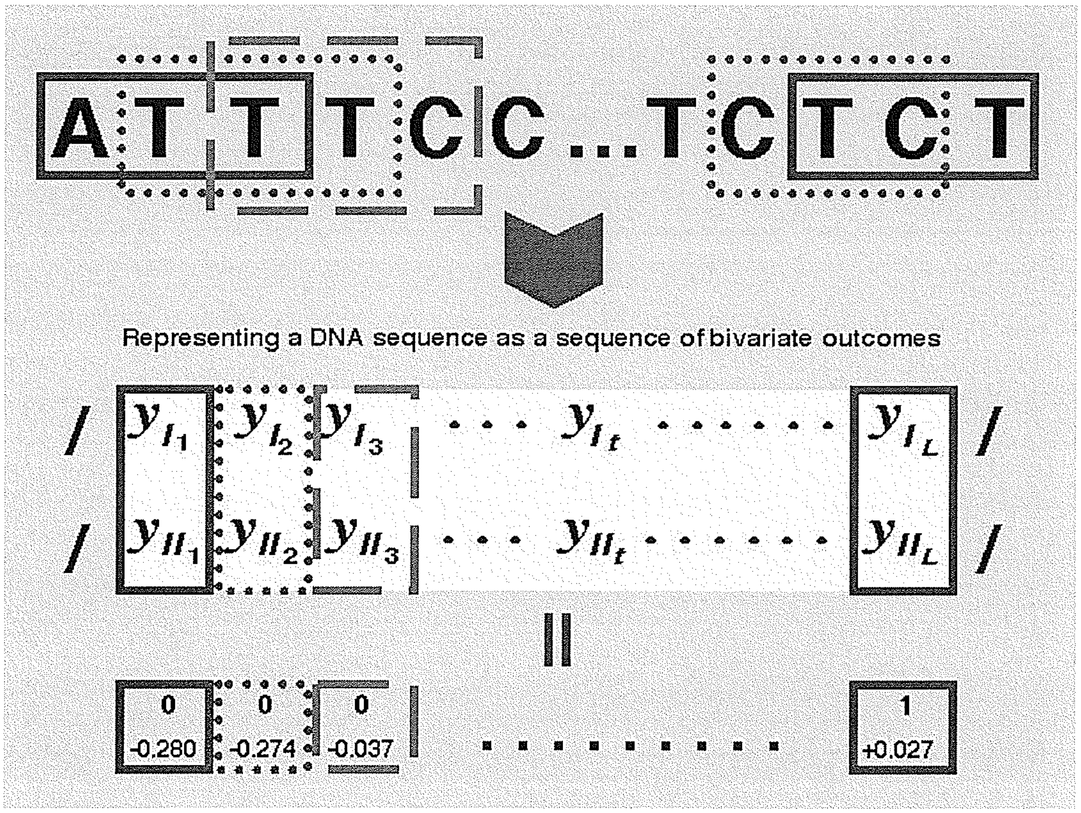

A T T T C C ...T C T C T
Representing a DNA sequence as a sequence of bivariate outcomes
y_{I_1} y_{I_2} y_{I_3} ... y_{I_t} ...... y_{I_L}
y_{II_1} y_{II_2} y_{II_3} ... y_{II_t} ...... y_{II_L}
0 0 0 1
-0.280 -0.274 -0.037 +0.027

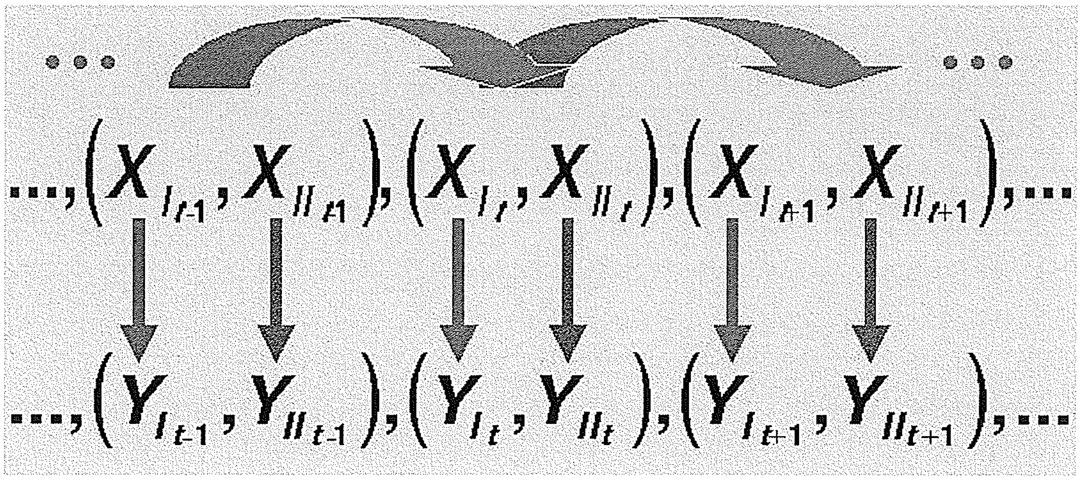
Figure 5.1: Illustration of a Bivariate Data Representation of a DNA Sequence

The "Simplest" Model Structure of an HM^3

The "simplest" model structure of an HM^3 or denoted as the Simplest HM^3 is designed to have the following dependence structure. The "first layer" of the Simplest HM^3 is a hidden discrete bivariate stochastic process which follows a bivariate Markov chain, i.e. the hidden bivariate state process. The "second layer" of the Simplest HM^3 is an observable stochastic process of bivariate vectors with each component of a bivariate outcome random vector (Y_{I_t}, Y_{II_t}) follows a probability distribution/density determined by the corresponding component of the current discrete state vector $((X_{I_t}, X_{II_t}) = (x_{I_t}, x_{II_t}))$ of the hidden bivariate Markov chain (Refer to Figure 5.2).

In summary, assuming the underlying hidden bivariate Markov chain is homogeneous, the Simplest HM^3[‡] with N_I hidden discrete C+G richness states, N_{II} hidden discrete bendability states

[‡] NOTE: One will have the simplest case of the Simplest HM^3 when $N_I = N_{II} = M = 2$ (Refer to page 125).



...
...
$\ldots,(X_{I_{t-1}},X_{II_{t-1}}),(X_{I_t},X_{II_t}),(X_{I_{t+1}},X_{II_{t+1}}),\ldots$
$\ldots,(Y_{I_{t-1}},Y_{II_{t-1}}),(Y_{I_t},Y_{II_t}),(Y_{I_{t+1}},Y_{II_{t+1}}),\ldots$

Figure 5.2: Graphical Representation of the Dependence Structure of the Simplest HM^3

and M observable discrete DNA base-compositional outcomes is fully described by the following elements:

- A set of *hidden discrete C+G richness states*: $\{1,\ldots,N_I\}$; and a set of *hidden discrete bendability states*: $\{1,\ldots,N_{II}\}$
  - Hence, a hidden bivariate Markov chain $\{\mathbf{X}_t=(X_{I_t},X_{II_t}),t=1,2,\ldots\}$ is defined on the bivariate state space $\{1,\ldots,N_I\}\times\{1,\ldots,N_{II}\}$.
- A set of *possible discrete DNA base-compositional outcomes*: $\{1,\ldots,M\}$
  - Hence, the discrete component $\{Y_{I_t},t=1,2,\ldots\}$ of a bivariate outcome process $\{\mathbf{Y}_t=(Y_{I_t},Y_{II_t}),t=1,2,\ldots\}$ takes values in the set $\{1,\ldots,M\}$.
  - And, the continuous component $\{Y_{II_t},t=1,2,\ldots\}$ of the bivariate outcome process $\{\mathbf{Y}_t=(Y_{I_t},Y_{II_t}),t=1,2,\ldots\}$ takes values in the set of real numbers, $\Re$.
- The *initial state distribution* of the hidden bivariate Markov chain as represented by $\pi=(\pi_{(1,1)},\ldots,\pi_{(N_I,N_{II})})$ with

$$\pi_{\mathbf{i}}=Pr(\mathbf{X}_1=\mathbf{i});\qquad \mathbf{i}=(i_I,i_{II})\in\{1,\ldots,N_I\}\times\{1,\ldots,N_{II}\}. \tag{5.1}$$

- Or, $\pi_{(i_I,i_{II})} = Pr((X_{I_1}, X_{II_1}) = (i_I, i_{II}))$.

- The *state-transition distribution* or the *transition probability matrix*, $\mathcal{A}_{\mathbf{X}} = [a_{\mathbf{ij}}]$ or $[a_{((i_I,i_{II})(j_I,j_{II}))}]$, of the hidden bivariate Markov chain with

$$a_{\mathbf{ij}} = Pr(\mathbf{X}_{t+1} = \mathbf{j} | \mathbf{X}_t = \mathbf{i}); \qquad \mathbf{i}, \mathbf{j} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}. \tag{5.2}$$

  - Or, $a_{((i_I,i_{II})(j_I,j_{II}))} = Pr((X_{I_{t+1}}, X_{II_{t+1}}) = (j_I, j_{II}) | (X_{I_t}, X_{II_t}) = (i_I, i_{II}))$.

- The *discrete outcome-emission distribution* of Y_{I_t} conditioning on $X_{I_t} = j_I$ or the *outcome-emission probability matrix*, $\mathcal{B}_{Y_I|X_I} = [b_{j_I(k)}]$, with

$$b_{j_I(k)} = Pr(Y_{I_t} = k | X_{I_t} = j_I); \qquad j_I \in \{1, \ldots, N_I\}, k \in \{1, \ldots, M\}. \tag{5.3}$$

- The *continuous outcome-emission density* of Y_{II_t} conditioning on $X_{II_t} = j_{II}$

$$f_{j_{II}}(y_{II_t}); \qquad j_{II} \in \{1, \ldots, N_{II}\}, y_{II_t} \in \Re. \tag{5.4}$$

  - An attempt to explicitly define $f_{j_{II}}(y_{II_t})$ is to assume that it is from a well-known parametric family of probability density functions.

    E.g. It takes the form of a normal density function:

$$f_{j_{II}}(y_{II_t}) = \frac{1}{\sigma_{j_{II}}\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{y_{II_t} - \mu_{j_{II}}}{\sigma_{j_{II}}})^2}.$$

Now, the parameters of this Simplest HM^3 become $\boldsymbol{\theta} = (\pi, \mathcal{A}_{\mathbf{X}}, \mathcal{B}_{Y_I|X_I}, \phi)$; where ϕ is the parameter set of $f_{j_{II}}(y_{II_t}), \forall j_{II}$ (for example, $\phi = (\mu_1, \ldots, \mu_{N_{II}}, \sigma_1, \ldots, \sigma_{N_{II}})$). The likelihood function $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ of this model can be expressed as follows:

IMPORTANT REMARK

$Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]} | \boldsymbol{\theta})$ is now representing a rather complex function which involves both discrete probability distributions and continuous probability densities.

$$
\begin{aligned}
\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta}) &= Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}|\boldsymbol{\theta}) \\
&= \sum_{\mathrm{x}_1=1}^{N} \cdots \sum_{\mathrm{x}_L=1}^{N} Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}|\mathrm{x}_{[1,L]}, \mathcal{B}_{Y_I|X_I}, \phi) Pr(\mathbf{X}_{[1,L]} = \mathrm{x}_{[1,L]}|\pi, \mathcal{A}_{\mathrm{X}}) \qquad (5.5) \\
&= \sum_{\mathrm{x}_1=1}^{N} \cdots \sum_{\mathrm{x}_L=1}^{N} Pr(Y_{I_{[1,L]}} = y_{I_{[1,L]}}|x_{I_{[1,L]}}, \mathcal{B}_{Y_I|X_I}) \mathcal{L}_{Y_{II}}(\phi) Pr(\mathbf{X}_{[1,L]} = \mathrm{x}_{[1,L]}|\pi, \mathcal{A}_{\mathrm{X}});
\end{aligned}
$$

where

$$Pr(Y_{I_{[1,L]}} = y_{I_{[1,L]}}|x_{I_{[1,L]}}, \mathcal{B}_{Y_I|X_I}) = b_{x_{I_1}(y_{I_1})} \cdot b_{x_{I_2}(y_{I_2})} \cdots b_{x_{I_L}(y_{I_L})} = \prod_{t=1}^{L} b_{x_{I_t}(y_{I_t})};$$

$$\mathcal{L}_{Y_{II}}(\phi) = f_{x_{II_1}}(y_{II_1}) \cdot f_{x_{II_2}}(y_{II_2}) \cdots f_{x_{II_L}}(y_{II_L}) = \prod_{t=1}^{L} f_{x_{II_t}}(y_{II_t});$$

and

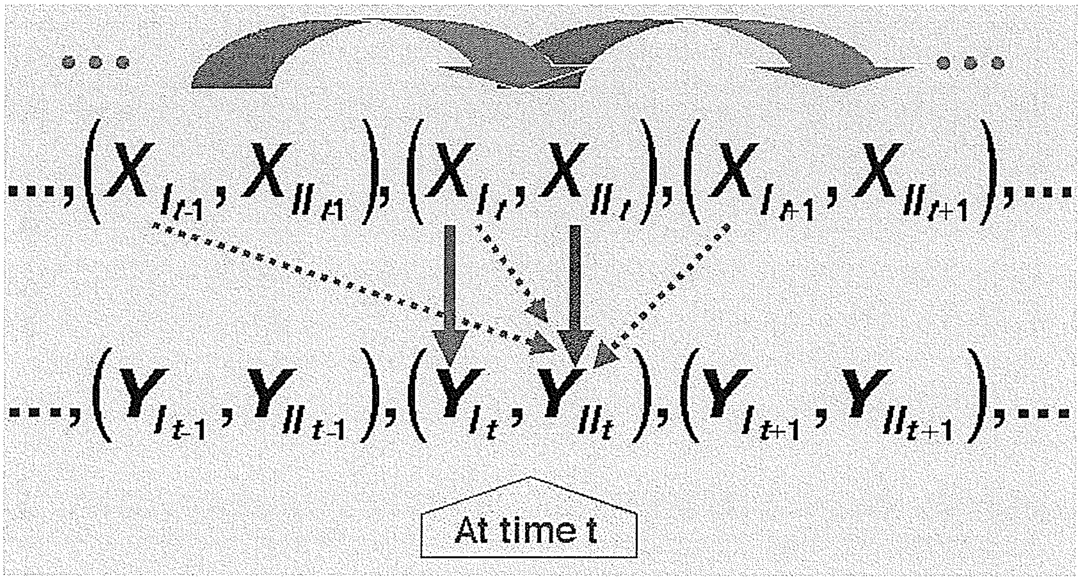
$$Pr(\mathbf{X}_{[1,L]} = \mathrm{x}_{[1,L]}|\pi, \mathcal{A}_{\mathrm{X}}) = \pi_{\mathrm{x}_1} \cdot a_{\mathrm{x}_1\mathrm{x}_2} \cdots a_{\mathrm{x}_{L-1}\mathrm{x}_L} = \pi_{\mathrm{x}_1} \prod_{t=2}^{L} a_{\mathrm{x}_{t-1}\mathrm{x}_t}.$$

" Triplet" Model Structures of an HM^3

Since the Simplest HM^3 ignores the finest details of the dependency between a particular bendability scale and its specific trinucleotide base combination. More elaborate model structures may be of interest. In particular, two different HM^3 model structures are designed to try to reflect the finest details of the dependency between the bendability scale and its trinucleotide base combination. They are defined as "triplet" model structures of an HM^3. One is named as the X_I–Triplet HM^3 (Refer to Figure 5.3), and the other is named as the Y_I–Triplet HM^3 (Refer to Figure 5.4).

