Incorporating Regional Context into Pairwise Alignments of Biological Sequences

by

Raymond Sammut

A thesis submitted for the degree of
Doctor of Philosophy
of
The Australian National University

February 2009

DECLARATION

The author draws attention to the following:

i Approximately 30% of the computer source code relevant to this thesis was sourced from the Cogent A toolkit for statistical analysis of biological sequences, version 1.0, located at http://sourceforge.net/projects/pycogent.

ii The computer source code relevant to this thesis is available from the author upon request.

iii Approximately 5% of data analyses contained in this thesis was contributed by the thesis supervisor Dr Gavin Huttley, Computational Genomics, JCSMR, ANU.

With the exception of the above, the studies and results presented in this thesis, and the relevant computer source code, constitute the original work of the author unless otherwise stated both in the text of this thesis and in the relevant computer source code notes. This thesis conforms to the ANU rules and guidelines. None of the work contained in this thesis or in any of the relevant computer source code has been submitted for the purpose of obtaining any other degree at the ANU or at any other university.

Name: Raymond Sammut

Sign: Ray Sammut

Date: 15.9.2009

ACKNOWLEDGMENTS

The Australian Research Council

Cray (Australia) Pty Ltd

Peter Maxwell
Computer Scientist, CBiS, ANU

The Python Software Foundation

The R Foundation for Statistical Computing

Greg Ewing
The Pyrex Language

Stephan Deibel
Wingware Python IDE
Archaeopteryx Software, Inc.

Martin Ott, Martin Pittenauer and Dominik Wagner
The SubEthaEdit Text Editor

The Wikimedia Foundation, Inc.

David Eberly
Derivative Approximation by Finite Differences
Geometric Tools, LLC.
www.geometrictools.com

Richard Koch, Max Horn and Gerben Wierda
http://www.texshop.org

Sylvain Lombardy and Jacques Sakarovitch
A package for drawing automata and graphs
VAUCANSON – G

Lapo Filippo Mori
Tables in LaTeX2e
The PracTEX Journal, 2007, No. 1

Patrick W. Daly
Graphics and Colour with LaTeX
MAX-PLANCK-INSTITUT FÜR AERONOMIE

K. Border
Using the kbordermatrix package

Rolf Niepraschk and Hubert Gäßlein
The sidecap package

J.A.M. Vermaseren
AXO DRAW

ABSTRACT

Background: Phylogenetically independent sequence pairs (PIPs) are units of information on evolutionary processes. One of these processes is the rate of residue insertions and deletions (indels). PIPs are sufficient for investigating the effects of secondary structure components on indels. One way of studying these effects is by identifying hydrophilic residues in coding DNA, and employing a parameter to raise the background probability of these residues relevant to the background probabilities of all other residues. In RNA encoding, the indel rate can be studied in conjunction with the substitution rate which depends on the degree of conservation in regulatory and structurally important regions. The indel rate also plays an important role in the DNA pyrosequencing technology where we need to differentiate between indels due to evolutionary processes and indels caused by the inherent physics of the sequencing machine. Overall, therefore, a PIP can be seen as an envelope of types of information that can be spatially teased out, using suitable experimental settings, with the aim of studying evolutionary processes at the molecular level.

Methods: Along with the hydrophilicity parameter, I also introduced additional parameters to model secondary structure explicitly across the two broad regions of the molecule, namely, the conserved and non-conserved regions. Only sufficient parameters were added to the likelihood function to ensure that estimators I obtained from maximum likelihood were efficient. That is, my method aims at making best use of information contained in PIPs.

For the purpose of aligning PIPs, I employed two identical sets of parameters. One set models variations in parts of the molecule that are important to structure, function, regulation and catalyses. The other set models changes that are thought to be mostly random and inconsequential to phenotype. To each of these two sets of parameters I assigned a pair-hidden Markov model (PHMM). This modelling gives me two main advantages. First, it does not treat alignment sites independently. Through its dynamic program, namely the forward algorithm, each PHMM considers

the preceding state before it resolves the current state. Second, it can deal directly with indels in the second sequence of the pairwise alignment. To optimise my two sets of parameters, I used simulated annealing to obtain maximum likelihood (ML) for all estimators endogenously.

My two PHMMs formed the lower layer of my model. A second higher layer consisted of a conventional two-state HMM designed to connect the two regional PHMMs. This configuration makes my model regional context dependent when I align PIPs made from any of the three biological encodings, namely, protein, codon, and RNA.

I constructed data sets consisting of (1) protein sequences from the BAliBASE database and (2) RNA sequences from the European ribosomal RNA database. In each case, I extracted PIPs from phylogenetic trees which I constructed from curated multiple alignments taken from these two databases. PIPs were selected using post-order traversals to ensure that each PIP had a unique ancestor. I forced evolutionary distances of PIPs to lie between 0.25 and 1.25 in the case of protein encoding and between 0.0 and 0.5 in the case of RNA encoding, after eliminating outliers. DNA equivalent PIPs of the protein PIPs were also used to construct the codon sample.

To investigate the error rate caused by the pyrosequencing machine, namely, the Roche GS 20, PIPs were constructed randomly within a specified bandwidth of percentage identity. Homopolymer insertions of exactly one base in length could be located in these PIPs using a three-region configuration with three independent sets of parameters.

Results: I found that

i the difference between rates of slow and fast replacements (or substitutions) in the two broad regions of the molecule is unequivocal across all three biological encodings, namely, RNA, protein, and codon,

ii under the assumption of regional heterogeneity, high substitution rates in coding DNA are mostly located on the surface of the molecule which is more amenable to water and furthest from the core,

iii in coding DNA, high indel rates – like high substitution rates – are mostly located in the solvent parts of the molecule, but indel lengths, whether short or long, are not,

iv confounding is strong between hydrophilicity and substitution rates when aligning codon encoded biological sequences using the mechanistic GY94 model,

v substitution rates in the two regions are independent, and appear to be normally (lognormally) distributed in slow (fast) regions of codon encoded sequences,

vi the natural selection parameter ω plays a statistically significant role when it varies freely and independently in slow and fast rate regions, and can be estimated efficiently with a two-step ML procedure,

vii the chemical agents of high codon usage and of codons that flank indels are mutually exclusive in fast rate regions of codon encoded pairwise alignments,

viii homopolymer insert errors of exactly one base committed by the Roche GS 20 are caused more often by cytosine and very rarely by thymine, even after imbalances of A/T and C/G extensions had been accounted for in the reads.

Conclusion: A regional context model, using a combination of two PHMMs and a classical HMM, provides a powerful method for aligning sequence pairs for all the three biological encoding types. Under this setting, and ensuring that sequence pairs are phylogenetically independent, biologically useful inferences can be made on molecular evolution.

CONTENTS

DECLARATION ii

ACKNOWLEDGMENTS iii

ABSTRACT iv

LIST OF TABLES x

LIST OF FIGURES xi

1. A Short Discourse on HMMs 1
   - 1.1 The Hidden Coins Problem 1
     - 1.1.1 The General State-Space Model 2
       - 1.1.1.1 Forward Recursions 4
       - 1.1.1.2 Backward Recursions 4
     - 1.1.2 The Discrete Case 5
     - 1.1.3 The HMM 8
     - 1.1.4 Maximum-Likelihood Estimation 11
     - 1.1.5 Silent States in HMMs 15
     - 1.1.6 Optimising the Likelihood 17

2. The One-Region Model 21
   - 2.1 The Probability Matrix from Blocks Model 21
     - 2.1.1 PMB Construction 21
       - 2.1.1.1 The Stationary Markov Model 24
       - 2.1.1.2 The Protein Replacement Model 28
   - 2.2 The Goldman-Yang Model 29
     - 2.2.1 The Standard Genetic Code 30
       - 2.2.1.1 The Codon Substitution Model 34
       - 2.2.1.2 Computing $\hat{d}_S$ and $\hat{d}_N$ 35
   - 2.3 The Hasegawa-Kishino-Yano Model 37
   - 2.4 Insertions and Deletions 42
   - 2.5 The Pair Hidden Markov Model 46
     - 2.5.1 Transition Probabilities Matrix 47
     - 2.5.2 Emission Probabilities Matrix 48
     - 2.5.3 One-Region Modelling 50
       - 2.5.3.1 The Trace-Back Procedure 51