As in the Simplest HM^3, the "first layer" of the X_I–Triplet HM^3 is a hidden bivariate Markov chain. However, the "second layer" of the X_I–Triplet HM^3 is an observable bivariate process with the discrete outcome random variable Y_{I_t} depends only on the corresponding hidden state $X_{I_t} = x_{I_t}$, and the continuous outcome random variable Y_{II_t} depends on the corresponding hidden state $X_{II_t} = x_{II_t}$ and three other nearest hidden states $X_{I_{t-1}} = x_{I_{t-1}}, X_{I_t} = x_{I_t}$, and $X_{I_{t+1}} = x_{I_{t+1}}$. Specifically, Y_{I_t} follows a probability distribution determined by the current state x_{I_t}, and Y_{II_t} follows a probability density determined by the states $x_{II_t}, x_{I_{t-1}}, x_{I_t}$, and $x_{I_{t+1}}$ (For listability purpose, I use $x_{I_{[t-1,t+1]}}$ to represent $x_{I_{t-1}}, x_{I_t}, x_{I_{t+1}}$).

Again, as in the Simplest HM^3, the "first layer" of the Y_I–Triplet HM^3 is a hidden bivariate



$\ldots,(X_{I_{t-1}},X_{II_{t-1}}),(X_{I_t},X_{II_t}),(X_{I_{t+1}},X_{II_{t+1}}),\ldots$
$\ldots,(Y_{I_{t-1}},Y_{II_{t-1}}),(Y_{I_t},Y_{II_t}),(Y_{I_{t+1}},Y_{II_{t+1}}),\ldots$
At time t

Figure 5.3: Graphical Representation of the Dependence Structure of the X_I–Triplet HM^3

Markov chain. However, the "second layer" of the Y_I–Triplet HM^3 is an observable bivariate process with the discrete outcome random variable Y_{I_t} depends only on the corresponding hidden state $X_{I_t} = x_{I_t}$, and the continuous outcome random variable Y_{II_t} depends on the corresponding hidden state $X_{II_t} = x_{II_t}$ and three other nearest discrete outcomes $Y_{I_{t-1}} = y_{I_{t-1}}, Y_{I_t} = y_{I_t}$, and $Y_{I_{t+1}} = y_{I_{t+1}}$. Specifically, Y_{I_t} follows a probability distribution determined by the current state x_{I_t}, and Y_{II_t} follows a probability density determined by the state x_{II_t}, and the nearest discrete outcomes $y_{I_{t-1}}, y_{I_t}$, and $y_{I_{t+1}}$ (For listability purpose, I use $y_{I_{[t-1,t+1]}}$ to represent $y_{I_{t-1}}, y_{I_t}, y_{I_{t+1}}$).

In summary, the X_I–Triplet HM^3 or the Y_I–Triplet HM^3 with N_I hidden discrete C+G richness states, N_{II} hidden discrete bendability states and M observable discrete DNA base-compositional outcomes is fully described by the similar elements in the Simplest HM^3 except the corresponding continuous outcome-emission density is now involved with more terms.

- The *continuous outcome-emission density* of

  Y_{II_t} conditioning on $X_{II_t} = j_{II}, X_{I_{[t-1,t+1]}} = x_{I_{[t-1,t+1]}}$ in the X_I–Triplet HM^3

$$f_{j_{II}\, x_{I_{[t-1,t+1]}}}(y_{II_t}); \qquad j_{II} \in \{1, \ldots, N_{II}\}, x_{I_{[t-1,t+1]}} \in \{1, \ldots, N_I\}, y_{II_t} \in \Re. \tag{5.6}$$

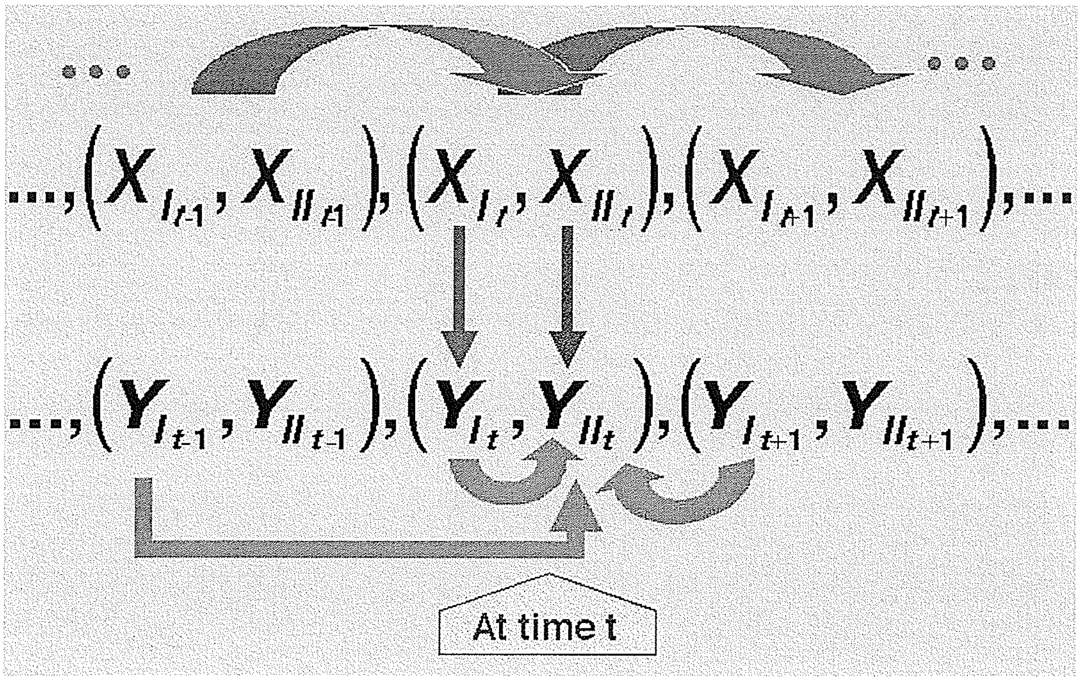

...,$(X_{I_{t-1}}, X_{II_{t-1}}), (X_{I_t}, X_{II_t}), (X_{I_{t+1}}, X_{II_{t+1}})$,...
...,$(Y_{I_{t-1}}, Y_{II_{t-1}}), (Y_{I_t}, Y_{II_t}), (Y_{I_{t+1}}, Y_{II_{t+1}})$,...
At time t

Figure 5.4: Graphical Representation of the Dependence Structure of the Y_I–Triplet HM^3

Y_{II_t} conditioning on $X_{II_t} = j_{II}, Y_{I_{[t-1,t+1]}} = y_{I_{[t-1,t+1]}}$ in the Y_I–Triplet HM^3

$$f_{j_{II}\, y_{I_{[t-1,t+1]}}}(y_{II_t}); \qquad j_{II} \in \{1, \ldots, N_{II}\}, y_{I_{[t-1,t+1]}} \in \{1, \ldots, M\}, y_{II_t} \in \Re. \tag{5.7}$$

- It is now more challenging to explore these continuous outcome-emission densities. Similar to the Simplest HM^3 case, an attempt to explicitly define $f_{j_{II}\, x_{I_{[t-1,t+1]}}}(y_{II_t})$ and $f_{j_{II}\, y_{I_{[t-1,t+1]}}}(y_{II_t})$ is to assume that they are from a well-known parametric family of probability density functions.

- Other attempts, such as making use of non-parametric density estimation techniques and empirical distributions derived from data, may also provide ways to define $f_{j_{II}\, x_{I_{[t-1,t+1]}}}(y_{II_t})$ and $f_{j_{II}\, y_{I_{[t-1,t+1]}}}(y_{II_t})$. Further explorations are left for future research.

All three selected model structures of an HM^3 — the Simplest HM^3, the X_I–Triplet HM^3 and the Y_I–Triplet HM^3 — can be used for modelling DNA. A conceptual framework of applying an HM^3 to capture a base-compositional property (e.g. C+G richness) and a structural property (e.g. trinucleotide-based bendability) of DNA is provided in Figure 5.5.

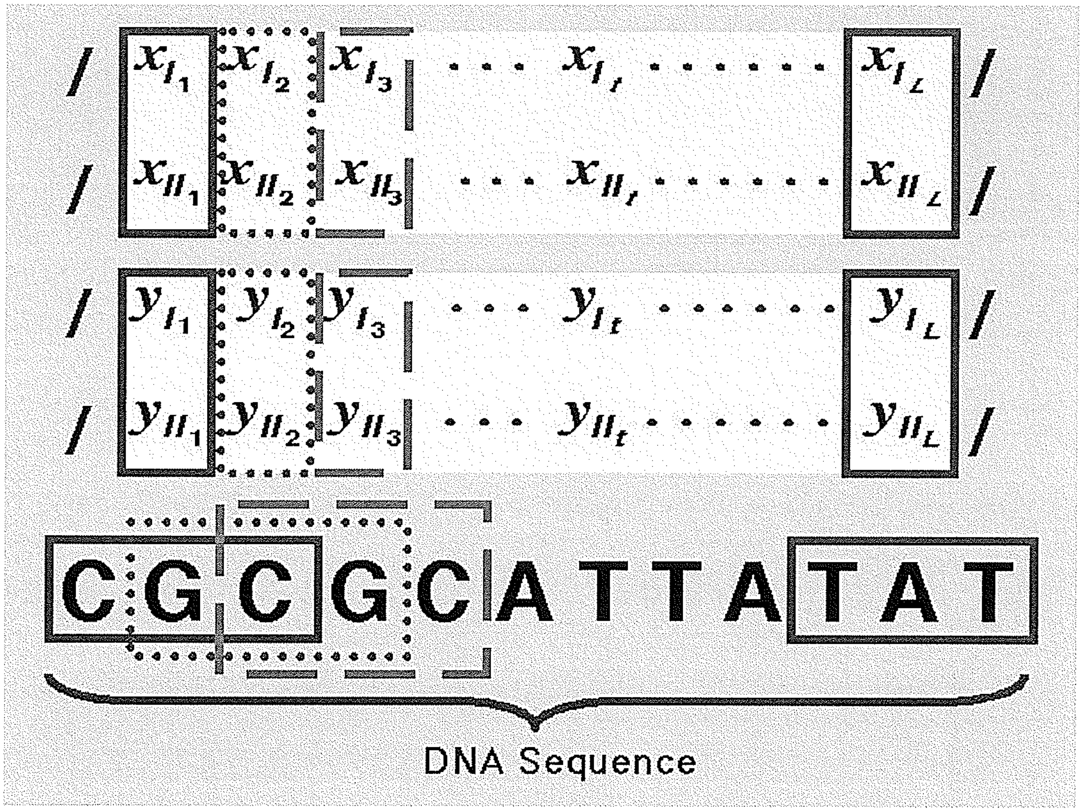


Figure 5.5: Conceptual Framework of an HM^3 for DNA

5.2.2 The Simplest HM^3: The First Step

Since the development of the Simplest HM^3 will form a foundation for the X_I–Triplet HM^3, the Y_I–Triplet HM^3, and other even more complicated HM^{3}s, the focal point is the Simplest HM^3. In parallel with the typical HMMs, three similar problems — the scoring problem, the alignment problem, and the training problem — are associated with the Simplest HM^3. The definitions of these problems are same as in HMMs, except the following:

- The realization of the hidden discrete state sequence is now a bivariate sequence. I.e.

$$\mathrm{x}_{[1,L]} = \{(x_{I_1}, x_{II_1}), \dots, (x_{I_t}, x_{II_t}), \dots, (x_{I_L}, x_{II_L})\}.$$

- The observed outcome sequence is now a bivariate sequence with the first component y_{I_t} being a discrete outcome and the second component y_{II_t} being a continuous outcome at each time/position $t = 1, 2, \ldots, L$. I.e.

$$\mathbf{y}_{[1,L]} = \{(y_{I_1}, y_{II_1}), \ldots, (y_{I_t}, y_{II_t}), \ldots, (y_{I_L}, y_{II_L})\}.$$

- And, the parameter set $\boldsymbol{\theta} = (\pi, \mathcal{A}_{\mathbf{X}}, \mathcal{B}_{Y_I|X_I}, \phi)$ is now even more highly dimensional.

With appropriate modifications, the forward algorithm, the backward algorithm, the Viterbi algorithm, and the EM algorithm can be applied for solving the scoring problem, the alignment problem, and the training problem associated with the Simplest HM^3. An important remark should be emphasized on the interpretations of terms that involve both the discrete and continuous components of the outcome sequence.

IMPORTANT REMARK

For $t = 1, \ldots, L$ and $1 \leq t + T \leq L$;

Similar to the likelihood function $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ or $Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}|\boldsymbol{\theta})$, $Pr(\mathbf{Y}_{[t,t+T]} = \mathbf{y}_{[t,t+T]}|\boldsymbol{\theta})$ is now representing a function which involves both discrete probability distributions and continuous probability densities.

Before proceeding to the details of these problems and their solutions/algorithms, I would like to explicitly re-emphasize the three key model assumptions under the Simplest HM^3 model structure.

SIMPLEST HM^3 ASSUMPTION 1

(I.e. $\mathbf{X}_{[1,L]}$ is a first-order bivariate Markov chain)

For $t = 1, \ldots, L$;

$$Pr(\mathbf{X}_t = \mathbf{x}_t|\mathbf{X}_{[1,t-1]} = \mathbf{x}_{[1,t-1]}, \pi, \mathcal{A}_{\mathbf{X}}) = Pr(\mathbf{X}_t = \mathbf{x}_t|\mathbf{X}_{t-1} = \mathbf{x}_{t-1}, \pi, \mathcal{A}_{\mathbf{X}}).$$

SIMPLEST HM^3 ASSUMPTION 2

(I.e. $Y_{I_{[1,t-1]}}$ & Y_{I_t} are conditionally independent given X_{I_t}.)

For $t = 1, \ldots, L$;

$$Pr(Y_{I_t} = y_{I_t}|X_{I_t} = x_{I_t}, Y_{I_{[1,t-1]}} = y_{I_{[1,t-1]}}, \mathcal{B}_{Y_I|X_I}) = Pr(Y_{I_t} = y_{I_t}|X_{I_t} = x_{I_t}, \mathcal{B}_{Y_I|X_I}).$$

NOTE: The following is also true under the Simplest HM^3 model structure.

$$Pr(Y_{I_t} = y_{I_t} | X_{I_{[1,L]}} = x_{I_{[1,L]}}, Y_{I_{[1,t-1]}} = y_{I_{[1,t-1]}}, \mathcal{B}_{Y_I|X_I}) = Pr(Y_{I_t} = y_{I_t} | X_{I_t} = x_{I_t}, \mathcal{B}_{Y_I|X_I}).$$

SIMPLEST HM^3 ASSUMPTION 3

(I.e. $Y_{II_{[1,t-1]}}$ & Y_{II_t} are conditionally independent given X_{II_t}.)

For $t = 1, \ldots, L$;

$$f_{x_{II_t}\, y_{II_{[1,t-1]}}}(y_{II_t}) = f_{x_{II_t}}(y_{II_t}).$$

NOTE: The following is also true under the Simplest HM^3 model structure.

$$f_{x_{II_{[1,L]}}\, y_{II_{[1,t-1]}}}(y_{II_t}) = f_{x_{II_t}}(y_{II_t}).$$

5.2.3 The Simplest HM^3 Scoring Problem and Its Solution/Algorithm

The scoring problem associated with the Simplest HM^3 is the problem of computing the value of the likelihood function $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ for a bivariate outcome sequence $\mathbf{y}_{[1,L]} = \{(y_I, y_{II})_{[1,L]}\}$ given all the parameters are known. As for the HMM scoring problem, the forward algorithm (with appropriate modifications) can be used for solving the Simplest HM^3 scoring problem. The forward probabilities $\alpha_t(i)$'s defined previously in the HMM subsections can be extended for the Simplest HM^3. They are denoted as $\alpha_t^*(\mathbf{i})$'s; where t is the time/position index (i.e. $t = 1, \ldots, L$), and $\mathbf{i}$ is one of the possible discrete bivariate states of the hidden bivariate Markov chain (i.e. $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$).

DEFINITION

For $t = 1, \ldots, L$ and $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\alpha_t^*(\mathbf{i}) = Pr(\mathbf{Y}_1 = \mathbf{y}_1, \ldots, \mathbf{Y}_t = \mathbf{y}_t, \mathbf{X}_t = \mathbf{i}|\boldsymbol{\theta}) = Pr(\mathbf{Y}_{[1,t]} = \mathbf{y}_{[1,t]}, \mathbf{X}_t = \mathbf{i}|\boldsymbol{\theta}).$$

From an interpretative viewpoint, the $\alpha_t^*(\mathbf{i})$'s are different from the straightforward discrete forward probabilities $\alpha_t(i)$'s. These $\alpha_t^*(\mathbf{i})$'s are termed as the *forward mixes* to reflect their part-discrete-part-continuous "hybrid nature".

NOTE: Analogous to the previous derivations under an HMM, we now have:

$$\begin{aligned}\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta}) &= Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}|\boldsymbol{\theta}) \\ &= \sum_{\text{over all } \mathbf{x}_t} Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}, \mathbf{X}_t = \mathbf{x}_t|\boldsymbol{\theta}) \\ &\qquad \text{(Setting } t = L) \\ &= \sum_{\text{over all } \mathbf{x}_L} \alpha_L^*(\mathbf{x}_L).\end{aligned}$$

And, for $t = 1, \ldots, L-1$;

$$\begin{aligned}\alpha_{t+1}^*(\mathbf{x}_{t+1}) &= Pr(\mathbf{Y}_{[1,t+1]} = \mathbf{y}_{[1,t+1]}, \mathbf{X}_{t+1} = \mathbf{x}_{t+1}|\boldsymbol{\theta}) \\ &\qquad \text{(By Simplest HM}^3\text{ Assumptions 1, 2 \& 3)} \\ &= \Bigg(\sum_{\text{over all } \mathbf{x}_t} \alpha_t^*(\mathbf{x}_t) a_{\mathbf{x}_t\mathbf{x}_{t+1}}\Bigg) b_{x_{I_{t+1}}(y_{I_{t+1}})} f_{x_{II_{t+1}}}(y_{II_{t+1}}).\end{aligned}$$

Although the simplicity of the Simplest HM^3 may allow the use of the forward algorithm described previously for solving the Simplest HM^3 scoring problem, the problem of underflow may prohibit direct use. We have investigated the impact of the sequence length L on the computation of $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ by the forward algorithm under the Simplest HM^3 with $N_I = N_{II} = M = 2$ and $f_{j_{II}}(y_{II_t})$ being a normal density function. Our investigations have shown that the problem of underflow indeed becomes prominent even for relatively short sequence lengths. Specifically, with all model parameters set at moderate size, when the length of the sequence L is about 23, the computation of $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ by the forward algorithm suffers from rather serious underflow. In other words, during the computation process, the $\alpha_t^*(\mathbf{i})$'s get very small when $t \to 23$ and they eventually become indistinguishable from zero (i.e. underflow to zero). Scaling the $\alpha_t^*(\mathbf{i})$'s at each time/position point can avoid the problem of underflow, so a scaling procedure is implemented within the forward algorithm. This forward algorithm with a scaling subroutine will avoid the problem of underflow and will in turn compute the value of the likelihood function $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ for a bivariate outcome sequence $\mathbf{y}_{[1,L]}$ on a logarithm scale. Specific algorithmic details are provided below.

- **Algorithmic Solution**: The recursive forward algorithm with an added scaling subroutine consists of an initialization step, a recursion/induction step, and a termination step. These three steps are explicitly shown below.

1. **Initialization Step:**

   For $\mathbf{i} = (i_I, i_{II}) \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\begin{aligned} \alpha_1^*(\mathbf{i})^{\text{scaled}} &= \alpha_1^*(\mathbf{i}) \\ &= Pr(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{X}_1 = \mathbf{i}|\boldsymbol{\theta}) = Pr(\mathbf{X}_1 = \mathbf{i}|\boldsymbol{\theta}) Pr(\mathbf{Y}_1 = \mathbf{y}_1 | \mathbf{X}_1 = \mathbf{i}, \boldsymbol{\theta}) \qquad (5.8) \\ &= \pi_{\mathbf{i}} b_{i_I(y_{I_1})} f_{i_{II}}(y_{II_1}). \end{aligned}$$

   And,

$$Adjustment_1 = 0. \qquad (5.9)$$

2. **Recursion/Induction Step:**

   For $t = 1, \ldots, L-1$ and $\mathbf{i}, \mathbf{j} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\alpha_{t+1}^*(\mathbf{j})^{\text{core}} = \left(\sum_{\text{over all } \mathbf{i}} \alpha_t^*(\mathbf{i})^{\text{scaled}} a_{\mathbf{ij}} \right) b_{j_I(y_{I_{t+1}})} f_{j_{II}}(y_{II_{t+1}}). \qquad (5.10)$$

   And,

$$Scale_{t+1} = \frac{1}{N_I \times N_{II}} \sum_{\text{over all } \mathbf{j}} \alpha_{t+1}^*(\mathbf{j})^{\text{core}}. \qquad (5.11)$$

   Then, scale $\alpha_{t+1}^*(\mathbf{j})^{\text{core}}$ by the scale factor $Scale_{t+1}$

$$\alpha_{t+1}^*(\mathbf{j})^{\text{scaled}} = \frac{\alpha_{t+1}^*(\mathbf{j})^{\text{core}}}{Scale_{t+1}}. \qquad (5.12)$$

   Also, update the adjustment

$$Adjustment_{t+1} = Adjustment_t + \log Scale_{t+1}. \qquad (5.13)$$

3. **Termination Step:**

$$\begin{aligned}\log \mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta}) &= \log\Bigg(\sum_{\text{over all } \mathbf{i}} \alpha_L^*(\mathbf{i})\Bigg) \\ &= \log\Bigg(\frac{\sum_{\text{over all } \mathbf{i}} \alpha_L^*(\mathbf{i})}{\prod_{t=1}^{L-1} Scale_{t+1}} \prod_{t=1}^{L-1} Scale_{t+1}\Bigg) \\ &= \log\Bigg(\sum_{\text{over all } \mathbf{i}} \alpha_L^*(\mathbf{i})^{\text{scaled}}\Bigg) + Adjustment_L. \end{aligned} \tag{5.14}$$

5.2.4 The Simplest HM^3 Alignment Problem and Its Solution/Algorithm

The alignment problem associated with the Simplest HM^3 is the problem of reconstructing estimates of the sequence of the hidden discrete bivariate states, when we have the observed sequence of bivariate outcomes $\mathbf{y}_{[1,L]}$. Having a bivariate model setting, we can investigate the joint behaviour of the hidden bivariate states to help reveal potentially interesting patterns and structures that we would have missed in individual univariate model settings. This will open the door for not only the incorporation/implementation of existing powerful multivariate statistical techniques, but also the development of innovative multivariate statistical methods to further improve pattern recognition tools.

As for the HMM alignment problem, we can first attack the Simplest HM^3 alignment problem by assuming all the model parameters are "known" (or use the maximum likelihood estimates of the parameters to specify the model). We can follow a similar local approach and a similar global approach discussed previously in the HMM subsections to solve the problem. With appropriate modifications, the backward algorithm and the Viterbi algorithm can be used to obtain a "local" solution and a "global" solution for the Simplest HM^3 alignment problem when the model is fully specified.

A Local Approach for the Simplest HM^3 Alignment Problem

Similar to the local approach mentioned in the HMM alignment problem, a local approach for solving the Simplest HM^3 alignment problem is to find the most probable hidden bivariate state at each time/position t by using the corresponding marginal conditional probability $Pr(\mathbf{X}_t = \mathbf{x}_t | \mathbf{y}_{[1,L]}, \boldsymbol{\theta})$. In other words, it is to find the $\mathbf{x}_t$ which will give the highest $Pr(\mathbf{X}_t = \mathbf{x}_t | \mathbf{y}_{[1,L]}, \boldsymbol{\theta})$ at each time/position t. I.e.

$$\widehat{\mathbf{X}_t} = \underset{\mathbf{x}_t}{\operatorname{argmax}} \, Pr(\mathbf{X}_t = \mathbf{x}_t | \mathbf{y}_{[1,L]}, \boldsymbol{\theta}).$$

In parallel with the HMM case, the key idea of the backward algorithm for solving the Simplest HM^3 local alignment problem is based on rewriting $Pr(\mathbf{X}_t = \mathbf{x}_t | \mathbf{y}_{[1,L]}, \boldsymbol{\theta})$ in terms of the *"forward"* and *"backward"* quantities (i.e. analogous to the probabilities $\alpha_t(i)$'s and $\beta_t(i)$'s under an HMM setting). The forward quantities (i.e. the forward mixes) have been discussed in the previous subsection on the HM^3 scoring problem. Now, I will discuss the backward quantities. The backward probabilities $\beta_t(i)$'s defined previously in the HMM subsections can be extended for the Simplest HM^3. They are denoted as $\beta_t^*(\mathbf{i})$'s; where t is the time/position index (i.e. $t = 1, \ldots, L$), and $\mathbf{i}$ is one of the possible discrete bivariate states of the hidden bivariate Markov chain (i.e. $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$).

DEFINITION

For $t = 1, \ldots, L$ and $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\beta_t^*(\mathbf{i}) = Pr(\mathbf{Y}_{t+1} = \mathbf{y}_{t+1}, \ldots, \mathbf{Y}_L = \mathbf{y}_L | \mathbf{X}_t = \mathbf{i}, \boldsymbol{\theta}) = Pr(\mathbf{Y}_{[t+1,L]} = \mathbf{y}_{[t+1,L]} | \mathbf{X}_t = \mathbf{i}, \boldsymbol{\theta}).$$

The $\beta_t^*(\mathbf{i})$'s are different from the straightforward discrete backward probabilities $\beta_t(i)$'s. To resemble the terminology of *forward mixes* for the $\alpha_t^*(\mathbf{i})$'s, these $\beta_t^*(\mathbf{i})$'s are termed as the *backward mixes* to reflect their part-discrete-part-continuous "hybrid nature".

NOTE: Analogous to the previous derivations under an HMM, we now have:

$$Pr(\mathbf{X}_t = \mathbf{x}_t | \mathbf{y}_{[1,L]}, \boldsymbol{\theta}) = \frac{Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}, \mathbf{X}_t = \mathbf{x}_t | \boldsymbol{\theta})}{Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]} | \boldsymbol{\theta})}$$

(BY SIMPLEST HM^3 ASSUMPTIONS 2 & 3)

$$= \frac{\alpha_t^*(\mathbf{x}_t)\beta_t^*(\mathbf{x}_t)}{\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})}.$$

And, for $t = L-1, \ldots, 1$;

$$\beta_t^*(\mathbf{x}_t) = Pr(\mathbf{Y}_{[t+1,L]} = \mathbf{y}_{[t+1,L]} | \mathbf{X}_t = \mathbf{x}_t, \boldsymbol{\theta})$$

(BY SIMPLEST HM^3 ASSUMPTIONS 1, 2 & 3)

$$= \sum_{\text{over all } \mathbf{x}_{t+1}} b_{x_{I_{t+1}}(y_{I_{t+1}})} f_{x_{II_{t+1}}}(y_{II_{t+1}}) \beta_{t+1}^*(\mathbf{x}_{t+1}) a_{\mathbf{x}_t \mathbf{x}_{t+1}}.$$

Since the backward algorithm is a reverse version of the forward algorithm, so it also suffers from the problem of underflow (as the forward algorithm) during the computation process of the $\beta_t^*(\mathbf{i})$'s. Same scaling trick implemented within the forward algorithm can be applied again for the computations of the backward mixes. Furthermore, I should point out the fact that the scaling procedure added in the forward algorithm is nested within the recursion loop, and the scaling factor at time/position t (i.e. the $Scale_t$) is the same for all forward mixes $\alpha_t^*(\mathbf{i})$'s at a particular t. The same fact also holds for the scaling procedure added in the backward algorithm below. Therefore, scaling the forward and backward mixes will not change their original ranking at a particular time/position, and it preserves the location of the maximum. Specific algorithmic details are provided below.