3. The Two-Region Model 52
3.1 Modelling Heterogeneity in Molecular Evolution 52
3.2 The Two-Tiered HMM-PHMM Topology 55
3.2.1 The Two-Region Transition Matrix 56
3.2.2 Emission Matrices for the Two-Region Model 59
3.2.3 Two-Region Modelling 59
3.3 Model Testing 61
3.3.1 Simulations 61
3.4 Data Sets 62
3.4.1 The Protein Data Set 64
3.4.2 The Codon Data Set 66
3.4.3 The RNA Data Set 67
3.5 The Experimental Setting 68
3.6 Results 70
3.6.1 Hypotheses Testing – Protein 70
3.6.2 Replacement Rates in Hydrophilic Regions 74
3.6.3 Indels in Hydrophilic Regions 76
3.6.4 Hypotheses Testing – Codon 78
3.6.5 Collocations – Codon 78
3.6.6 Model Dependence 78
3.6.7 Hypotheses Testing – RNA 81
3.6.8 Concordances 82
4. Further Results 85
4.1 Evolutionary Rates Distributions 85
4.2 Optimising ω in the Two-Region Model 87
4.2.1 Two-Step Estimation of ω 89
4.3 Indel Analyses 91
4.3.1 Regional Indel Averages 91
4.3.2 Regional Codon Preference 92
4.3.3 Regional Codon Usage 93

5. Detecting Pyrosequencing Errors 94
5.1 Real-Time Sequencing 94
5.1.1 The ELIDA Concept 94
5.2 Massively Parallel Pyrosequencing 96
5.3 The Homopolymer Problem 97
5.3.1 The Homopolymer Effect – Experimental Setting 98
5.3.1.1 The Experimental Data Set 98
5.3.1.2 The Three-Region Model 99
5.3.1.3 Second Order Markov Chain 101
5.3.1.4 Emission Probabilities of Monoinserts 102
5.3.2 The Homopolymer Effect – Hypothesis Testing 105
5.3.3 The Homopolymer Effect – Results 107
5.3.4 Conclusions 110
6. Discussion 112
APPENDICES
A. Chapter One 117
A.1 Silent Chains 117
A.2 General Formulas 119
A.2.1 Notation for the General Algorithms 119
A.2.2 The General Forward Algorithm 121
A.2.3 The General Backward Algorithm 122
A.2.4 The General Baum-Welch Algorithm 122
B. Chapter Two 124
B.1 Taylor Series Expansions 124
B.2 The Protein Replacement Model 126
B.3 Pascarella and Argus Methods 127
C. Chapter Four 129
C.1 Codon Preference 129
C.2 Codon Usage in Regions 1 and 2 130
D. Chapter Five 131
Bibliography 136

LIST OF TABLES

1.1 Churchill's Equations and the Two-Coin Problem 9

1.2 HMM Posterior Probabilities 12

1.3 ML Estimation 14

2.1 Genetic Code 30

2.2 Variation of Synonymous and Nonsynonymous Counts 31

2.3 Nei and Gojobori versus Goldman and Yang – 1 33

2.4 Nei and Gojobori versus Goldman and Yang – 2 36

3.1 PIPs Sampling 67

3.2 Hypotheses Testing – Protein PIPs 71

3.3 Hypotheses Testing – Codon PIPs 79

3.4 Collocations - Codon 80

3.5 Model Dependence 81

3.6 Hypotheses Testing – RNA PIPs 81

4.1 Fast Rates Distribution 86

4.2 Moments for Slow and Fast Rates. 87

4.3 Two-Step Estimation 89

4.4 ω Across Two Regions 90

5.1 Homopolymer Experiment – Results 110

B.1 The PMB Matrix R 126

C.1 Codon Preference in Region Two 129

C.2 Regional Codon Usage 130

LIST OF FIGURES

1.1 HMM Matrices – Arbitrary Specification 6

1.2 Transition Matrix with Silent States 16

2.1 PMB's Quadratic Equation 23

2.2 The HKY Q Matrix 38

2.3 Spectral decomposition of HKY Q 38

2.4 Product of Spectral Decomposition of HKY Q 39

2.5 Three-State PHMM – Schematic Diagram 46

2.6 PHMM Emission Matrix 50

2.7 One-Region Modelling – Schematic Diagram 50

3.1 Double PHMM topology 56

3.2 Conceptual Two-Region Transition Matrix 57

3.3 Region Switch 58

3.4 Two-Region Transition Matrix Implementation 58

3.5 Double PHMM Emission Matrices 59

3.6 Two-Region Modelling – Schematic Diagram 60

3.7 Simulation Tables 63

3.8 PIPs Construction 65

3.9 PIPs Boxplot 66

3.10 Protein Bins Boxplots 68

3.11 RNA Boxplots Before and After Pruning 69

3.12 Hydrophilicity Testing – Protein 75

3.13 Hydrophilicity Testing – Codon 80

4.1 Slow Rates Distribution 85

4.2 Fast Rates Distribution 86

5.1 ELIDA Concept 94
5.2 Cyclic ELIDA 95
5.3 Three-Region Model – 1 99
5.4 Three-Region Model – 2 100
5.5 Coding the Second Order Markov Process 103
5.6 Geometric Distribution 108
5.7 Three-Region Transition Matrix 109
A.1 HMM Matrices for a Model Example 120
A.2 HMM Transition Matrix Reduction 121

CHAPTER 1
A Short Discourse on HMMs

1.1 The Hidden Coins Problem

CHURCHILL (1989) was first to use HMMs for the analysis of biological sequences. Earlier workers had used methods that could not deal effectively with the heterogeneity of DNA composition (CHURCHILL, 1989). A more advanced model, such as an HMM, is more suitable for teasing out hidden processes that cause compositional variation. Other methods are also possible. For example, one can scan a DNA sequence with a fixed-size window and methodically compute statistical summaries. This method, however, is subjective as it requires the arbitrary choice of window size (CHURCHILL, 1989).

To analyse DNA data, CHURCHILL (1989) did not strictly use an HMM. He constructed instead a state-space model which is a generalisation of the HMM and other stochastic models. He based his model on previous studies by KITAGAWA (1987) and other workers. Kitagawa had worked on models that perform *smoothing* on non-stationary time series. The smoothing problem in time series analyisis – where data are noisy, and exhibit both an abrupt and a gradual change in the mean – is a long-standing problem found in many fields. For example, a type of state-space model called the Kalman Filter allows econometricians to infer the smoothed estimate of the GNP (gross national product) for a specific year given an economic time series spanning several years (HAMILTON, 1994). Various smoothing methods exist, but the one proposed by Kitagawa is of especial interest in computational biology. Kitagawa's method incorporates two procedures, namely, the forward and backward algorithms. In the discrete case of DNA data, these algorithms can effectively be synthesised numerically using dynamic programming. Fundamentally, the model uses Bayes' theorem together with the law of total probability and conditional probability. I set out to explain this basic model using the notation in CHURCHILL (1989) in conjunction with a two-coin tossing experiment for illustration.

1.1.1 The General State-Space Model

CHURCHILL (1989) considers a finite set of n random variables $\{Y_i : i = 1, 2, ..., n\}$. Each of these variables has a probability distribution determined by a corresponding state $\{s_i : i = 1, 2, ..., n\}$. This means that Y_i is not necessarily Gaussian distributed. The sequence of observed outcomes up to time t is denoted by $y^t = y_1, y_2, ..., y_t$ and the corresponding sequence of states by $s^t = s_1, s_2, ..., s_t$. Each observation has a probability distribution which is denoted by $p(y_t|s_t, y^{t-1})$. The term y^{t-1} simply emphasises that the observations are not necessarily independently and identically distributed (i.i.d.).