- **Algorithmic Solution**: The recursive backward algorithm with an added scaling subroutine consists of an initialization step, a recursion/induction step, and a termination step. These three steps are explicitly shown below.

  1. **Initialization Step:**

For $\mathbf{i} = (i_I, i_{II}) \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\begin{aligned} \beta_L^*(\mathbf{i})^{\text{scaled}} &= \beta_L^*(\mathbf{i}) \\ &= 1. \end{aligned} \tag{5.15}$$

2. **Recursion/Induction Step:**

For $t = L-1, \ldots, 1$ and $\mathbf{i}, \mathbf{j} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\beta_t^*(\mathbf{i})^{\text{core}} = \sum_{\text{over all } \mathbf{j}} b_{j_I(y_{I_{t+1}})} f_{j_{II}}(y_{II_{t+1}}) \beta_{t+1}^*(\mathbf{j})^{\text{scaled}} a_{\mathbf{ij}}. \tag{5.16}$$

And,

$$Scale_t = \frac{1}{N_I \times N_{II}} \sum_{\text{over all } \mathbf{i}} \beta_t^*(\mathbf{i})^{\text{core}}. \tag{5.17}$$

Then, scale $\beta_t^*(\mathbf{i})^{\text{core}}$ by the scale factor $Scale_t$

$$\beta_t^*(\mathbf{i})^{\text{scaled}} = \frac{\beta_t^*(\mathbf{i})^{\text{core}}}{Scale_t}. \tag{5.18}$$

3. **Termination Step:**

For $t = 1, \ldots, L$ and $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

We can use the $\alpha_t^*(\mathbf{i})^{\text{scaled}}$ and the $\beta_t^*(\mathbf{i})^{\text{scaled}}$ to find the "pointwise" estimate for each $\mathbf{X}_t$ by

$$\begin{aligned} \widehat{\mathbf{X}_t} &= \operatorname*{argmax}_{\mathbf{i}} Pr(\mathbf{X}_t = \mathbf{i} | \mathbf{y}_{[1,L]}, \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\mathbf{i}} \frac{\alpha_t^*(\mathbf{i}) \beta_t^*(\mathbf{i})}{\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})} \\ &\qquad (\because \mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta}) \text{ is a constant}) \\ &= \operatorname*{argmax}_{\mathbf{i}} \alpha_t^*(\mathbf{i}) \beta_t^*(\mathbf{i}) \\ &\qquad (\because \text{Scaling won't change the location of the maximum}) \\ &= \operatorname*{argmax}_{\mathbf{i}} \alpha_t^*(\mathbf{i})^{\text{scaled}} \beta_t^*(\mathbf{i})^{\text{scaled}}. \end{aligned} \tag{5.19}$$

A Global Approach for the Simplest HM^3 Alignment Problem

Similarly, a global approach for solving the Simplest HM^3 alignment problem is to find the most probable hidden bivariate state sequence as a whole by using the full conditional distribution $Pr(\mathbf{X}_{[1,L]} = \mathbf{x}_{[1,L]} | \mathbf{y}_{[1,L]}, \boldsymbol{\theta})$. It can be viewed as an estimation of $\mathbf{X}_{[1,L]}$, and it suggests

$$\widehat{\mathbf{X}_{[1,L]}} = \operatorname*{argmax}_{\mathbf{x}_{[1,L]}} Pr(\mathbf{X}_{[1,L]} = \mathbf{x}_{[1,L]} | \mathbf{y}_{[1,L]}, \boldsymbol{\theta}).$$

We can use the Viterbi algorithm as in §3.2.3 (p.39–), but quantities corresponding to $\delta_t(i)$ and $\psi_t(j)$ are defined on a logarithm scale in order to avoid the numerical underflow problem.

DEFINITION

For $t = 2, \ldots, L$ and $\mathbf{i}, \mathbf{j} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

$$\delta^*_t(\mathbf{j}) = \max_{\mathbf{x}_{[1,t-1]}} \log Pr(\mathbf{X}_{[1,t-1]} = \mathbf{x}_{[1,t-1]}, \mathbf{X}_t = \mathbf{j}, \mathbf{Y}_{[1,t]} = \mathbf{y}_{[1,t]} | \boldsymbol{\theta});$$

and

$$\psi^*_t(\mathbf{j}) = \operatorname*{argmax}_{\mathbf{i}} \left(\delta^*_{t-1}(\mathbf{i}) + \log a_{\mathbf{ij}} \right).$$

- **Algorithmic Solution** (Global Approach): The four steps of the Viterbi algorithm become the following.

1. **Initialization Step:**

   For $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

   $$\delta^*_1(\mathbf{i}) = \log \left(\pi_{\mathbf{i}} b_{i_I(y_{I_1})} f_{i_{II}}(y_{II_1}) \right). \tag{5.20}$$

2. **Recursion/Induction Step:**

   For $t = 2, \ldots, L$ and $\mathbf{i}, \mathbf{j} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

   $$\delta^*_t(\mathbf{j}) = \max_{\mathbf{i}} \left[\delta^*_{t-1}(\mathbf{i}) + \log a_{\mathbf{ij}} \right] + \log \left(b_{j_I(y_{I_t})} f_{j_{II}}(y_{II_t}) \right); \tag{5.21}$$

   and

   $$\psi^*_t(\mathbf{j}) = \operatorname*{argmax}_{\mathbf{i}} \left(\delta^*_{t-1}(\mathbf{i}) + \log a_{\mathbf{ij}} \right). \tag{5.22}$$

3. **Termination Step:**

   For $\mathbf{i} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$;

   $$\max_{\mathbf{x}_{[1,L]}} \log Pr(\mathbf{X}_{[1,L]} = \mathbf{x}_{[1,L]}, \mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]} | \boldsymbol{\theta}) = \max_{\mathbf{i}} \delta^*_L(\mathbf{i}); \tag{5.23}$$

and

$$\widehat{\mathrm{X}_L} = \underset{\mathrm{i}}{\mathrm{argmax}}\, \delta^*_L(\mathrm{i}). \tag{5.24}$$

4. **Path Backtracking Step:**

   For $t = L-1, \dots, 1$;

$$\widehat{\mathrm{X}_t} = \psi^*_{t+1}(\widehat{\mathrm{X}_{t+1}}). \tag{5.25}$$

5.2.5 The Simplest HM^3 Training Problem and Its Solution/Algorithm

Maximum Likelihood Approach for the HM^3 Training Problem

Direct maximization of the likelihood function $\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta})$ is going to be extremely difficult, if not impossible. Treating the hidden bivariate state sequence $\mathbf{X}_{[1,L]}$ as "missing" augmented-data, we can apply the EM algorithm by working with the *augmented-data likelihood function* $\mathcal{L}_{\mathbf{YX}}(\boldsymbol{\theta})$ to estimate the parameters of the HM^3. Under the simplest HM^3 structure, the augmented-data likelihood function is

$$\begin{aligned} \mathcal{L}_{\mathbf{YX}}(\boldsymbol{\theta}) &= Pr(\mathbf{Y}_{[1,L]} = \mathbf{y}_{[1,L]}, \mathbf{X}_{[1,L]} = \mathbf{x}_{[1,L]}|\boldsymbol{\theta}) \\ &= \pi_{\mathbf{x}_1} \prod_{t=1}^{L} b_{x_{I_t}(y_{I_t})} f_{x_{II_t}(y_{II_t})} \prod_{t=2}^{L} a_{\mathbf{x}_{t-1}\mathbf{x}_t}. \end{aligned} \tag{5.26}$$

Taking the logarithm of the augmented-data likelihood function, we have

$$\begin{aligned}
\log \mathcal{L}_{\mathbf{YX}}(\boldsymbol{\theta}) &= \log \pi_{\mathbf{x}_1} + \sum_{t=1}^{L} \log b_{x_{I_t}(y_{I_t})} + \sum_{t=1}^{L} \log f_{x_{II_t}(y_{II_t})} \\
&\quad + \sum_{t=2}^{L} \log a_{\mathbf{x}_{t-1}\mathbf{x}_t} \\
&\quad \text{(Capturing the ``missing data'' by indicator variables } u_{\mathbf{i}}(t)\text{'s,} \\
&\quad u_{j_I}(t)\text{'s, } u_{j_{II}}(t)\text{'s \& } v_{\mathbf{ij}}(t)\text{'s.)} \\
&= \sum_{\mathbf{i}} u_{\mathbf{i}}(1) \log \pi_{\mathbf{i}} + \sum_{j_I} \sum_{t=1}^{L} u_{j_I}(t) \log b_{j_I(y_{I_t})} \\
&\quad + \sum_{j_{II}} \sum_{t=1}^{L} u_{j_{II}}(t) \log f_{j_{II}(y_{II_t})} + \sum_{\mathbf{i}} \sum_{\mathbf{j}} (\log a_{\mathbf{ij}}) \sum_{t=2}^{L} v_{\mathbf{ij}}(t);
\end{aligned} \tag{5.27}$$

$$\text{where for } t = 1, \ldots, L,\ u_{\mathbf{i}}(t) = \begin{cases} 1 & \text{if } \mathbf{x}_t = (x_{I_t}, x_{II_t}) = \mathbf{i} = (i_I, i_{II}), \\ 0 & \text{otherwise;} \end{cases}$$

$$u_{j_I}(t) = \begin{cases} 1 & \text{if } x_{I_t} = j_I, \\ 0 & \text{otherwise;} \end{cases} \qquad u_{j_{II}}(t) = \begin{cases} 1 & \text{if } x_{II_t} = j_{II}, \\ 0 & \text{otherwise;} \end{cases}$$

$$\text{and for } t = 2, \ldots, L,\ v_{\mathbf{ij}}(t) = \begin{cases} 1 & \text{if } \mathbf{x}_{t-1} = \mathbf{i} \ \&\ \mathbf{x}_t = \mathbf{j}, \\ 0 & \text{otherwise.} \end{cases}$$

- **Algorithmic Solution** (Maximum Likelihood Approach): For convenience, we assume that the continuous outcome-emission takes the form of a normal density function, i.e.

$$f_{j_{II}}(y_{II_t}) = \frac{1}{\sigma_{j_{II}} \sqrt{2\pi}}\, e^{-\frac{1}{2}(\frac{y_{II_t} - \mu_{j_{II}}}{\sigma_{j_{II}}})^2}.$$

The EM algorithm for training an HM^3 starts with an initial guess/estimate $\widehat{\boldsymbol{\theta}}^{(0)} = (\widehat{\pi_{\mathbf{i}}}^{(0)}$'s, $\widehat{a_{\mathbf{ij}}}^{(0)}$'s, $\widehat{b_{j_I(k)}}^{(0)}$'s, $\widehat{\mu_{j_{II}}}^{(0)}$'s, $\widehat{\sigma_{j_{II}}}^{(0)}$'s), where $\mathbf{i}, \mathbf{j} \in \{1, \ldots, N_I\} \times \{1, \ldots, N_{II}\}$ and $k \in \{1, \ldots, M\}$. Then the EM cycles begin. For $c = 0, 1, \ldots$, where c is the EM cycle index, each cycle has an expectation step (E-step) and a maximization step (M-step) as follows.

1. **E-step:**

   Replace the "missing data" represented by $u_{\mathbf{i}}(t)$'s, $u_{j_I}(t)$'s, $u_{j_{II}}(t)$'s and $v_{\mathbf{ij}}(t)$'s in the $\log \mathcal{L}_{\mathbf{YX}}(\boldsymbol{\theta})$ by their conditional expectations given the observed outcomes $\mathbf{y}_{[1,L]}$ and the current parameter estimate $\widehat{\boldsymbol{\theta}}^{(c)}$. I.e.

$$\begin{aligned}\widehat{u_{\mathbf{i}}(t)}^{(c)} &= E(u_{\mathbf{i}}(t)|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) = Pr(\mathbf{X}_t = \mathbf{i}|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) \\ &= \frac{\alpha_t^*(\mathbf{i})^{(c)}\beta_t^*(\mathbf{i})^{(c)}}{\mathcal{L}_{\mathbf{Y}}(\widehat{\boldsymbol{\theta}}^{(c)})},\end{aligned} \tag{5.28}$$

$$\begin{aligned}\widehat{u_{j_I}(t)}^{(c)} &= E(u_{j_I}(t)|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) = \sum_{j_{II}} Pr(\mathbf{X}_t = \mathbf{j}|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) \\ &= \sum_{j_{II}} \widehat{u_{\mathbf{j}}(t)}^{(c)},\end{aligned} \tag{5.29}$$

$$\begin{aligned}\widehat{u_{j_{II}}(t)}^{(c)} &= E(u_{j_{II}}(t)|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) = \sum_{j_I} Pr(\mathbf{X}_t = \mathbf{j}|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) \\ &= \sum_{j_I} \widehat{u_{\mathbf{j}}(t)}^{(c)},\end{aligned} \tag{5.30}$$

   and

$$\begin{aligned}\widehat{v_{\mathbf{ij}}(t)}^{(c)} &= E(v_{\mathbf{ij}}(t)|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) = Pr(\mathbf{X}_{t-1} = \mathbf{i}, \mathbf{X}_t = \mathbf{j}|\mathbf{y}_{[1,L]},\widehat{\boldsymbol{\theta}}^{(c)}) \\ &= \frac{\alpha_{t-1}^*(\mathbf{i})^{(c)}\widehat{b_{j_I(y_{I_t})}}^{(c)} f_{j_{II}}(y_{II_t})^{(c)}\beta_t^*(j)^{(c)}\widehat{a_{\mathbf{ij}}}^{(c)}}{\mathcal{L}_{\mathbf{Y}}(\widehat{\boldsymbol{\theta}}^{(c)})},\end{aligned} \tag{5.31}$$

   where $f_{j_{II}}(y_{II_t})^{(c)} = \frac{1}{\widehat{\sigma_{j_{II}}}^{(c)}\sqrt{2\pi}}\, e^{-\frac{1}{2}(\frac{y_{II_t}-\widehat{\mu_{j_{II}}}^{(c)}}{\widehat{\sigma_{j_{II}}}^{(c)}})^2}$. Keeping track of the scale factors during the scaling procedures, the $\alpha_t^*(\mathbf{i})$'s and $\beta_t^*(\mathbf{i})$'s can be reconstructed from their scaled counterparts $\alpha_t^*(\mathbf{i})^{\text{scaled}}$'s and $\beta_t^*(\mathbf{i})^{\text{scaled}}$'s for all t and $\mathbf{i}$.

2. **M-step:**

   Treat the conditional expected values $\widehat{u_j(t)}^{(c)}$ and $\widehat{v_{ij}(t)}^{(c)}$ from the E-step as data and solve the augmented-data likelihood maximization problem to get an updated estimate $\widehat{\boldsymbol{\theta}}^{(c+1)}$. The closed-form solution is

$$\widehat{\pi_{\mathrm{i}}}^{(c+1)} = \widehat{u_{\mathrm{i}}(1)}^{(c)}, \tag{5.32}$$

$$\widehat{a_{\mathrm{ij}}}^{(c+1)} = \frac{\sum_{t=2}^{L} \widehat{v_{\mathrm{ij}}(t)}^{(c)}}{\sum_{t=1}^{L-1} \widehat{u_{\mathrm{i}}(t)}^{(c)}}, \tag{5.33}$$

$$\widehat{b_{j_I(k)}}^{(c+1)} = \frac{\sum_{t=1}^{L} \widehat{u_{j_I}(t)}^{(c)} 1_{y_{I_t}}}{\sum_{t=1}^{L} \widehat{u_{j_I}(t)}^{(c)}}, \tag{5.34}$$

where the indicator variable $1_{y_{I_t}} = 1$ if $Y_{I_t} = k$, otherwise $1_{y_{I_t}} = 0$;

$$\widehat{\mu_{j_{II}}}^{(c+1)} = \frac{\sum_{t=1}^{L} \widehat{u_{j_{II}}(t)}^{(c)} \; y_{II_t}}{\sum_{t=1}^{L} \widehat{u_{j_{II}}(t)}^{(c)}}, \tag{5.35}$$

and

$$\widehat{\sigma^2_{j_{II}}}^{(c+1)} = \frac{\sum_{t=1}^{L} \widehat{u_{j_{II}}(t)}^{(c)} (y_{II_t} - \widehat{\mu_{j_{II}}}^{(c+1)})^2}{\sum_{t=1}^{L} \widehat{u_{j_{II}}(t)}^{(c)}}, \tag{5.36}$$

The EM cycles are iterated until convergence.

5.2.6 The Simplest HM^3 Simulator

Similar to an HMM, the Simplest HM^3 can be viewed as a generative system which eventually produces sequences of bivariate observations or outcomes. With a simulator of the Simplest HM^3, we can examine different properties of parameter estimators, and can investigate model features by conducting simulation studies. A simulator for the Simplest HM^3 has been fully developed and tested. A graphical representation of it is shown in Figure 5.6.

With user-defined HM^3 model parameters, i.e. $\boldsymbol{\theta} = (\pi, \mathcal{A}_{\mathrm{X}}, \mathcal{B}_{Y_I|X_I}, \phi)$; the length of individual sequence, i.e. L; and a desired number of sequences as input for the simulator, several data sets

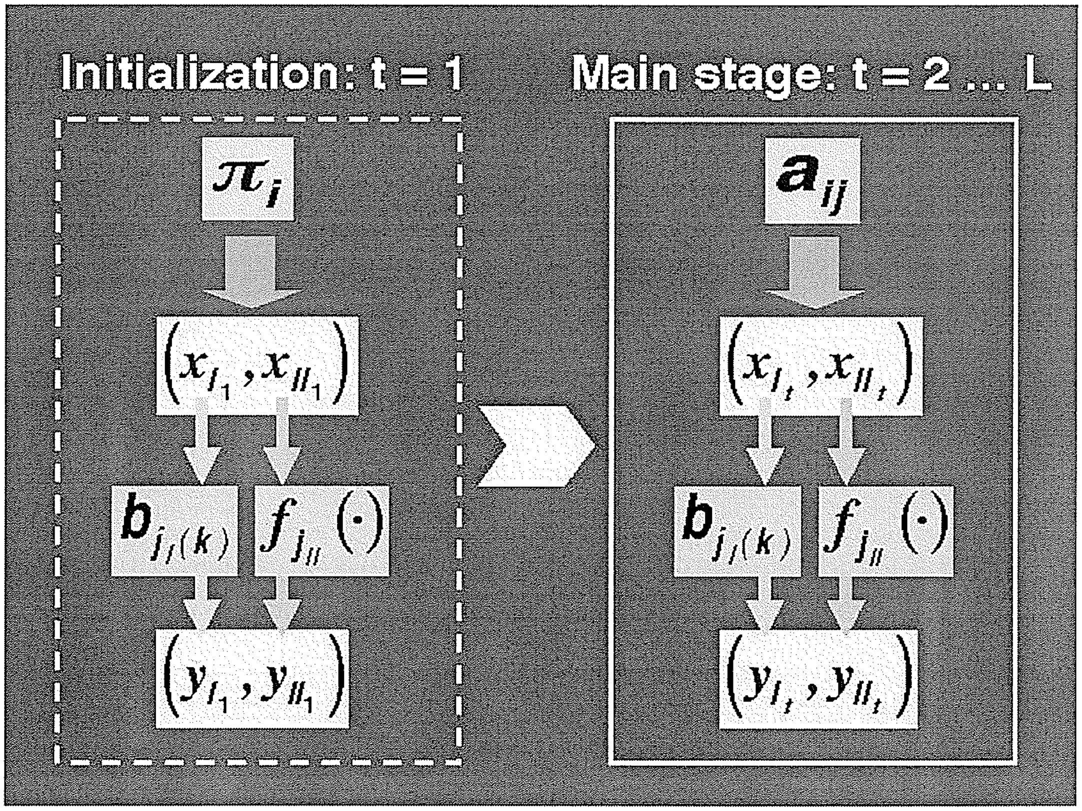


Figure 5.6: Graphical Representation of the Simplest HM^3 Simulator

have been generated under the Simplest HM^3 with $N_I = N_{II} = M = 2$ and $f_{j_{II}}(y_{II_t})$ being a normal density function. Preliminary analyses on the simulated data have helped "visualize" some of the features of the Simplest HM^3. For example, with different sets of model parameters, an informal investigation on the level of randomness within a simulated sequence of length $L = 5000$ has suggested the idea of developing some type of runs-related statistics for capturing certain data structures. Further investigations are left open for future research.

Chapter 6

DISCUSSION AND FUTURE RESEARCH

The main focus of this work has been on the investigation of the double runs statistic (N_s, R) under a binary HMM for DNA pattern recognition. Having studied the double runs statistic under different binary HMM parameter sets, probabilistic profiles of (N_s, R) are created. The creation of the probabilistic profiles of (N_s, R) has essentially provided a "backbone" for the MLE-trapping scheme. With the MLE-trapping scheme, a HMM-MLE bootstrap engine is built and used for constructing simple confidence intervals for the HMM parameters. Applications of the conditional runs statistic, the double runs statistic, and the probabilistic profiles in conjunction with binary HMMs for DNA pattern recognition are demonstrated through the analysis of dichotomized DNA bendability scales. Owing to the continuous nature of the bendability scales, an HMM with continuous outcome-emission densities may be capable of capturing the mosaic bendability structure better than the HMMs used in this thesis. However, the existing FMCI technique and the runs statistics are only defined for sequences of discrete outcomes. The work on developing a continuous Markov (or a semi-Markov) process imbedding technique to analyze a sequence of continuous outcomes still requires some ingenuity, and it is to be explored in the future. Furthermore, other structural features of DNA, such as propeller-twisting of DNA base-pairs, have also been quantified in continuous scales. Hence, extending the simple FMCI concepts to describe patterns in a sequence of continuous outcomes may provide a more direct approach for the analysis of these structural features.

In conclusion, the key contribution of this work is the use of the finite Markov chain imbedding technique to study runs and patterns in a sequence under a hidden Markov model framework. The FMCI technique can certainly be extended to study various multiple runs and patterns of a sequence of polychotomous outcomes under a more complex HMM structure. However, as the length of the sequence L increases and/or the number of parameters increases due to a complex structure, the computational burden also increases. Additional assumptions on model parameters and/or computational tricks will be needed in these cases.

Appendix A

Data Set DcSet: It contains 262 human genomic DNA sequences. Each sequence has coding subsequences in the 250 nt downstream region of the start site of transcription of a RNA Polymerase II transcribed gene.

GenBank Locus	Accession Number
HBNLF1	X58140
HSA1MBG1	X54816
HSA2MGLB1	Z11711
HSACAA1	X65140
HSACKI10	X14487
HSAMY2B1	X07057
HSAPOA2G	X02905
HSAPRT	Y00486
HSARG1	X12662
HSATIH101	X69532
HSB3A	X72861
HSCALRT1	X56668
HSCAM3X1	X52606
HSCATG1	X04085
HSCNTFG1	X55889
HSCOL4A12	X12784

GenBank Locus	Accession Number
HSU66061	U66061
HSU78073	U78073
HSUPA	X02419
HSVATPB2R	Z37165
HSXBXVIII	X71937
HUM2OD1	D32056
HUMA1GLY2	M21540
HUMADAG	M13792
HUMADH6	M68895
HUMADRBRA	J02960
HUMAFP	M16110
HUMAGG	M11567
HUMALIFA	M63420
HUMANFA	K02043
HUMANT2X	M57424
HUMAPEB	M99703

GenBank Locus	Accession Number
HSCPH70	X52851
HSCRBP12	X07437
HSCRCANTA	Z21818
HSCST3G	X52255
HSCTPI1T5	X90780
HSCYP2D7B	X58468
HSCYP450	X02612
HSDNAAMHI	X89013
HSDNAMIA	X84707
HSDRTK123	X98208
HSEPB72E1	X85116
HSERCC25	X52221
HSERF15	X79066
HSERPG	X02158
HSEWS01	X72990
HSFBRGG	X02415
HSG13G	X98053
HSGCAP2	Z70295
HSGCKRIN1	Y09593
HSGCSFG	X03656
HSGELS3	X07065
HSGI2A1	X07854
HSGLBN	V00517
HSGLUD11	X66300
HSGPIPI1	X51501
HSGYPC	X14242
HUMAPOA4C	M14642
HUMAPOAICI	J00098
HUMAPOCIB	M20903
HUMAPOLIAA	L07899
HUMATP1A2	J05096
HUMATPSAS	D28126
HUMATPSYB	M27132
HUMBNPA	M31776
HUMCAIX	M33987
HUMCD19A	M84371
HUMCFTC	M58478
HUMCFVII	J02933
HUMCG1A1P	J02829
HUMCHYMASE	M64269
HUMCOL3A1A	M26939
HUMCRYABA	M28638
HUMCRYGBC	M19364
HUMCS3	M15894
HUMCSPA	M72150
HUMCYP7A	L13460
HUMCYPIIE	J02843
HUMDEF5A	M97925
HUMDES	M63391
HUMDHLPDH	M99384
HUMEDHB17	M27138
HUMFABP	M18079

GenBank Locus	Accession Number
HSH2B2H2	X57138
HSHAIL8G	X65858
HSHCC1GEN	Z49269
HSHEPSH	X07732
HSHIST	X57985
HSHLASBA	X03100
HSHMG17G	X13546
HSHNRNPA	X12671
HSHOX3D	X61755
HSIFD1	V00531
HSIFNAR	X60459
HSIFNG	V00536
HSIGGRE3B	Z46223
HSIL1RECA	X64532
HSIL8RB4	U11869
HSIRBPG	X53044
HSKALGENE	X82034
HSKER7E1	X13320
HSLACD691	Z30426
HSLACTG	X05153
HSLAG1G	X53682
HSLCATG	X04981
HSLPAPGEN	X97267
HSLYSOZY	X14008
HSMBP1A	X15954
HSMDCDIX1	Z25821

GenBank Locus	Accession Number
HUMFERHX	J04755
HUMFIBRB	M64983
HUMFOS	K00650
HUMG0S19B	M24110
HUMGALTB	M96264
HUMGCIA	L29478
HUMGLAA	M20317
HUMGLPEX	M83094
HUMGLUT4B	M91463
HUMGRP78	M19645
HUMGUSBA	M65002
HUMH1T	M60094
HUMHBA1	J00182
HUMHBA4	J00153
HUMHIS3PRM	M26150
HUMHIS4	M16707
HUMHKATPC	M63962
HUMHMG14A	M21339
HUMHMGIY	L17131
HUMHP2HPR	M69197
HUMHSP70D	M11717
HUMI309	M57506
HUMIFNB1F	J00218
HUMIGFBP1A	M74587
HUMIL2PR	M13879
HUMIL4A	M23442

GenBank Locus	Accession Number
HSMHCGE1	X03339
HSMHCPU15	Z14977
HSMLC1G1	X58851
HSMPOG	X15377
HSMTFDNA	X64269
HSMUC181	X68264
HSNGALGEN	X99133
HSNIDEXON	X82245
HSP45SCC	X14257
HSPARP1	X56140
HSPAT133	X69438
HSPHKBE1	X84909
HSPR264SC	X75755
HSPRB4S	X07882
HSPRKAR2A	X99455
HSPROL1	X00368
HSPROSCHY	X71874
HSPS2G1	X05030
HSPTPAX1	X86428
HSPVALB	X63578
HSRPBG1	X02775
HSRPII145	Z23102
HSRPS3AGE	X87373
HSSG1SG2	Z47556
HSSLIPG	X04502
HSSPHAR	X82554

GenBank Locus	Accession Number
HUMIL5	J03478
HUMIL9A	M86593
HUMIMPDH	L33842
HUMINSPR	M10039
HUMKER18	M24842
HUMKER2A	M21389
HUMKRT1X	M98776
HUMLCKA	M21510
HUMLUCT	D14283
HUMLYTOXBB	L11016
HUMMAG1A	M77481
HUMMDR1	M14758
HUMMETIF	M10943
HUMMETIPG	M13073
HUMMHB27B	M12967
HUMMHDC3B	K02405
HUMMIF	L19686
HUMMITCORA	L16842
HUMMLC3	J05027
HUMMSSP01	D82351
HUMMYCL2A	J03069
HUMNPATB	D83244
HUMNUCLEO	M60858
HUMP45C17	M19489
HUMPALD	M11844
HUMPBGD1	M18799

GenBank Locus	Accession Number
HSSPRO	X05006
HSSURF3	X61923
HST28B	X05929
HSTCR3G1	X06026
HSTCRT3D	X03934
HSTOP15	X52601
HSTRELFA	X73534
HSTUBB2	X02344
HSU04636	U04636
HSU08198	U08198
HSU11270	U11270
HSU16815	U16815
HSU16824	U16824
HSU19107	U19107
HSU20230	U20230
HSU21730	U21730
HSU22028	U22028
HSU25816	U25816
HSU27266	U27266
HSU30787	U30787
HSU31120	U31120
HSU31767	U31767
HSU32576	U32576
HSU33446	U33446
HSU37055	U37055
HSU37106	U37106
HUMPCBD	L41560
HUMPEM	M61170
HUMPGAMMG	J05073
HUMPGEPEB	M65126
HUMPNMTA	J03280
HUMPOLBA	J04201
HUMPP14B	M34046
HUMPP2AA	M60483
HUMPSAA	M27274
HUMRIGA	M32405
HUMRIGBCHA	M89796
HUMROD1X	M96759
HUMRPS6B	M77232
HUMSOD2TS	L34157
HUMSOMI	J00306
HUMSPERSYN	M64231
HUMSRYZ	L08063
HUMTDTB	M21195
HUMTEF1	M63896
HUMTFA01	M15673
HUMTFPB	J02846
HUMTKRA	M15205
HUMTNFAB	M16441
HUMTPI	M10036
HUMTROC	M37984
HUMTYRA	M27160

GenBank Locus	Accession Number
HSU40369	U40369
HSU41284	U41284
HSU46692	U46692
HSU48795	U48795
HSU50136	U50136
HSU51243	U51243
HSU52111	U52111
HSU52111	U52111
HSU56979	U56979
HSU63108	U63108
HSU65896	U65896

GenBank Locus	Accession Number
HUMUDPCNA	M61829
HUMVAVPO	M59834
HUMXRCC1	M36089
S52659	S52659
S53354	S53354
S58717	S58717
S68860	S68860
S70567	S70567
S72043	S72043
S74230	S74230
S79876	S79876

Appendix B

Data Set DncSet: It contains 362 human genomic DNA sequences. Each sequence does not have any coding subsequences in the 250 nt downstream region of the start site of transcription of a RNA Polymerase II transcribed gene (i.e. a 5′ UTR of at least 250 nt downstream).