In a two-coin tossing experiment, in which the coins are *hidden*, the sequence of states is unobservable. All I can do is try to infer this sequence from the data. That is, I want to estimate a smoothed average from my noisy and non-stationary sequence of observations. Given the n observations, I do this by using the joint distribution of s_t and s_{t+1}, namely $p(s_t, s_{t+1}|y^n)$, and integrate out s_{t+1}. To carry out this integration I use the following equation based on Kitagawa's 2.5[1]

$$p(s_t|y^n) = \int_{-\infty}^{\infty} p(s_t, s_{t+1}|y^n) ds_{t+1}. \tag{1.1}$$

The term $p(s_t, s_{t+1}|y^n)$ in 1.1 can be reformulated using the definition of conditional probability, namely $P(A \cap B) = P(A)P(B|A)$, as

$$p(s_t, s_{t+1}|y^n) = p(s_{t+1}|y^n)p(s_t|s_{t+1}, y^n). \tag{1.2}$$

From the second term of the right-hand side of 1.2 it is easy to see that each time I estimate s_{t+1}, observations that follow time t (that is, from y^{t+1} onwards) become redundant and can be discarded. This allows me to re-write 1.2 as

$$p(s_t, s_{t+1}|y^n) = p(s_{t+1}|y^n)p(s_t|s_{t+1}, y^t). \tag{1.3}$$

By applying the definition of conditional probability, namely $P(B|A) = \frac{P(A \cap B)}{P(A)}$, to the term $p(s_t|s_{t+1}, y^t)$ in 1.3, I can re-write 1.3 as

[1]For a theoretical treatment and derivation of this equation, see KITAGAWA (1987), p. 1033.

$$p(s_t, s_{t+1}|y^n) = \frac{p(s_{t+1}|y^n)p(s_t, s_{t+1}|y^t)}{p(s_{t+1}|y^t)},$$

and applying the definition once more, this time to the term $p(s_t, s_{t+1}|y^t)$, I obtain

$$p(s_t, s_{t+1}|y^n) = \frac{p(s_{t+1}|y^n)p(s_{t+1}|s_t)p(s_t|y^t)}{p(s_{t+1}|y^t)}. \tag{1.4}$$

What remains now is to substitute 1.4 into 1.1 to obtain the following equation

$$p(s_t|y^n) = p(s_t|y^t)\int_{-\infty}^{\infty} \frac{p(s_{t+1}|y^n)p(s_{t+1}|s_t)}{p(s_{t+1}|y^t)} ds_{t+1}. \tag{1.5}$$

Note that the term $p(s_t|y^t)$ in 1.5 can be taken outside of the integral sign because I want to integrate out s_{t+1} and not s_t, as I stated earlier.

Equation 1.5 brings me close to computing a *smoother* for my sequence of observations y^n. Before proceeding, however, there are some terms in this equation that deserve further attention.

First, $p(s_{t+1}|s_t)$ is a first-order Markov process which Churchill calls the *state equation.* In HMM terminology this is referred to as the **transition matrix** which is pre-specified by the investigator. Second, $p(s_{t+1}|y^t)$ is defined through the law of total probability, namely, $P(A) = \sum_i P(A|B_i)P(B_i)$. That is

$$p(s_t|y^{t-1}) = \int_{-\infty}^{\infty} p(s_t|s_{t-1}, y^{t-1})p(s_{t-1}|y^{t-1})ds_{t-1}. \tag{1.6}$$

To define the term $p(s_{t-1}|y^{t-1})$ in 1.6, I appeal to Bayes' theorem, namely, $P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_k P(B_k)P(A|B_k)}$. That is

$$p(s_t|y^t) = \frac{p(s_t|y^{t-1})p(y_t|s_t, y^{t-1})}{\int_{-\infty}^{\infty} p(s_u|y^{t-1})p(y_t|s_u, y^{t-1})ds_u}. \tag{1.7}$$

Finally, the term $p(y_t|s_t, y^{t-1})$ in 1.7 is defined by Churchill as the *observation equation.* In HMM terminology this is referred to as the **emission matrix** and, like the transition matrix, is pre-specified.

The smoothed average of y^n can now be obtained by employing Churchill's forward and backward recursions.

1.1.1.1 Forward Recursions

I start with some arbitrary initial value y_0 and a corresponding initial state s_0 before making the first forward pass. Substituting $p(s_0|y^0)$ into 1.6 gives $p(s_1|y^0)$, and substituting this value into 1.7 gives $p(s_1|y^1)$. Thus, through the following recursions:

$$p(s_1|y^0) = \int_{-\infty}^{\infty} p(s_1|s_0, y^0)p(s_0|y^0)ds_0,$$

$$p(s_2|y^1) = \int_{-\infty}^{\infty} p(s_2|s_1, y^1)p(s_1|y^1)ds_1,$$

$$\ldots \quad = \quad \ldots$$

$$p(s_{n+1}|y^n) = \int_{-\infty}^{\infty} p(s_{n+1}|s_n, y^n)p(s_n|y^n)ds_n,$$

I obtain the value $p(s_{n+1}|y^n)$.

1.1.1.2 Backward Recursions

The smoothed average can now be computed directly through recursive substitutions of the value obtained from the forward algorithm into 1.5, producing the following backward recursions:

$$p(s_n|y^n) = p(s_n|y^n) \int_{-\infty}^{\infty} \frac{p(s_{n+1}|y^n)p(s_{n+1}|s_n)}{p(s_{n+1}|y^n)} ds_{n+1}, \tag{1.8}$$

$$p(s_{n-1}|y^n) = p(s_{n-1}|y^{n-1}) \int_{-\infty}^{\infty} \frac{p(s_n|y^n)p(s_n|s_{n-1})}{p(s_n|y^{n-1})} ds_n,$$

$$\ldots \quad = \quad \ldots$$

$$p(s_1|y^n) = p(s_1|y^1) \int_{-\infty}^{\infty} \frac{p(s_2|y^n)p(s_2|s_1)}{p(s_2|y^1)} ds_2.$$

I should point out here that in 1.8 (the first backward pass) the integral is required to sum to one. This shows that the first order Markov process in HMM studies should always be specified as a row stochastic matrix.

1.1.2 The Discrete Case

When implementing Churchill's state-space model to DNA, the observations take a discrete form, and therefore I need a discrete formulation of the key equations. I denote the transition matrix by $\boldsymbol{T}$ whose element T_{ij} describes a transition from the state that has index i to the state that has index j, and the emission matrix by $\boldsymbol{E}$ whose element E_{jk} describes the emission of the item that has index k when the HMM is in the state that has index j. The number of states is finite and is denoted by r, and the number of observable items is also finite and is denoted by K. The key equations, namely, 1.5, 1.6, and 1.7 can be re-written respectively in discrete form as follows:

$$p(s_t^{(i)}|y^n) = p(s_t^{(i)}|y^t) \sum_{j=1}^{r} \frac{T_{ij} p(s_{t+1}^{(j)}|y^n)}{p(s_{t+1}^{(j)}|y^t)}, \quad i = 1, 2, ..., r. \tag{1.9}$$

$$p(s_t^{(j)}|y^{t-1}) = \sum_{i=1}^{r} T_{ij} p(s_{t-1}^{(i)}|y^{t-1}), \quad j = 1, 2, ..., r. \tag{1.10}$$

$$p(s_t^{(j)}|y^t) = \frac{E_{jk} p(s_t^{(j)}|y^{t-1})}{\sum_{i=1}^{r} E_{ik} p(s_t^{(i)}|y^{t-1})}, \quad j = 1, 2, ..., r; \;\; k = 1, 2, ..., K. \tag{1.11}$$

A short draw, say, $\mathbb{HHTHTHTTHHHH}$, from my two-coin experiment can now be used as an example to illustrate how these three equations are used to compute the smoothed output of the sequence of heads ($\mathbb{H}$) and tails ($\mathbb{T}$). The transition and emission matrices shown in Figure 1.1 are specified first. These matrices are exogenous to the model since their elements are merely my best guess based on intuition. $\mathfrak{B}$ is the begin state of the system. Its sole purpose is to determine randomly whether the system starts with coin 1 or coin 2. Similarly, $\mathfrak{E}$ is the end state

whose sole purpose is to randomly end the process either at coin 1 or coin 2. Both $\mathfrak{B}$ and $\mathfrak{E}$ do not emit a symbol and hence they are referred to as *silent* states. The iterative computations using the three key equations in accordance with the forward and backward recursions are shown in Table 1.1.

$$
\begin{array}{cc}
 & \begin{array}{ccccc} s^{(1)} & s^{(2)} & \mathfrak{E} & \quad\mathbb{H} & \mathbb{T} \end{array} \\
\begin{array}{c} \\ \mathfrak{B} \\ \textit{(Coin 1)}\ \ s^{(1)} \\ \textit{(Coin 2)}\ \ s^{(2)} \end{array} &
\left[\begin{array}{ccc} 0.2 & 0.8 & 0.0 \\ 0.3 & 0.7 & 0.0^{\dagger} \\ 0.6 & 0.4 & 0.0^{\dagger} \end{array}\right]
\left[\begin{array}{cc} 0.0 & 0.0 \\ 0.1 & 0.9 \\ 0.2 & 0.8 \end{array}\right]
\end{array}
$$

Figure 1.1: The diagram shows the two matrices of a two-state HMM emitting one of two symbols at each time slot. The first matrix is the transition matrix $\boldsymbol{T}$ and has two states, namely, $s^{(1)}$ and $s^{(2)}$. The second matrix is the emission matrix $\boldsymbol{E}$ and has two symbols, namely, $\mathbb{H}$ and $\mathbb{T}$. The first two elements in the first row of the transition matrix form the initial vector and their values are set arbitrarily. The third element in this row is set to zero because no flow is allowed from the begin state $\mathfrak{B}$ to the end state $\mathfrak{E}$. The other four elements in the first two columns of this matrix model the Markov process with states on the left-hand side being the source, whereby $s^{(1)}$ represents the first hidden coin and $s^{(2)}$ represents the second. The first two elements of the emission matrix are set to zero because $\mathfrak{B}$ is silent and hence does not emit a symbol.

†These probabilities are very small and are shown rounded to 0.0.

The first two rows of the computations in Table 1.1 are the emission probabilities. They are invoked by the observed outcome shown in the first row of the table. This row shows the observed sequence with corresponding discrete time t slots in the second row. Together these two rows constitute the input of the discrete space-time model. The last two rows of the computations show the posterior probabilities for each possible outcome (with $K = 2$ in this case) at each time slot. The bottom row of the table shows the inference made by the investigator from the smoothed output, namely, the posterior probabilities. It is interesting to note that even with the parameters of the two matrices $\boldsymbol{T}$ and $\boldsymbol{E}$ being intuitively specified, the inference turns out to be accurate at every time slot. However, what should be noted here is that these parameters are not a major assumption of the model. I am more concerned about the fact that the model has two states. This choice is only tentative, and there is nothing to suggest as to why I should be assuming that the sequence is being generated by exactly two coins. Since these coins are hidden from me, I simply do not know how many coins there actually are. I still need the theory that can guide me in making a correct guess.

For this purpose, I first need a method that can help me find estimators that

can "best" explain the observed sequence of heads and tails *given the model.* Such a method would provide me with a probability score of observing the data *after* I had hypothesised the number of coins. I would obtain a score for each number of coins in my hypothesis, say, one coin, or two coins. Using the appropriate statistic, I could then analyse the resulting scores in order to infer with some desired confidence the correct number of coins.

This technique is useful in DNA analysis. For example, starting with a six-state model (or six coins by my analogy), CHURCHILL (1989) found that a four-state model best explained the composition of bacteriophage λ DNA. His analysis was mostly in agreement with that obtained by earlier workers using density gradient centrifugation techniques. Both the GC content in the first homogeneous section of the molecule and the average GC content in the second heterogeneous section were accurately measured. The only shortcoming of the state-space model in this application was that a discrete model could not faithfully describe compositional fluctuations in the highly heterogeneous region.

My motivation for using state-space modelling stems from the fact that the pairwise alignment is unobservable. It is the product of putative random processes whose generators are hidden and for their greater part not extant. The pairwise alignment problem is therefore best approached with a well structured probability model such as the state-space model. Furthermore, the intuitive notion that heterogeneity of the molecular structure has a significant effect on the "true" alignment of a DNA sequence pair continues to challenge computational biologists.

Pairwise alignment methods, both stochastic and deterministic, have traditionally assumed that the molecule is homogeneous. By employing Churchill's state-space model within a larger HMM topology I propose a novel way of aligning pairs of DNA sequences. My method is regional context dependent. That is, the algorithm is freed from the usual constraint that each base pair and each indel at each site of the alignment has to be "smoothed" across the entire linear DNA molecule. My model has the additional degree of freedom to position or to "smooth" each base pair and each indel in either the region which is conserved or in the region where a putatively high rate of mutations does not pose a deleterious impact on phenotype.

Furthermore, the model can operate in each region independently from the other region, in the same way one coin shows a head or a tail irrespective of what the other coin is showing.

1.1.3 The HMM

The general state-space model described by CHURCHILL (1989) does not address the issue of computational efficiency. RABINER (1989) shows that to compute the probability $P(y^n|HMM)$, that is, the probability of observing the sequence $y^n = y_1 y_2 ... y_n$ given the HMM model, would require computing the equation

$$P(y^n|HMM) = \sum_{s_1 \ldots s_n} \mathfrak{B}_{s_1} E_{s_1 y_1} T_{s_1 s_2} E_{s_2 y_2} \ldots T_{s_{(n-1)} s_n} E_{s_n y_n} \mathfrak{E}, \qquad (1.12)$$

where in the case of my two-coin problem $s_t \in \{s^{(1)}, s^{(2)}\}$, $y_t \in \{\mathbb{H}, \mathbb{T}\}$ and $t = 1, 2, \ldots, n$. Computing 1.12, in turn, would need $(2n-1)r^n$ multiplications and $r^n - 1$ additions (e.g. RABINER, 1989, p. 262), recalling that r is the number of states assumed to be in the system. In the case of a two-state model with a short sequence of, say, 100 observations, these computations could be handled by a machine with relative ease. However, when dealing with alignments of long sections of DNA, an efficient dynamic program would be needed. Two equivalent dynamic programs are available for this purpose. One is the forward algorithm, which I list in matrix form as follows:

$$\boldsymbol{f}_1 = \mathfrak{B}' \bullet \boldsymbol{E}(y_1),$$

$$\boldsymbol{f}_t = \left(\boldsymbol{T}' \boldsymbol{f}_{t-1}\right) \bullet \boldsymbol{E}(y_t),$$

$$P(y^n|HMM) = \boldsymbol{f}_n' \mathfrak{E},$$

Outcome		$\mathbb{H}$	$\mathbb{H}$	$\mathbb{T}$	$\mathbb{H}$	$\mathbb{T}$	$\mathbb{H}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{T}$	
t	0	1	2	3	4	5	6	7	8	9	10	11	12	13
E_{1k}		0.1000	0.1000	0.9000	0.1000	0.9000	0.1000	0.9000	0.9000	0.1000	0.1000	0.1000	0.9000	
E_{2k}		0.2000	0.2000	0.8000	0.2000	0.8000	0.2000	0.8000	0.8000	0.2000	0.2000	0.2000	0.8000	
$p(s_t^{(1)}\|y^{t-1})$		0.5400	0.4890	0.5029	0.4403	0.5153	0.4366	0.5162	0.4363	0.4604	0.5103	0.4972	0.5007	0.4410
$p(s_t^{(2)}\|y^{t-1})$		0.4600	0.5110	0.4971	0.5597	0.4847	0.5634	0.4838	0.5637	0.5396	0.4897	0.5028	0.4993	0.5590
$p(s_t^{(1)}\|y^t)$	0.2000	0.3699	0.3237	0.5323	0.2823	0.5446	0.2793	0.5455	0.4655	0.2990	0.3426	0.3309	0.5301	
$p(s_t^{(2)}\|y^t)$	0.8000	0.6301	0.6763	0.4677	0.7177	0.4554	0.7207	0.4545	0.5345	0.7010	0.6574	0.6691	0.4699	
$p(s_t^{(1)}\|y^n)$	0.2277	0.4315	0.3087	0.5981	0.2716	0.5907	0.3076	0.6009	0.5048	0.3294	0.3896	0.3231	0.5302	0.4408
$p(s_t^{(2)}\|y^n)$	0.7943	0.5905	0.7133	0.4239	0.7504	0.4218	0.6895	0.4348	0.4952	0.6706	0.6104	0.6769	0.4698	0.5592
Inference		H	H	T	H	T	H	T	T	H	H	H	T	

Table 1.1: The table shows the computations using Churchill's equations for the discrete state-space model after the transition and emission matrices had been arbitrarily specified. k is 1 if $\mathbb{H}$ is emitted, and 2 if $\mathbb{T}$ is emitted. The last two rows of the computations show the smoothed output, and the bottom row shows the inference made from the smoother whereby my *guess* is $\mathbb{H}$ if $p(s_t^{(2)}|y^n) > p(s_t^{(1)}|y^n)$.

where $\boldsymbol{f}_t$ is the forward score up to time t, $t = 2, 3, \ldots, n$, and $\mathfrak{B}$ and $\mathfrak{E}$ are both $r \times 1$ vectors of probabilities. RABINER (1989) does not formally assign begin and end probabilities but others, such as DURBIN *et al.* (1998) and ISAEV (2004), do. $\boldsymbol{E}(y_t)$ is the column of emission probabilities of the emission matrix $\boldsymbol{E}$ corresponding to the item y_t emitted at time slot t. $\boldsymbol{T}$ is the transition matrix, and $(\bullet)$ is the Hadamard product [2].

The other is the backward dynamic program which I list in matrix form as follows:

$$\boldsymbol{b}_n = \mathfrak{E},$$

$$\boldsymbol{b}_t = \boldsymbol{T}(\boldsymbol{b}_{t+1} \bullet \boldsymbol{E}(y_{t+1})),$$

$$P(y^n|HMM) = \mathfrak{B}(\boldsymbol{b}_1 \bullet \boldsymbol{E}(y_1)),$$

where $\boldsymbol{b}_t$ is the backward score up to time t, $t = n-1, n-2, \ldots, 1$.