GenBank Locus	Accession Number
D85375S1	D85375
HS106I20A	Z81313
HS179D3A	Z81364
HS179D3A	Z81364
HS179D3A	Z81364
HS179D3B	Z81370
HS179D3B	Z81370
HS181N1	Z82899
HS227P17	Z81007
HS326L13	Z82170
HS326L13	Z82170
HS333B10	Z81450
HS369O24	Z81008
HS369O24	Z81008
HS397C4	Z81308
HSU48393	U48393
HSU48937	U48937
HSU49869	U49869
HSU50871	U50871
HSU51899	U51899
HSU52111	U52111
HSU52111	U52111
HSU52111	U52111
HSU52111	U52111
HSU52428	U52428
HSU55847	U55847
HSU56438	U56438
HSU59831	U59831
HSU60232	U60232
HSU63833	U63833

GenBank Locus	Accession Number
HS41P2A	Z81357
HS41P2A	Z81357
HS41P2B	Z81314
HS41P2B	Z81314
HS41P2B	Z81314
HS41P2B	Z81314
HS41P2B	Z81314
HS41P2B	Z81314
HS473J10	Z81009
HS506G2A	Z82901
HS506G2A	Z82901
HS5HT1A	Z11168
HS67C13	Z80896
HSA1280	Z83307
HSA1280	Z83307
HSA1280	Z83307
HSA1280	Z83307
HSA1280	Z83307
HSACTH	V01510
HSADHVII1	U16286
HSAFGF1B	Z14150
HSAFGF1C	Z14151
HSAMPD3S06	U29929
HSANX5S01	U01681
HSAPACPP01	U07083
HSAPC3G	X03120

GenBank Locus	Accession Number
HSU66082	U66082
HSU66083	U66083
HSU66083	U66083
HSU66083	U66083
HSU77732	U77732
HSUBA52G	X56997
HSUNGG	X79093
HSUPA	X02419
HSV1RG1	U11079
HSV602D8	Z83131
HSVGLY	X64281
HSVMYCLC2	Z15030
HSVWF123	X06828
HSWRSX1B	X67919
HSWTWIT1	X77549
HSX11G	Z32676
HSXLRPGN1	X94766
HSXLRPGN3	X94768
HUB384D8	U62317
HUB384D8	U62317
HUB384D8	U62317
HUM2OGDH	D10523
HUM4F2HG1	M21898
HUMACTGA	M19283
HUMAE1ERY	L35930
HUMAGAL	M59199

GenBank Locus	Accession Number
HSAQP501	U46566
HSASG5E	X03258
HSATPCP1	X69907
HSATPCP2	X69908
HSB11B7	Z82171
HSBAX1	U17193
HSBDONE1	U50930
HSBETACAT	X89448
HSBM40DNA	X82259
HSBNGF	V01511
HSBTKS1	U10084
HSC1INHIB	X54486
HSC45B2B	X06399
HSCAMHCA	Z20656
HSCHAT	X56585
HSCKBG	X15334
HSCKIIBE	X57152
HSCKPSEU	X64692
HSCOLLX	X98568
HSCYP45C	X04300
HSDAO	X78212
HSDARC	X85785
HSDBH1	X13257
HSDNAHCGV	X89902
HSDNATEF1	X84839
HSDYSE51	X51934

GenBank Locus	Accession Number
HUMAK1	J04809
HUMALDB1	M15657
HUMAMPD1X	M98818
HUMAPEXN	D13370
HUMAPOAICI	J00098
HUMARAF1G	L24038
HUMARPR	M58158
HUMATP1B2A	L23414
HUMATPAC1	M30309
HUMBCRE	L02935
HUMBMYH7	M57965
HUMBN51PRO	L15301
HUMCACY	J02763
HUMCAIII1	M29452
HUMCD40L1	D31793
HUMCD43	M61827
HUMCELL	M94580
HUMCETP1	M32992
HUMCFCGRI1	M63830
HUMCHRAS1	M17220
HUMCKMM1	M21487
HUMCOL5A1A	L38808
HUMCR1SF01	L17390
HUMCYAR01	M30795
HUMCYAR02	M30796
HUMCYPB1	M32863

GenBank Locus	Accession Number
HSE132D12	Z80897
HSE132D12	Z80897
HSE132D12	Z80897
HSE132D12	Z80897
HSE132D12	Z80897
HSE132D12	Z80897
HSE132D12	Z80897
HSE92H8	Z81309
HSECP1	X16545
HSENKB1	X02536
HSENO2	X51956
HSENO35	X55976
HSESTEI1	X62259
HSF67D6	Z81000
HSF77D12	Z82097
HSFAU1	X65921
HSFOLA	X69516
HSFURIN	X15723
HSG11ASKI	X99296
HSG6PDGEN	X55448
HSG6PDGEN	X55448
HSGAA1	X55079
HSGAPIGNA	X74322
HSGDF5	X80915
HSGI3APR	X54048
HSHB9HB1	U07663

GenBank Locus	Accession Number
HUMCYPX1	M31664
HUMDECOR02	M98262
HUMDMDPR	M32058
HUMEF1A	J04617
HUMELAMB	M64485
HUMEMBPA	M34462
HUMESTTCT	L35592
HUMFAS	D31968
HUMFIXG1	K02048
HUMFKBPA01	M92422
HUMFOL1	K01612
HUMG0S2PE	M72885
HUMGAD45A	L24498
HUMGAPDHG	J04038
HUMGASTA	M15958
HUMGFAPJ	M67446
HUMGLNIN	L11144
HUMGPIIB1	M33319
HUMGPXP1	D16360
HUMGRPRA	M32284
HUMGSTPIA	M37065
HUMHBGF1	M23017
HUMHCF2	M58600
HUMHER201	M16789
HUMHIAPPA	M26650
HUMHMG2A	M83665

GenBank Locus	Accession Number
HSHCGIX23	X95288
HSHH3X3B	Z48950
HSHMGICP	X92518
HSHNMT01	U44106
HSHOX51	X17360
HSHSD11K1	U27317
HSIGGFCII	X68090
HSIGNT1	L41605
HSIKLVA13	X63395
HSIKLVA26	X63399
HSIL1AG	X03833
HSL165D7	Z68273
HSL247F6	Z68279
HSL25A3	Z68280
HSL3G9A	Z69387
HSLAMB2I	Z68155
HSLDHB1	X13794
HSLG9B	Z69711
HSMECDAG	X62654
HSMEHG	X15459
HSMNCA9	Z54349
HSMOGG	Z48051
HSMPR461	X56253
HSMYC2	X00247
HSN119A7	Z80901
HSN119A7	Z80901

GenBank Locus	Accession Number
HUMHMGCOB	M15959
HUMHODB1	M28162
HUMHSP89KD	M27024
HUMHSP90B	J04988
HUMIDO	M58159
HUMIGFBP5X	L27560
HUMIGFR1PR	M69229
HUMIGHVU	J04097
HUMIL11A	M81890
HUMIL1B	M15840
HUMIL2RBA	M32979
HUMIL2RC	M16285
HUMIL2RG01	L12178
HUMIL8A	M28130
HUMIL8R	M99412
HUMKIN01	M11438
HUMKIP2	D64137
HUMLBPB	L42172
HUMLBR01	L25932
HUMLFACD	M87662
HUMLYAM1	M32406
HUMLYRE	M30447
HUMMGI1	M10090
HUMMHDQ1A	M33765
HUMMRP14A	M21064
HUMMRP8A	M21005

GenBank Locus	Accession Number
HSN53A9	Z81002
HSN80H12	Z80902
HSNCAM5R	X53243
HSNEURK1	X65172
HSNFKB2PR	X83768
HSNNMT1	U20970
HSNPTX2A1	U29191
HSNRAMP2	X82016
HSNTRK1	X71445
HSODCG	X16277
HSP3	X12458
HSP53G	X54156
HSP55TNF	X69810
HSPABPS01	U68093
HSPCRF	V00571
HSPIT11	X77223
HSPMLPROM	X91752
HSPPOXG	X99450
HSPRELP01	U41292
HSPROPG	X70872
HSPTHRP5	X14304
HSPTHRP5	X14304
HSRARG2D	X57280
HSREOR	X95536
HSRNPAII1	U09120
HSROSSA1	U13657
HUMMTH11	D38591
HUMMTH12	D38592
HUMMYCBLK	M12027
HUMMYCLYA	M13211
HUMNAKATP1	M25161
HUMNITOX01	L10693
HUMOP18A	M31303
HUMPAI2AA1	M22469
HUMPAIA	J03764
HUMPAP	L15533
HUMPCI	M68516
HUMPCNAPRM	J05614
HUMPDGFAA	M58602
HUMPDK01	U54618
HUMPECAM01	L34631
HUMPGK1	L00159
HUMPOLYUBI	D63791
HUMPP2AB	M60484
HUMPPARG	D83233
HUMPPPA	M11726
HUMPRECX	L13994
HUMPRF1A	M31951
HUMPROFILX	M96943
HUMPSAPA	M86181
HUMPYYP1	D13897
HUMRAF1PR	M38134

GenBank Locus	Accession Number
HSRPS7	Z25749
HSSAA1B	X56652
HSSB4B46	X03028
HSSP17	Z48570
HSSUBP1G	X65177
HSTAX1EX1	X84419
HSTFAP2GN	X95235
HSTGF31	X14885
HSTGFBG1	X05839
HSTHDA	X70286
HSTHR2	V00596
HSTNNTX9	X98481
HSTRAP	X67123
HSTRE17	X63596
HSTRP	X05339
HSU01965	U01965
HSU11239	U11239
HSU11870	U11870
HSU15963	U15963
HSU17084	U17084
HSU173H7	Z80774
HSU173H7	Z80774
HSU173H7	Z80774
HSU18671	U18671
HSU20499	U20499
HUMREGIBB	D17291
HUMRIBRED	L10342
HUMSAACT	M20543
HUMSANT	M38064
HUMSRD5A2X	L03843
HUMSTATH1	M31077
HUMTBXAS01	D34613
HUMTGFB3B	M60556
HUMTGL1	M29186
HUMTHROMPR	J04835
HUMTNFAB	M16441
HUMTPAA1	M11888
HUMTPO01	M25701
HUMTRP2AA	D28767
HUMTSHBA1	M23981
HUMTSPY	M98524
HUMVWFAB	M60676
HUMWEGAUTO	M97911
PLCB3X01	Z37544
S54531	S54531
S55222	S55222
S67998	S67998
S68043	S68043
S68887	S68887
S70157	S70157

GenBank Locus	Accession Number
HSU20860	U20860
HSU20982	U20982
HSU31519	U31519
HSU33208	U33208
HSU34301	U34301
HSU34804	U34804
HSU34879	U34879
HSU37574	U37574
HSU40391	U40391
HSU41315	U41315
HSU43901	U43901
S74903	S74903
S75590	S75590
S75654	S75654
S75955	S75955
S77920	S77920
S78723	S78723
S79432	S79432
S79812	S79812
S80050	S80050
S81868	S81868
U00682	U00682

Bibliography

[1] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular Biology of the Cell.* New York, NY: Garland Publishing Inc., 1994.

[2] M. Amitar. Hidden Models in Biopolymers. *Science*, 282:1436–1437, 1998.

[3] F. Antequera and A. Bird. Number of CpG Islands and Genes in Human and Mouse. *Proceedings of the National Academy of Sciences of the United States of America*, 90:11995–11999, 1993.

[4] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach.* Cambridge, MA: MIT Press, 1998.

[5] P. Baldi, S. Brunak, Y. Chauvin, J. Engelbrecht, and A. Krogh. Hidden Markov Models for Human Genes. *Advances in Neural Information Processing Systems*, 6:761–768, 1994.

[6] P. Baldi, S. Brunak, Y. Chauvin, and A. G. Pedersen. Structural Basis for Triplet Repeat Disorders: A Computational Analysis. *Bioinformatics*, 15:918–929, 1999.

[7] P. Baldi and Y. Chauvin. Hybrid Modeling, HMM/NN Architectures, and Protein Applications. *Neural Computation*, 8:1541–1565, 1996.

[8] P. Baldi, Y. Chauvin, S. Brunak, and A. G. Gorodkin, J. Pedersen. Computational Applications of DNA Structural Scales. In *Proceedings of the 6th International Conference on Intelligent Systems for Molecular Biology*, pages 35–42. Menlo Park, CA: AAAI Press, 1998.

[9] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure. Hidden Markov Models of Biological Primary Sequence Information. *Proceedings of the National Academy of Sciences of the United States of America*, 91:1059–1063, 1994.