It can be shown (e.g. RABINER, 1989, p. 263) that these two dynamic programs can be combined to provide a direct way of computing a smoother for state $s^{(i)}$, given the data and the HMM, at time slot t with the following formulation:

$$p(s_t^{(i)}|y^n, HMM) = \frac{f_t^{(i)} b_t^{(i)}}{\sum_{j=1}^{r} f_t^{(j)} b_t^{(j)}}, \tag{1.13}$$

and that the denominator of 1.13 is simply $P(y^n|HMM)$. The latter quantity is very helpful in speeding up computations since it has to be computed only once. Using 1.13, I can reproduce the results obtained from Churchill's equations (Table 1.1). These are shown in Table 1.2. They differ slightly from those obtained from Churchill's equations because the two formulations treat the begin state differently. The differences are in fact more noticeable over the initial time slots. It is easy to show that with long sequences computations from the two formulations converge to the same values.

[2]The Hadamard product of matrices $A_{m\times n}$ and $B_{m\times n}$ is defined by $[A \bullet B]_{ij} = [A]_{ij}[B]_{ij} \; \forall \, 1 \leq i \leq m,\ 1 \leq j \leq n$.

1.1.4 Maximum-Likelihood Estimation

I have used the HMM as a stochastic signalling device to simulate a hypothetical two-coin signal. I could potentially be receiving this signal from an unknown source of interest, and I wish to determine whether this signal was generated by exactly two coins and not, for example, by a single coin. For this purpose, I convert my HMM from a signalling device to a signal receiver. That is, I adjust the parameters in order to "tune" my HMM to the incoming signal. To do so, I have to estimate the model parameters by maximising the likelihood function

$$f(y_1, y_2, \cdots, y_n|\boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i, \boldsymbol{\theta}). \tag{1.14}$$

The right-hand side of 1.14 is the probability of receiving my particular signal – consisting of a series of heads and tails – on the assumption that the functions $f(y_1, \boldsymbol{\theta}), f(y_2, \boldsymbol{\theta}), \cdots, f(y_n, \boldsymbol{\theta})$ are independent. Independence means that these functions have a joint distribution which is the likelihood function denoted by $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{Y})$, where $\boldsymbol{Y}$ represents the data and $\boldsymbol{\theta}$ is the vector that holds all the parameters. In my two-coin experiment, these consist of all the cell values of the transition and the emission matrices of the HMM.

The aim is to find the $\boldsymbol{\theta}$ that maximises $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{Y})$. GREENE (1997, pp. 198-218) provides a discussion on how to deal with this problem. It is preferable if $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{Y})$ can be maximised by formally deriving first and second derivatives. However, formal derivations when working with HMMs, in general, are not practical (e.g. RABINER, 1989, p. 264). Alternatively, first and second derivatives can be obtained by means of numeric approximations and PRESS *et al.* (1992, pp. 186-9) deal with the critical issues that affect computer implementation. For the purpose of my experiments with biological sequences, I considered only those methods that do not require derivatives. One such method is the expectation-maximization (EM) algorithm by DEMPSTER *et al.* (1977). The EM method is known to be stable and intuitive, and its special case – the Baum-Welch algorithm (BAUM *et al.*, 1970) – has been standardly used for parameter estimation for HMMs (DURBIN *et al.*, 1998, p. 63).

Outcome	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{T}$	$\mathbb{H}$	$\mathbb{T}$	$\mathbb{H}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{T}$
t	1	2	3	4	5	6	7	8	9	10	11	12
$p(s_t^{(1)}\|y^n)$	0.1347	0.3714	0.5649	0.2685	0.5913	0.2776	0.5247	0.5049	0.3294	0.3896	0.3231	0.5301
$p(s_t^{(2)}\|y^n)$	0.8653	0.6286	0.4351	0.7315	0.4087	0.7224	0.4753	0.4951	0.6706	0.6104	0.6769	0.4699
Inference	H	H	T	H	T	H	T	T	H	H	H	T

Table 1.2: The table shows computations of posterior probabilities using the HMM formulation in RABINER (1989).

The Baum-Welch algorithm is build around the following two definitions:

1. $\sum_{t=1}^{n} p(s_t^{(i)}|y^n, HMM)$ is the expected number of visits to state $s^{(i)}$, $i = 1, 2, ..., r$,

2. $\sum_{t=1}^{n} p(s_t^{(i)}, s_{t+1}^{(i)}|y^n, HMM)$ is the expected number of transitions from state $s^{(i)}$ to state $s^{(j)}$, $i, j = 1, 2, ..., r$,

where

$$p(s_t^{(i)}|y^n, HMM) = \frac{f_t^{(i)} b_t^{(i)}}{\sum_{j=1}^{r} f_t^{(j)} b_t^{(j)}}, \ i = 1, 2, \ldots, r,$$

and

$$p(s_t^{(i)}, s_{t+1}^{(j)}|y^n, HMM) = \frac{f_t^{(i)} T_{ij} E_{jk} b_{t+1}^{(j)}}{\sum_{i=1}^{r}\sum_{j=1}^{r} f_t^{(i)} T_{ij} E_{jk} b_{t+1}^{(j)}}, \ i, j = 1, 2, \ldots, r,$$

and k is the index of the item emitted at time slot $t + 1$.

Using these definitions, together with the Lagrange multiplier, differential calculus, and ensuring that probabilities in $\mathfrak{B}$, and in rows of $\boldsymbol{T}$ and of $\boldsymbol{E}$ add to one, it can be shown that

$$\overline{\mathfrak{B}_j} = p(\mathfrak{B}, s_1^{(j)}|y^n, HMM), \ j = 1, 2, \ldots, r. \tag{1.15}$$

$$\overline{T_{ij}} = \sum_{t=2}^{n-1} p(s_t^{(i)}, s_{t+1}^{(j)}|y^n, HMM), \ i, j = 1, 2, \ldots, r. \tag{1.16}$$

$$\overline{\mathfrak{E}_i} = p(s_n^{(i)}, \mathfrak{E}|y^n, HMM), \ i = 1, 2, \ldots, r. \tag{1.17}$$

$$\overline{E_{jk}} = \sum_{t=1}^{n} p(s_t^{(j)}|y^n, HMM)\delta_{y_t=v_k}, \ j = 1, 2, \ldots, r; \ k = 1, 2, \ldots, K. \tag{1.18}$$

In 1.18, δ is the Kronecker delta. It implies that $\overline{E_{jk}}$ has to be estimated for each

element in the items set $\{v_1, v_2, \ldots, v_K\}$. In the case of the two-coin experiment, for example, $\overline{E_{jk}}$ has to be estimated twice since $K = 2$.

ISAEV (2004), RABINER (1989), and others show how 1.15 to 1.18 are used in conjunction with training data to re-estimate probabilities for $\mathfrak{B}$, $\mathfrak{E}$, $\boldsymbol{T}$, and $\boldsymbol{E}$ using an iterative procedure, whereby the value of the likelihood function increases at each iteration up to convergence. Convergence guarantees optimal probabilities, and hence a global maximum, if the likelihood function is concave throughout its surface, otherwise we could have simply a local maximum. In the latter case, several techniques can be tried to improve the maximum likelihood (ML) values. A common approach is to re-start the iterative procedure using different initial values – drawn from some random distribution – in the $\mathfrak{B}$ vector.

ML	$T_1^{(1)}$	$E_1^{(1)}$	$T_{11}^{(2)}$	$T_{21}^{(2)}$	$E_{11}^{(2)}$	$E_{21}^{(2)}$	LR	p-value
1	1.0	0.430	0.80162	0.26556	0.09577	0.87357	25.44852	0.00001
2	1.0	0.475	0.63112	0.46762	0.06062	0.99514	6.81287	0.07811
3	1.0	0.355	0.73617	0.32727	0.00169	0.81042	18.89060	0.00029
4	1.0	0.275	0.76680	0.64945	0.01348	0.99031	6.31871	0.09709
5	1.0	0.425	0.66355	0.45876	0.01118	0.99463	9.72834	0.02102
6	1.0	0.405	0.74869	0.36806	0.02155	0.96012	26.71640	0.00001
7	1.0	0.425	0.69439	0.45584	0.05189	0.98751	9.90730	0.01937
8	1.0	0.430	0.64122	0.47691	0.02006	0.98172	7.10719	0.06856
9	1.0	0.300	0.89237	0.16435	0.07527	0.65379	22.40755	0.00005
10	1.0	0.285	0.79738	0.52630	0.03895	0.95136	10.47002	0.01497

Table 1.3: The table shows results from ten simulated experiments. In each experiment, a continuous random stream of heads and tails was generated using a two-state HMM. A large section of this stream was sliced into ten sequences, each consisting of 200 observations. Each of these sequences was used as training data first for a one-state HMM and then for a two-state HMM, using arbitrary initial values for each. The Baum-Welch algorithm was used to obtain the maximum likelihood (ML) estimators for each. In the second last column were recorded the computations of the likelihood ratio (LR) test for each experiment with three degrees of freedom. Superscripts indicate one- or two-coin, and subscripts indicate cell addresses of the respective HMM.

Table 1.3 shows the optimal results after the Baum-Welch algorithm was applied to a *one-coin* and a *two-coin* ten experiments, with results from each experiment recorded as shown in each row. The third column suggests that the signal of heads and tails, in my *two hidden coins* tossing experiment, is originating from a biased coin emitting heads approximately 40% of the time. This would be the conclusion of an untrained eye. However, p-values in the last column strongly suggest

that the one-coin hypothesis is not true. Instead, the source turns out to be more complex than just a biased coin. With the exceptions of experiments 2, 4, and 8, at the 5% level of significance, the evidence is clearly in favour of the alternative hypothesis with three degrees of freedom.

The true emitter is almost certain to consist of two coins. Furthermore, from column four I can infer that once in state one, the system is likely to remain in state one, and likewise (from column five) for state two. However, the tendency to remain in the same state is higher for state one than for state two. At the same time, state one is more likely to emit tails, which explains the bias in favour of tails in the incoming stream.

My experiment illustrates the effectiveness of the HMM. It is a device that provides the investigator with the means to make good inference in what otherwise would be a highly arguable problem. The most remarkable characteristic of the HMM, in spite of its complex theory, is its simplicity when applied to real data.

The random processes of molecular evolution, stored in the four symbols of DNA, can be viewed as signals. These signals, like the series of heads and tails emitted by hidden coins, can also be studied and understood with a signalling device such as the HMM. In the next chapter I show how an HMM with three states can be optimised in order to identify the best alignment between two DNA sections drawn from two different molecules where one is considered to have evolved from the other through mostly random mutations consisting of DNA base substitutions, base insertions and base deletions.

1.1.5 Silent States in HMMs

I have discussed the HMM as having only – with the exceptions of the begin $\mathfrak{B}$ and the end $\mathfrak{E}$ states – the type of states which, when visited, they emit an element from the finite set $\mathcal{V}$ of observable items $\{v_1, v_2, \ldots, v_K\}$. There is another type of state that can be incorporated in HMM design and is called *silent state*. This HMM element has the property of not emitting a symbol when visited. This means that a silent state, or a group of silent states, should not be looped back onto themselves as this would cause the HMM to degenerate into an uninteresting oscillator. Another

implication is that when maximising the HMM, using a method such as the Baum-Welch algorithm, *all* possible paths between *all* possible pairs of states through *all* silent states have to be taken into account. For this reason, I define the transition matrix as shown in Figure 1.2. This matrix incorporates both the $\mathfrak{B}$ and $\mathfrak{E}$ vectors, and sets the probability of transiting from state $\mathfrak{B}$ to state $\mathfrak{E}$ to 0.0. Silent states are denoted by $d^{(n)}$, $n = 1, 2, \ldots, D$.

I also introduce the notation Ω_{ab} defined as *the probability of a transition from state* `"a"` *to state* `"b"` *through all the possible silent chains between* `"a"` *and* `"b"`. By a silent chain I mean a flow from state `"a"` to state `"b"` in which all transit states are silent. Furthermore, either `"a"` or `"b"` or both can be either emitting or silent. The former can also be $\mathfrak{B}$ and the latter can also be $\mathfrak{E}$. In Appendix A.1, I explain a method of how to determine all the possible silent chains between states `"a"` and `"b"` in an HMM with any number of silent states.

ISAEV (2004) covers HMMs with silent states in some detail, and provides the fundamental algebra that generalises the forward algorithm. In Appendix A.2, I show how this algebra can be extended and summarised into compact formulas so that generalised forward, backward, and Baum-Welch algorithms for HMMs with any number of silent states can easily be implemented in computer code.

$$
\begin{array}{c}
 \\
\mathfrak{B} \\
d^{(1)} \\
d^{(2)} \\
\vdots \\
d^{(D)} \\
s^{(1)} \\
s^{(2)} \\
\vdots \\
s^{(r)}
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc}
d^{(1)} & d^{(2)} & \ldots & d^{(D)} & s^{(1)} & s^{(2)} & \ldots & s^{(r)} & \mathfrak{E}
\end{array} \\
\left[\begin{array}{ccccccccc}
T_{\mathfrak{B}z} & T_{\mathfrak{B}z} & \ldots & T_{\mathfrak{B}z} & T_{\mathfrak{B}j} & T_{\mathfrak{B}j} & \ldots & T_{\mathfrak{B}j} & 0.0 \\
T_{wz} & T_{wz} & \ldots & T_{wz} & T_{wj} & T_{wj} & \ldots & T_{wj} & T_{w\mathfrak{E}} \\
T_{wz} & T_{wz} & \ldots & T_{wz} & T_{wj} & T_{wj} & \ldots & T_{wj} & T_{w\mathfrak{E}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
T_{wz} & T_{wz} & \ldots & T_{wz} & T_{wj} & T_{wj} & \ldots & T_{wj} & T_{w\mathfrak{E}} \\
T_{iz} & T_{iz} & \ldots & T_{iz} & T_{ij} & T_{ij} & \ldots & T_{ij} & T_{i\mathfrak{E}} \\
T_{iz} & T_{iz} & \ldots & T_{iz} & T_{ij} & T_{ij} & \ldots & T_{ij} & T_{i\mathfrak{E}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
T_{iz} & T_{iz} & \ldots & T_{iz} & T_{ij} & T_{ij} & \ldots & T_{ij} & T_{i\mathfrak{E}}
\end{array}\right]
\end{array}
$$

Figure 1.2: The transition matrix of an HMM with silent states has to incorporate transitions which do not emit a character. These transitions are shown in the upper left quadrant with subscripts wz.

Silent states have been used successfully by BALDI *et al.* (1994) and KROGH *et al.* (1994) in the design of HMM profilers to identify, for example, ho-

mologues for the globin and kinase families of protein sequences. In this type of HMM modelling, silent states were found to increase the likelihood in the presence of deletes in the amino acids sequence. In Chapter 3, I use one silent state in my HMM modelling.

1.1.6 Optimising the Likelihood

The Baum-Welch algorithm has two main disadvantages. First, the method is known to take long to converge (DEMPSTER *et al.*, 1977). Methods to obtain better convergence times have been developed, as described for example by JAMSHIDIAN and JENNRICH (1997). Nevertheless, it is preferable to use a method which can deal directly with convergence when working with many alignments and hypotheses testing. Second, a Baum-Welch maximisation of a single pairwise alignment would typically require several re-runs with different starting values in order to increase its chance of finding the global optimum (see for example, LAAN *et al.* (2006) and CHUONG and BATZOGLOU (2008)). Repeating maximisation runs for a large number of pairwise alignments individually on a mainframe computer is impractical. A possible workaround would be to use training data, consisting of "curated" pairwise alignments, to carry out pilot runs on suitable samples until high likelihood values can be detected. The corresponding estimators would then be used as starting values for the individual alignments. But this procedure would have been time costly because my experimental samples of sequence pairs were drawn from across a wide range of pairwise evolutionary distances. Furthermore, "curated" alignments of non-homologous pairs are not guaranteed to be "good" alignments since, by definition, an alignment is unobservable and hence a random variable. I needed, therefore, a method which has relatively short convergence times and which does not depend critically on initial values.

For these reasons, the simulated annealing method proposed by GOFFE *et al.* (1994) for maximising the likelihood function was more practical for my purpose. The tenet of this method is that for a given position in the search space, it considers several neighbouring possibilities. The higher the "temperature" the larger is the number of these possibilities. If none of these possibilities offer a better value,

an inferior one can be chosen in accordance with some random distribution. This means that, unlike the Baum-Welch, this algorithm does not exclude the possibility of going downhill temporarily before proceeding uphill once again towards the global optimum. Through this lateral feature, simulated annealing avoids getting "stuck" in one of the local optima. As the system "cools", so the pool of alternatives reduces, and thus the probability of being forced to choose an inferior alternative keeps decreasing. As the limit of cooling is reached, however, it is hoped that the search is very close to the global optimum. Furthermore, the final outcome is largely independent of the starting values. This feature was very important in my work which involved sequences from many different species and constructed from the three types of biological encodings. In addition, my modelling involved several parameters. These factors adversely affect computational time, and hence it was essential that maximisation could be achieved with single runs that required starting values which I could readily judge as plausible.