[10] D. Balding, M. Bishop, and C. Cannings, editors. *Handbook of Statistical Genetics.* New York, NY: John Wiley and Sons Inc., 2001.

[11] A. D. Bates and A. Maxwell. *DNA Topology.* Oxford, UK: Oxford University Press, 1993.

[12] L. E. Baum and T. Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966.

[13] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[14] C. J. Benham. Theory of DNA Superhelicity. In *Mathematics in Biology and Medicine (Bari, 1983). Lecture Notes in Biomathematics, 57*, pages 253–259. New York, NY: Springer-Verlag Inc., 1985.

[15] C. J. Benham. Mechanics and Equilibria of Superhelical DNA. In *M. Waterman, editors. Mathematical Methods for DNA Sequences*, pages 255–278. Boca Raton, FL: CRC Press Inc., 1989.

[16] C. J. Benham. Computation of DNA Structural Variability — A New Predictor of DNA Regulatory Regions. *Computer Applications in the Biosciences*, 12(5):375–381, 1996.

[17] C. J. Benham. The Topologically Driven Strand Separation Transition in DNA — Methods of Analysis and Biological Significance. In *M. Farach-Colton, F. S. Roberts, M. Vingron and M. Waterman, editors. Mathematical Support for Molecular Biology*, pages 173–198. Providence, RI: American Mathematical Society, 1999.

[18] G. Bernardi. The Isochore Organization of the Human Genome and Its Evolutionary History — A Review. *Gene*, 135:57–66, 1993.

[19] G. Bernardi. The Human Genome: Organization and Evolutionary History. *Annual Review of Genetics*, 29:445–476, 1995.

[20] P. J. Bickel and Y. Ritov. Inference in Hidden Markov Models I: Local Asymptotic Normality in the Stationary Case. *Bernoulli*, 2:199–228, 1996.

[21] P. J. Bickel, Y. Ritov, and T. Rydén. Asymptotic Normality of the Maximum Likelihood Estimator for General Hidden Markov Models. *Annals of Statistics*, 26:1614–1635, 1998.

[22] M. J. Bishop and E. A. Thompson. Maximum Likelihood Alignment of DNA Sequences. *Journal of Molecular Biology*, 190:159–165, 1986.

[23] H. Branswell. 'Book of Life' is Written: World Lauds Monumental Genetic Map of Humanity. *Winnipeg Free Press*, page B1, 2000.

[24] V. Brendel, J. S. Beckmann, and E. N. Trifonov. Linguistics of Nucleotide Sequences: Morphology and Comparison of Vocabularies. *Journal of Biomolecular Structure and Dynamics*, 4:11–20, 1986.

[25] I. Brukner, R. Sánchez, D. Suck, and S. Pongor. Sequence-Dependent Bending Propensity of DNA as Revealed By DNase I: Parameters for Trinucleotides. *European Molecular Biology Organization Journal*, 14(8):1812–1818, 1995.

[26] I. Brukner, R. Sánchez, D. Suck, and S. Pongor. Trinucleotide Models for DNA Bending Propensity: Comparison of Models Based on DNase I Digestion and Nucleosome Packaging Data. *Journal of Biomolecular Structure and Dynamics*, 13(2):309–317, 1995.

[27] C. Burge and S. Karlin. Prediction of Complete Gene Structures in Human Genomic DNA. *Journal of Molecular Biology*, 268:78–94, 1997.

[28] C. Burge and S. Karlin. Finding the Genes in Genomic DNA. *Current Opinion in Structural Biology*, 8:346–354, 1998.

[29] B. P. Carlin and T. A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis.* London, UK: Chapman and Hall, 1996.

[30] G. Casella and E. I. George. Explaining the Gibbs Sampler. *American Statistician*, 46(3):167–174, 1992.

[31] G. A. Churchill. Stochastic Models for Heterogeneous DNA Sequences. *Bulletin of Mathematical Biology*, 51(1):79–94, 1989.

[32] G. A. Churchill. Hidden Markov Chains and the Analysis of Genome Structure. *Computers and Chemistry*, 16(2):107–115, 1992.

[33] G. A. Churchill. Accurate Restoration of DNA Sequences (With Discussion). In *C. Gatsaris, J. S. Hodges, R. E. Kass and N. D. Singpurwalla, editors. Case Studies in Bayesian Statistics*, pages 90–148. New York, NY: Springer-Verlag Inc., 1995.

[34] G. A. Churchill. Hidden Markov Models. In *Encyclopedia of Biostatistics*, pages 1908–1916. New York, NY: John Wiley and Sons Inc., 1998.

[35] G. A. Churchill and B. Lazareva. Bayesian Restoration of a Hidden Markov Chain with Applications to DNA Sequencing. *Journal of Computational Biology*, 6(2):261–277, 1999.

[36] F. S. Collins, A. Patrinos, E. Jordan, A. Chakravarti, R. Gesteland, and L. Walters. New Goals for the U.S. Human Genome Project: 1998-2003. *Science*, 282(5389):682–689, 1998.

[37] S. H. Cross and A. Bird. CpG Islands and Genes. *Current Opinion in Genetics and Development*, 5:309–314, 1995.

[38] E. M. Crowley, K. Roeder, and M. Bina. A Statistical Model for Locating Regulatory Regions in Genomic DNA. *Journal of Molecular Biology*, 268:8–14, 1997.

[39] F. N. David. A Power Function for Tests of Randomness in a Sequence of Alternatives. *Biometrika*, 34:335–339, 1947.

[40] J. R. Davie. Regulatory Elements and Initiation Factors of RNA Polymerase II Transcribed Genes. *(Biochemistry and Molecular Biology Course 82.725 — Gene Expression — Lecture Material)*, 1999.

[41] C. DeLisi. The Human Genome Project. *American Scientist*, 76:488–493, 1988.

[42] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm (With Discussion). *Journal of the Royal Statistical Society — Series B*, 39(1):1–38, 1977.

[43] R. E. Dickerson. DNA Structure from A to Z. *Methods in Enzymology*, 211:67–111, 1992.

[44] R. Dulbecco. A Turning Point in Cancer Research: Sequencing the Human Genome. *Science*, 231:1055–1056, 1986.

[45] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge, UK: Cambridge University Press, 1998.

[46] S. R. Eddy. Multiple Alignment Using Hidden Markov Models. In *Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology*, pages 114–120. Menlo Park, CA: AAAI Press, 1995.

[47] S. R. Eddy. Hidden Markov Models. *Current Opinion in Structural Biology*, 6(3):361–365, 1996.

[48] S. R. Eddy. Hidden Markov Models and Large-Scale Genome Analysis. *(Preprint) Transactions of the American Crystallographic Association. (http://www.genetics.wustl.edu/eddy/publications/)*, 1997.

[49] S. R. Eddy. Profile Hidden Markov Models. *Bioinformatics*, 14(9):755–763, 1998.

[50] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap.* London, UK: Chapman and Hall, 1993.

[51] W. H. Elliott and D. C. Elliott. *Biochemistry and Molecular Biology.* Oxford, UK: Oxford University Press, 1997.

[52] J. Felsenstein and G. A. Churchill. A Hidden Markov Model Approach to Variation Among Sites in Rate of Evolution. *Molecular Biology and Evolution*, 13(1):93–104, 1996.

[53] J. W. Fickett and A. G. Hatzigeorgiou. Eukaryotic Promoter Recognition. *Genome Research*, 7:861–878, 1997.

[54] M. D. Frank-Kamenetŝkii. *Unraveling DNA: The Most Important Molecule of Life.* Reading, MA: Addison-Wesley, 1997.

[55] J. C. Fu. Reliability of Consecutive-k-out-of-n : F System with $k-1$ Step Markov Dependence. *IEEE Transactions on Reliability*, R35:602–606, 1986.

[56] J. C. Fu. Distribution Theory of Runs and Patterns Associated with a Sequence of Multi-State Trials. *Statistica Sinica*, 6(4):957–974, 1996.

[57] J. C. Fu and S.-C. Chen. Pattern Matching of Non-Aligned DNA Sequences. *(Statistics Seminar Course 05.723 — Lecture Material)*, 1996.

[58] J. C. Fu and M. V. Koutras. Distribution Theory of Runs: A Markov Chain Approach. *Journal of the American Statistical Association*, 89(427):1050–1058, 1994.

[59] J. C. Fu, W. Y. W. Lou, and S.-C. Chen. On the Probability of Pattern Matching in Nonaligned DNA Sequences: A Finite Markov Chain Imbedding Approach. In *J. Glaz and N. Balakrishnan, editors. Scan Statistics and Applications*, pages 287–302. New York, NY: Birkhäuser Boston Inc., 1999.

[60] W. Fuller, M. H. F. Wilkins, H. R. Wilson, L. D. Hamilton, and S. Arnott. The Molecular Configuration of Deoxyribonucleic Acid. IV. X-Ray Diffraction Study the A Form. *Journal of Molecular Biology*, 12:60–80, 1965.

[61] A. P. Godbole. Specific Formulae for Some Success Run Distributions. *Statistics and Probability Letters*, 10:119–124, 1990.

[62] D. S. Goodsell and R. E. Dickerson. Bending and Curvature Calculations in B-DNA. *Nucleic Acids Research*, 22(24):5497–5503, 1994.

[63] A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, R. C. Lewontin, and W. M. Gelbart. *An Introduction to Genetic Analysis.* New York, NY: W. H. Freeman and Company, 2000.

[64] A. G. Hatzigeorgiou, N. Mache, and M. Reczko. Functional Site Prediction on the DNA Sequence by Artificial Neural Networks. In *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*, pages 12–17. Los Alamitos, CA: IEEE Computer Society Press, 1996.

[65] J. Henderson, S. Salzberg, and K. H. Fasman. Finding Genes in DNA with a Hidden Markov Model. *Journal of Computational Biology*, 4(2):127–141, 1997.

[66] K. Hirano. Some properties of the distributions of order k. In *A. N. Philippou, G. E. Bergum, and A. F. Horadam, editors. Fibonacci Numbers and Their Applications*, pages 43–53. Dordrecht: Reidel, 1986.

[67] K. Hirano and S. Aki. On the Number of Occurrences of Success Runs of Length k in a Two-State Markov Chain. *Statistica Sinica*, 3:313–320, 1993.

[68] R. Hughey and A. Krogh. Hidden Markov Models for Sequence Analysis: Extension and Analysis of the Basic Method. *Computer Applications in the Biosciences*, 12:95–107, 1996.

[69] C. A. Hunter. Sequence-Dependent DNA Structure: The Role of Base Stacking Interactions. *Journal of Molecular Biology*, 230:1025–1054, 1993.

[70] C. A. Hunter. Sequence-Dependent DNA Structure. *BioEssays*, 18(2):157–162, 1996.

[71] M. Jamshidian and R. I. Jennrich. Acceleration of EM Algorithm by Using Quasi-Newton Methods. *Journal of the Royal Statistical Society — Series B*, 59:569–587, 1997.

[72] H. Karas, R. Knüppel, W. Schulz, H. Sklenar, and E. Wingender. Combining Structural Analysis of DNA with Search Routines for the Detection of Transcription Regulatory Elements. *Computer Applications in the Biosciences*, 12(5):441–446, 1996.

[73] S. Karlin. Statistical Approaches in Assessing Structural Relationships in DNA Sequences. In *Proceedings of the International Biometric Conference*, volume 12, pages 298–? Biometric Society (Washington), 1984.

[74] S. Karlin, B. E. Blaisdell, R. J. Sapolsky, L. Cardon, and C. Burge. Assessments of DNA Inhomogeneities in Yeast Chromosome III. *Nucleic Acids Research*, 21:703–711, 1993.

[75] S. Karlin and V. Brendel. Chance and Statistical Significance in Protein and DNA Sequence Analysis. *Science*, 257:39–49, 1992.

[76] S. Karlin and G. Ghandour. DNA Sequence Patterns in Human, Mouse, and Rabbit Immunoglobulin Kappa-Genes. *Journal of Molecular Evolution*, 21:195–208, 1985.

[77] S. Karlin, G. Ghandour, F. Ost, S. Tavare, and L. J. Korn. New Approaches for Computer Analysis of Nucleic Acid Sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 80:5660–5664, 1983.

[78] S. Karlin, I. Ladunga, and B. E. Blaisdell. Heterogeneity of Genomes: Measures and Values. *Proceedings of the National Academy of Sciences of the United States of America*, 91:12837–12841, 1994.

[79] S. Karlin, J. Mrázek, and A. M. Campbell. Frequent Oligonucleotides and Peptides of the Haemophilus Influenzae Genome. *Nucleic Acids Research*, 24:4263–4272, 1996.

[80] S. Karlin, J. Mrázek, and A. M. Campbell. Compositional Biases of Bacterial Genomes and Evolutionary Implications. *Journal of Bacteriology*, 179:3899–3913, 1997.

[81] S. Karlin and F. Ost. Maximal Segmental Match Length Among Random Sequences From a Finite Alphabet. In *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer (Volume 1)*, pages 225–243. Belmont, CA: Wadsworth, 1985.

[82] A. Krogh. Two Methods for Improving Performance of an HMM and Their application for Gene Finding. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 179–186. Menlo Park, CA: AAAI Press, 1997.

[83] A. Krogh. An Introduction to Hidden Markov Models for Biological Sequences. In *S. L. Salzberg, D. B. Searls and S. Kasif, editors. Computational Methods in Molecular Biology*, pages 45–63. Amsterdam: Elsevier Science B. V., 1998.

[84] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov Models in Computational Biology: Applications to Protein Modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[85] A. Krogh, I. S. Mian, and D. Haussler. A Hidden Markov Model that Finds Genes in E. coli DNA. *Nucleic Acids Research*, 22(22):4768–4778, 1994.

[86] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A Generalized Hidden Markov Model for the Recognition of Human Genes in DNA. In *Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology*, pages 134–142. Menlo Park, CA: AAAI Press, 1996.

[87] R. Langridge, D. A. Marvin, W. E. Seeds, H. R. Wilson, and L. D. Hamilton. The Molecular Configuration of Deoxyribonucleic Acid. II. Molecular Models and Their Fourier Transforms. *Journal of Molecular Biology*, 2:38–64, 1960.

[88] R. Langridge, H. R. Wilson, C. W. Hooper, M. H. F. Wilkins, and L. D. Hamilton. The Molecular Configuration of Deoxyribonucleic Acid. I. X-Ray Diffraction Study of a Crystalline Form of the Lithium Salt. *Journal of Molecular Biology*, 2:19–37, 1960.