Although simulated annealing also does not guarantee a global optimum, it has been shown by GOFFE *et al.* (1994) that it offers excellent prospects under various types of modelling that require large numbers of parameters. They tested the method, for example, on a neural network function with ten hidden parameters. Although it failed to find the global optimum in this very difficult case, it convincingly outperformed competing conventional algorithms, namely, the simplex, quasi-Newton and conjugate gradient methods.

Simulated annealing has its roots in statistical physics. In their seminal paper, METROPOLIS *et al.* (1953) showed how a physical system of rigid spheres in two-dimensional space can attain optimal energy through a series of stochastic moves. The number of spheres is finite, and they are initially placed randomly in the X, Y plane. Spheres are then moved one at a time through a stochastic distance from $(X,\ Y)$ to $(X + \alpha \cdot \xi_x,\ Y + \alpha \cdot \xi_y)$, where α is some arbitrary constant and $\xi_x,\ \xi_y \sim U[-1, 1]$. After each move, the change in the energy of the system, namely, ΔE is measured. If this change is negative – a minimisation problem, in this case – the move is accepted. Otherwise, the move is accepted only with probability given by $e^{-\Delta E/kT}$. If this trial is unsuccessful, then no move is made. T is the temperature

of the system, which means that the probability of making a move away from the optimum will keep decreasing as T decreases. It was shown that this procedure is *ergodic*. That is, irrespective of the initial configuration, every possible configuration of the spheres can potentially be visited until the minimum level of energy is attained before the system is allowed to cool. Ergodicity implies that initial values do not, in theory, have bearing on the final outcome although they affect the length of time needed to find the optimum.

In the context of the pairwise alignment, and of numerical analyses in general, the terms "temperature" and "cooling" are, of course, only notional. In the actual implementation, the components of the algorithm consist of

1. the vector X of the values of the n parameters to be estimated, which only require some sensible initialisation at the start of the algorithm,

2. the function $f(X)$ definition which enables the algorithm to evaluate the likelihood value at each step,

3. the vector V of the current step lengths of each element in X,

4. the variable T, which holds the current "temperature", and

5. the random variable $r \sim U[-1, 1]$.

The algorithm starts by evaluating the function $f(X_0)$ using the initial values of the elements in X, and both $f(X_0)$ and X_0 are stored in the optimum register. It then changes x_i, $i = 1, 2, \ldots, n$. For each i, $x_i' = x_i + r \cdot v_i$ and $f(X')$ are computed. Each time $f(X') > f(X)$, X' becomes the new X in the optimum register, and we think of the algorithm as moving "uphill". If $f(X') \leq f(X)$, the probability $p = e^{\frac{1}{T}(f'-f)}$ is computed. Note that this probability is based on $p = e^{-\Delta E/kT}$ that was used in the Metropolis experiment, where k is the Boltzmann factor. In the case of my maximisation problem, this factor is not needed, and the analog to a change in energy E is my change in the value of the likelihood function f. It is this physical fundamental law that makes the simulated annealing algorithm highly functional and reliable when coded and used in numerical analyses.

The probability p is used to conduct the Metropolis trial. If the trial is successful, once again both $f(X)$ and X are updated in the optimum register, but this time we think of the algorithm as moving "downhill". If the trial fails, the algorithm simply stays put and moves on to the next step.

The right strategy on how T and V are managed by the experimenter are key to a successful optimisation. As a guide, T is reduced gradually in accordance with a simple stochastic relation, namely, $T' = \tau \cdot T$, where $\tau \sim U[0,1]$, and V is adjusted so that trial failures increase at an increasing rate as the algorithm gets closer to the global optimum. This makes sense since the closer we get to the global optimum, the less we want to go downhill and the more we are willing to get "stuck" so to speak, rather then move past the optimum. The terminating criterion is also managed by the experimenter. We need to make sure that this criterion is strict enough so that the algorithm does not stop prematurely without reaching the optimum. At the same time, if it is too strict we run the risk of failing to stop even when the algorithm is "close enough" to the optimum. A conservative rule would be to store the most recent three or four f values in the optimum register, and if they do not differ by a pre-specified amount ϵ, the algorithm stops and declares the final optimum along with the estimators in the vector X. In my implementation, default values were pre-set for X_0, T, V, τ, and ϵ across all optimisation runs. This ensured the elimination of subjectivity in my experiments.

GOFFE *et al.* (1994) found that the algorithm has several attributes. Two of these turn out to be particularly relevant to HMMs. First, the function $f(X)$ does not have to be differentiable. This is significant because in HMM modelling, as I discussed earlier in Section 1.1.4, the likelihood function is not analytically amenable. Second, simulated annealing can also deal effectively with rough surfaces. It is true that the forward algorithm provides a smooth surface because it sums the preceding probabilities at each iteration and across all possible alignments when aligning two sequences. However, the HMM models which I describe in later chapters use several emitting states. This makes more severe the problem of local maxima (DURBIN *et al.*, 1998, p. 63), and the smoothness of the surface of the likelihood function, therefore, is not necessarily ideal.

CHAPTER 2
The One-Region Model

2.1 The Probability Matrix from Blocks Model

For the purpose of aligning protein sequences, using standard applications such as BLAST, Fasta, and ClustalW, the PAM series (DAYHOFF *et al.*, 1978) and the BLOSUM series (HENIKOFF and HENIKOFF, 1992) have traditionally proved very useful. Nevertheless, these are score matrices restricted to classes of proteins that satisfy some specified percentage identity criteria. They also do not provide reliability measures in cases where the proteins under study do not belong to the body of data from which the matrices had been derived. Furthermore, when we compare biological sequences, we need to do more than simply compute a score if we wish to tease out hidden processes that underlie evolutionary change. For example, a more useful model would take into account both the multiple substitutions at individual sites and the complex processes of inserts and deletes (indels).

Biologists need an evolutionary model that can be adapted to any given pair of protein sequences. For this purpose, VEERASSAMY *et al.* (2003) developed the protein replacement model displayed in Appendix B.2. This model is highly versatile because it can be applied across a wide range of evolutionary time. Equally important, however, is the fact that it can also be coupled, as I show later in this chapter, to an HMM in order to allow the investigator deal stochastically with indels. Here I present a sketch of how these authors utilised the Blocks databases of BLOSUM in order to derive the probability matrix derived from blocks (PMB) of proteins that share a specified percentage c of homology, where $c \in C$, $C = \{0, 30, 32, \ldots, 98, 100\}$.

2.1.1 PMB Construction

To construct the PMB model, one starts with a BLOSUM frequency count matrix F that has a clustering percentage taken from the set C. For a specified c,

the observed amino acid frequencies π_i $(i = 1, 2, \ldots, 20)$ are computed from

$$\pi_i^{(c)} = \sum_{j=1}^{20} F_{ij}^{(c)} \Big/ \sum_{\substack{u=1 \\ v=1}}^{20} F_{uv}^{(c)},$$

and each mutation matrix $M^{(c)}$ is computed from

$$M_{ij}^{(c)} = F_{ij}^{(c)} \Big/ \sum_{k=1}^{20} F_{ik}^{(c)}.$$

A mutation matrix models an evolutionary process, and hence can be expressed as a function of the evolutionary time t. One can therefore denote such a matrix as $M^{(c)}(t)$, where t is unobservable, and one would want to estimate this parameter for each data block corresponding to c. To do so, one proceeds by initially expressing the average substitution rate as a function of the mutation matrix *with t held fixed* as follows

$$f\big(M^{(c)}(t)\big) = 1 - \sum_{i=1}^{20} \pi_i^{(c)} M_{ii}^{(c)}(t). \tag{2.1}$$

One can now appeal to Taylor's series expansions to derive an equation that would allow the first derivative of $f(M^{(c)}(t))$ in 2.1 to be estimated numerically. I present the derivation in Appendix B.1, and I write the equation for the present context (after applying the Chapman-Kolmogorov relation to each term in the summation) as follows

$$tf'\left(M^{(c)}(t)\right) = \frac{25}{3}\left[\sum_{\substack{\nu=-2 \\ \nu\neq 0}}^{2} \left[(-1)^{\nu}\frac{-2}{\nu}\right]^3 f\left(\left[M^{(c)}(t)\right]^{1.0+0.01\nu}\right)\right]. \tag{2.2}$$

2.1 and 2.2 can now be used to compute $tf'\left(M^{(c)}(t)\right)$ for every value in C, and when the results are plotted, they are found to have a quadratic form as shown in Figure 2.1.