[89] B. G. Leroux. Maximum-Likelihood Estimation for Hidden Markov Models. *Stochastic Processes and their Applications*, 40:127–143, 1992.

[90] B. G. Leroux and M. L. Puterman. Maximum-Penalized-Likelihood Estimation for Independent and Markov-Dependent Mixture Models. *Biometrics*, 48:545–558, 1992.

[91] M.-Y. Leung, G. M. Marsh, and T. P. Speed. Over- and Under-Representation of Short DNA Words in Herpesvirus Genomes. *Journal of Computational Biology*, 3:345–360, 1996.

[92] M.-Y. Leung, G. A. Schachtel, and H. S. Yu. Scan Statistics and DNA Sequence Analysis: The Search for an Origin of Replication in a Virus. *Nonlinear World*, 1:445–471, 1994.

[93] M.-Y. Leung and T. E. Yamashita. Applications of the Scan Statistic in DNA Sequence Analysis. In *J. Glaz and N. Balakrishnan, editors. Scan Statistics and Applications*, pages 269–286. New York, NY: Birkhäuser Boston Inc., 1999.

[94] B. Lewin. *Genes VI.* Oxford, UK: Oxford University Press, 1997.

[95] C. Liu and D. B. Rubin. The ECME Algorithm: A Simple Extension of EM and ECM with Fast Monotone Convergence. *Biometrika*, 81:633–648, 1994.

[96] C. Liu, D. B. Rubin, and Y. N. Wu. Parameter Expansion to Accelerate EM — The PX-EM Algorithm. *Biometrika*, 85:755–770, 1998.

[97] W. Y. W. Lou. On Runs and Longest Run Tests: A Method of Finite Markov Chain Imbedding. *Journal of the American Statistical Association*, 91(436):1595–1601, 1996.

[98] W. Y. W. Lou. An Application of the Method of Finite Markov Chain Imbedding to Runs Tests. *Statistics and Probability Letters*, 31:155–161, 1997.

[99] W. Y. W. Lou. The Finite Markov Chain Imbedding Approach to the K-tuple Statistic for DNA Tandem Repeats. *(Manuscript)*, 2000.

[100] E. L. Marshall. *The Human Genome Project: Cracking the Code within Us.* New York, NY: Franklin Watts Inc., 1996.

[101] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* New York, NY: John Wiley and Sons Inc., 1997.

[102] X.-L. Meng and D. van Dyk. The EM Algorithm — An Old Folk-Song Sung to a New Fast Tune (with Discussion). *Journal of the Royal Statistical Society — Series B*, 59:551–567, 1997.

[103] A. M. Mood. The Distribution Theory of Runs. *Annals of Mathematical Statistics*, 11:367–392, 1940.

[104] F. Mosteller. Note on an Application of Runs to Quality Control Charts. *Annals of Mathematical Statistics*, 12:228–232, 1941.

[105] L. C. Murphy. DNA Transcription and RNA Processing. *(Interdepartmental Course 36.724 — Nucleic Acids: Manipulation, Structure and Function — Lecture Material)*, 1998.

[106] L. C. Murphy. Recombinant DNA Technology I. *(Interdepartmental Course 36.724 — Nucleic Acids: Manipulation, Structure and Function — Lecture Material)*, 1998.

[107] S. Neidle. *DNA Structure and Recognition.* Oxford, UK: Oxford University Press, 1994.

[108] A. G. Pedersen, P. Baldi, S. Brunak, and Y. Chauvin. Characterization of Prokaryotic and Eukaryotic Promoters Using Hidden Markov Models. In *Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology*, pages 182–191. Menlo Park, CA: AAAI Press, 1996.

[109] A. G. Pedersen, P. Baldi, Y. Chauvin, and S. Brunak. DNA Structure in Human RNA Polymerase II Promoters. *Journal of Molecular Biology*, 281(4):663–673, 1998.

[110] A. G. Pedersen, P. Baldi, Y. Chauvin, and S. Brunak. The Biology of Eukaryotic Promoter Prediction — A Review. *Computers and Chemistry, special issue on "Genome and Informatics"*, 23(3–4):191–207, 1999.

[111] A. G. Pedersen and J. Engelbrecht. Investigations of *Escherichia coli* Promoter Sequences with Artificial Neural Networks: New Signals Discovered Upstream of the Transcriptional Startpoint. In *Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology*, pages 292–299. Menlo Park, CA: AAAI Press, 1995.

[112] A. G. Pedersen, L. L. Jensen, S. Brunak, and Stærfeldt. A DNA Structural Atlas for *Escherichia coli. Journal of Molecular Biology*, 299:907–930, 2000.

[113] A. G. Pedersen and H. Nielsen. Neural Network Prediction of Translation Initiation Sites in Eukaryotes: Perspectives for EST and Genome Analysis. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 226–233. Menlo Park, CA: AAAI Press, 1997.

[114] A. N. Philippou and F. S. Makri. Successes, Runs, and Longest Runs. *Statistics and Probability Letters*, 4:101–105, 1986.

[115] S. R. Presnell and F. E. Cohen. Artificial Neural Networks for Pattern Recognition in Biochemical Sequences. *Annual Review of Biophysics and Biomolecular Structure*, 22:283–298, 1993.

[116] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[117] C. J. Rawlings and J. P. Fox. Artificial Intelligence in Molecular Biology: A Review and Assessment. *Philosophical Transactions of the Royal Society of London — Series B*, 344:353–363, 1994.

[118] M. G. Reese, F. H. Eeckman, D. Kulp, and D. Haussler. Improved Splice Site Detection in Genie. *Journal of Computational Biology*, 4(3):311–323, 1997.

[119] S. K. Riis and A. Krogh. Hidden Neural Networks: A Framework for HMM/NN Hybrids. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3233–3236, 1997.

[120] S. C. Schmidler, J. S. Liu, and D. L. Brutlag. Bayesian Segmentation of Protein Secondary Structure. *Journal of Computational Biology*, 7(1/2):233–248, 2000.

[121] G. D. Schuler, M. S. Boguski, E. A. Stewart, L. D. Stein, G. Gyapay, K. Rice, R. E. White, P. Rodriguez-Tomé, A. Aggarwal, E. Bajorek, S. Bentolila, B. B. Birren, A. Butler, A. B. Castle, N. Chiannilkulchai, A. Chu, C. Clee, S. Cowles, P. J. R. Day, T. Dibling, N. Drouot, I. Dunham, S. Duprat, C. East, C. Edwards, J.-B. Fan, N. Fang, C. Fizames, C. Garrett,

L. Green, D. Hadley, M. Harris, P. Harrison, S. Brady, A. Hicks, E. Holloway, L. Hui, S. Hussain, C. Louis-Dit-Sully, J. Ma, A. MacGilvery, C. Mader, T. C. Maratukulam, A. Matise, K. B. McKusick, J. Morissette, A. Mungall, D. Muselet, H. C. Nusbaum, D. C. Page, A. Peck, S. Perkins, M. Piercy, F. Qin, J. Quackenbush, S. Ranby, T. Reif, S. Rozen, C. Sanders, X. She, J. Silva, D. K. Slonim, C. Soderlund, W.-L. Sun, P. Tabar, T. Thangarajah, N. Vega-Czarny, D. Vollrath, S. Voyticky, T. Wilmer, X. Wu, M. D. Adams, C. Auffray, N. A. R. Walter, R. Brandon, A. Dehejia, P. N. Goodfellow, R. Houlgatte, J. R. Hudson Jr., S. E. Ide, K. R. Iorio, W. Y. Lee, N. Seki, T. Nagase, K. Ishikawa, N. Nomura, C. Phillips, M. H. Polymeropoulos, M. Sandusky, K. Schmitt, R. Berry, K. Swanson, R. Torres, j. C. Venter, J. M. Sikela, J. S. Beckmann, J. Weissenbach, R. M. Myers, D. R. Cox, M. R. James, D. Bentley, P. Deloukas, E. S. Lander, and T. J. Hudson. A Gene Map of the Human Genome. *Science*, 274:540–546, 1996.

[122] D. B. Searls. Representing Genetic Information with Formal Grammars. In *Proceedings of the 7th National Conference on Artif ical Intelligence Applications*, pages 3–9. Los Alamitos, CA: IEEE Computer Society Press, 1988.

[123] R. R. Sinden. *DNA Structure and Function.* San Diego, CA: Academic Press Inc., 1994.

[124] R. R. Sinden, C. E. Pearson, V. N. Potaman, and D. W. Ussery. DNA: Structure and Function. In *R. S. Verma, editor. Advances in Genome Biology, Volume 5A*, pages 1–141. JAI Press Inc., 1998.

[125] A. D. Smith, S. P. Datta, G. H. Smith, P. N. Campbell, R. Bentley, and H. A. McKenzie, editors. *Oxford Dictionary of Biochemistry and Molecular Biology.* Oxford, UK: Oxford University Press, 1997.

[126] E. E. Snyder and G. D. Stormo. Identification of Coding Regions in Genomic DNA Sequences: An Application of Dynamic Programming and Neural Networks. *Nucleic Acids Research*, 21(3):607–613, 1993.

[127] W. L. Stevens. Distributions of Groups in a Sequence of Alternatives. *Annals of Eugenics*, 9:10–17, 1939.

[128] E. A. Thompson. What is Statistical Genetics in the New Millenium? (Internet Notice). *URL http://www.stat.washington.edu/thompson/Statgen/Whatis/Whatis.shtml*, 1999.

[129] J. L. Thorne and G. A. Churchill. Estimation and Reliability of Molecular Sequence Alignments. *Biometrics*, 51:100–113, 1995.

[130] A. Tucker. *Applied Combinatorics.* New York, NY: John Wiley and Sons Inc., 1995.

[131] R. M. Twyman. *Advanced Molecular Biology: A Concise Reference.* Oxford, UK: BIOS Scientific Publishers, 1998.

[132] M. Vingron. Bioinformatics Needs to Adopt Statistical Thinking. *Bioinformatics*, 17(5):389–390, 2001.

[133] A. J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, 1967.

[134] A. Wald and J. Wolfowitz. On a Test Whether Two Samples are From the Same Population. *Annals of Mathematical Statistics*, 11:147–162, 1940.

[135] M. S. Waterman. Secondary Structure of Single-Stranded Nucleic Acids. In *G.-C. Rota, editors. Studies in Foundations and Combinatorics. Advances in Mathematics, Supplementary Studies, 1*, pages 167–212. San Diego, CA: Academic Press Inc., 1978.

[136] M. S. Waterman. Combinatorics of RNA Hairpins and Cloverleaves. *Studies in Applied Mathematics*, 60:91–96, 1979.

[137] M. S. Waterman. General Methods of Sequence Comparison. *Bulletin of Mathematical Biology*, 46:473–500, 1984.

[138] M. S. Waterman. How Do You Spell DNA? *Nature*, 309:118, 1984.

[139] M. S. Waterman. Multiple Sequence Alignment by Consensus. *Nucleic Acids Research*, 14:9095–9102, 1986.

[140] M. S. Waterman. Probability Distributions for DNA Sequence Comparisons. In *Some Mathematical Questions in Biology—DNA Sequence Analysis (New York, 1984). Lectures on Mathematics in the Life Sciences, 17*, pages 29–56. Providence, RI: American Mathematical Society, 1986.

[141] M. S. Waterman. Computer Analysis of Nucleic Acid Sequences. *Methods in Enzymology*, 164:765–793, 1988.

[142] M. S. Waterman, editor. *Mathematical Methods for DNA Sequences.* Boca Raton, FL: CRC Press Inc., 1989.

[143] M. S. Waterman. Applications of Combinatorics to Molecular Biology. In *R. L. Graham, M. Grötschel and L. Lovász, editors. Handbook of Combinatorics. Vol. 1, 2*, pages 1983–2001. Cambridge, MA: MIT Press, 1995.

[144] M. S. Waterman. *Introduction to Computational Biology: Maps, Sequences and Genomes.* London, UK: Chapman and Hall, 1995.

[145] M. S. Waterman, R. Arratia, and D. J. Galas. Pattern Recognition in Several Sequences: Consensus and Alignment. *Bulletin of Mathematical Biology*, 46:515–527, 1984.

[146] M. S. Waterman and T. F. Smith. Rapid Dynamic Programming Methods for RNA Secondary Structure. *Advances in Applied Mathematics*, 7:455–464, 1986.

[147] M. S. Waterman, T. F. Smith, and W. A. Beyer. Some Biological Sequence Metrics. *Advances in Mathematics*, 20(3):367–387, 1976.

[148] B. S. Weir. Statistical Analysis of Molecular Genetic Data. *IMA Journal of Mathematics Applied in Medicine and Biology*, 2:1–39, 1985.

[149] B. S. Weir. Statistical Analysis of DNA Sequences. *Journal of the National Cancer Institute*, 80(6):395–406, 1988.

[150] A. P. Wolffe. *Chromatin — Structure and Function.* San Diego, CA: Academic Press Inc., 1998.

[151] A. P. Wolffe and H. R. Drew. DNA Structure: Implications for Chromatin Structure and Function. In *S. C. R. Elgin, editor. Chromatin Structure and Gene Expression*, pages 27–48. Oxford, UK: IRL Press, 1995.

[152] J. Wolfowitz. On the Theory of Runs With Some Applications to Quality Control. *Annals of Mathematical Statistics*, 14:280–288, 1943.

[153] W. H. Wong. Computational Molecular Biology. *Journal of the American Statistical Association*, 95(449):322–326, 2000.

[154] C. Wu. Artificial Neural Networks for Molecular Sequence Analysis. *Computers and Chemistry*, 21(4):237–256, 1997.

[155] T. Yada and M. Hirosawa. Detection of Short Protein Coding Regions within the Cyanobacterium Genome: Application of the Hidden Markov Model. *DNA Research*, 3:355–361, 1996.

[156] T. Yada, T. Sazuka, and M. Hirosawa. Analysis of Sequence Patterns Surrounding the Translation Initiation Sites on Cyanobacterium Genome Using the Hidden Markov Model. *DNA Research*, 4:1–7, 1997.

[157] T. Yada, Y. Totoki, K. Ishikawa, M. Asai, and K. Nakai. Automatic Extraction of Motifs Represented in the Hidden Markov Model from a Number of DNA Sequences. *Bioinformatics*, 14(4):317–325, 1998.

[158] C. Yarbrough and A. Thompson. International Human Genome Sequencing Consortium Announces "Working Draft" of Human Genome (National Human Genome Research Institute — Internet Newsletter). *URL http://www.nhgri.nih.gov/NEWS/sequencing_consortium.html*, 2000.