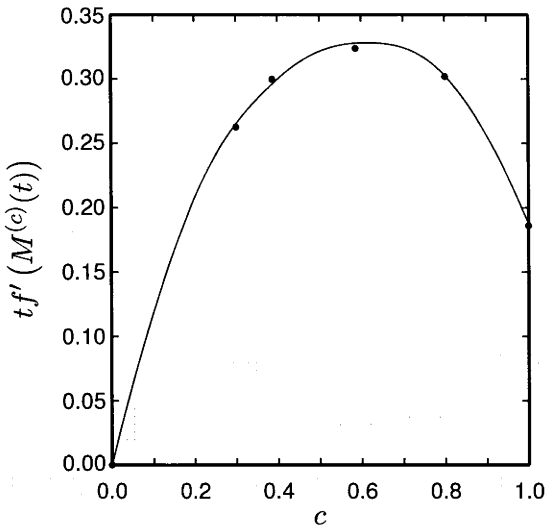


Figure 2.1: The plot of the numerical expression $tf'\left(M^{(c)}(t)\right)$ versus the elements in c turns out to be a quadratic curve. (Note that only elements $0\%, 30\%, 40\%, 60\%, 80\%$ and 100% are plotted here in order to simplify the diagram.)

The trick now is to run a standard linear regression of the form

$$\boldsymbol{y} = a_1\boldsymbol{x}^2 + a_2\boldsymbol{x} + a_3 + \boldsymbol{\epsilon}, \;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mu, \sigma^2). \tag{2.3}$$

In 2.3, $\boldsymbol{y}$ is the dependent vector, and represents values obtained from 2.2, while $\boldsymbol{x}$ is the independent vector, and represents values obtained from 2.1, with all values being computed for each element in C. The error vector $\boldsymbol{\epsilon}$ in 2.3 is assumed to be normally distributed, although the authors do not report the usual statistical analysis on $\boldsymbol{\epsilon}$. They report, however, that R^2 is sufficiently high to warrant a good fit.

Having estimated the coefficients for 2.3, it now remains to solve a separable differential equation of the form

$$\hat{t}\,\frac{dz}{d\hat{t}} = a_1z^2 + a_2z + a_3,$$

where $z = f\left(M^{(c)}(t)\right)$. The estimated evolutionary time $\hat{t}^{(c)}$, corresponding to every element in C, can then be computed.

2.1.1.1 The Stationary Markov Model

Equation (16) in VEERASSAMY *et al.* (2003) is the *stationary Markov model.* HASEGAWA *et al.* (1985) provide a description of this model using the DNA alphabet. The principles behind this model are as follows.

Each amino acid (or some other character such as a nucleotide) can be seen to be evolving according to a Markov process. That is, a letter i from the alphabet is replaced by another letter j from the same alphabet with probability $P_{ij}(t)$, where $P(t)$ is some row stochastic matrix. What we want to model, here, is the time needed for this replacement to take place.

One starts with some *substitution model* that takes the form of a square matrix R. The elements of this matrix are pre-defined. The well-known Jukes-Cantor $R^{(JC)}$, for example, is defined as

$$R^{(JC)} = \begin{array}{c} \\ \mathtt{A} \\ \mathtt{B} \\ \mathtt{C} \\ \mathtt{D} \end{array} \begin{array}{c} \begin{array}{cccc} \mathtt{A} & \mathtt{B} & \mathtt{C} & \mathtt{D} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array},$$

with a four-letter alphabet suitable for DNA data, and we can extend it to a twenty-letter alphabet to deal with amino-acids. This definition means that as we try to model the substitution (or replacement) rate, we assume that when some letter is replaced by some other letter, substitutions are all equally likely. This, of course, is a naive scheme, but it is useful as a starting point. Biologists introduce parameters to allow certain substitutions to occur more (or less) frequently relative to other substitutions. For example, in the definition

$$R(\alpha) = \begin{array}{c} \\ \mathtt{A} \\ \mathtt{B} \\ \mathtt{C} \\ \mathtt{D} \end{array} \begin{array}{c} \begin{array}{cccc} \mathtt{A} & \mathtt{B} & \mathtt{C} & \mathtt{D} \end{array} \\ \begin{bmatrix} 0 & 1 & \alpha & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array},$$

we introduce a parameter which would allow us to measure the rate of change from the letter A to the letter C relative to all other changes. If this parameter is smaller than one, then we infer that this substitution is less likely to occur than other substitutions, with everything else being equal. By imposing the restriction $\alpha = 1$, and calling this restriction the *null hypothesis*, we can test the null against some

alternative hypothesis of interest, say, $\alpha < 1$. If the evidence that emerges from an experiment cannot reject the null, we would then infer that for the given body of data, $R(\alpha)$ is statistically the same as $R^{(JC)}$.

RODRIGUEZ *et al.* (1990) present a number of substitution models for the nucleotide alphabet. One thing to note about these models is that some are symmetrical while others are not. Throughout this work, I choose models only from among those that are symmetrical, commonly known as *reversible models* (e.g. ISAEV, 2004, p. 126). Symmetry is a strong assumption, but for a typical data set of biological sequences used in this work it generally holds well, that is, the model proves to be robust and yielding reliable results.

My models also meet the four criteria of RODRIGUEZ *et al.* (1990), namely, that a substitution rate is (a) site independent, (b) constant over time, (c) the same for the two sequences in the pairwise alignment, and (d) a process set against a background of letter frequencies which is computed from the two sequences and which is the same as that of the distant ancestor.

To do the computation in (d) I use

$$q_i = 2x(t)_{ii} + \sum_{j=1}^{n} \left[x(t)_{ij} + x(t)_{ji} \right], \; i = 1, 2, \ldots, n, \tag{2.4}$$

(RODRIGUEZ *et al.*, 1990), where n is the size of the alphabet and $x(t)_{ij}$ is an element of the divergence matrix $X(t)$. This matrix, however, is constructed from the pairwise alignment which is unobservable. Hence, the best I can do to compute q_i is to do a simple count of each letter in the two sequences to construct $X(t)$.

In theory, an element in $X(t)$ gives a count of how many times letter $i \in \xi$, $\xi =$ some alphabet, matches (or mismatches) with letter $j \in \xi$ in the pairwise alignment at time t, bearing in mind that if we were to perform the alignment, say, one million years ago (that is, at $t - 10^6$) we would expect it to be different from what it would be today at time t measured in years. It is easy to see that at $t = 0$, $x(0)_{ij} = q_i$ if $i = j$, and is equal to zero otherwise. Thus $X(0) = \left(\boldsymbol{q}_n \otimes \mathbf{1}_n' \right) \bullet I_{n \times n}$, where (1) $\boldsymbol{q}_n$ is the vector of background frequency counts whose n elements are usually normalised so that they sum to one and are then called the *background probabilities*; (2) the

vector $\mathbf{1}_n$ has all n elements equal to 1 and is first transposed before it is multiplied with $\boldsymbol{q}_n$; and (3) $(\otimes)$ and $(\bullet)$ are the Kronecker and the Hadamard products, respectively. A final note about $X(t)$ is that in VEERASSAMY *et al.* (2003) it is called the *substitution frequency count matrix* F. This is because these workers are constructing an approximation, or some "guess", of $X(t)$ using the Blocks databases for a specified element in C. This element is a discrete value that corresponds to some unobservable evolutionary time.

Two other important matrices are (1) the matrix of substitution rates Q and (2) the evolutionary matrix $P(t)$. RODRIGUEZ *et al.* (1990) define Q as *the matrix whose elements are transient intensity functions of the stochastic process.* Transient implies that Q is not a function of evolutionary time, and for this reason it is also commonly called the *instantaneous rate matrix.* For the purpose of my experiments, I implement Q in accordance with the following definition.

$$\text{define}: \quad Q = R_{n\times n}\Big((\boldsymbol{q}_n \otimes \mathbf{1}_n')\bullet I_{n\times n}\Big) - I_{n\times n}, \tag{2.5}$$

$$\text{subject to}: \quad \sum_{i=1}^{n} q_i = 1, \tag{1}$$

$$\big[Q\big]_{ii} = -1, \ \text{for } i = 1, 2, \cdots, n, \ \text{and} \tag{2}$$

$$\sum_{j=1}^{n} \big[Q\big]_{ij} = 0, \ \text{for } i = 1, 2, \cdots, n, \tag{3}$$

where $\boldsymbol{q}_n$ is the vector of background probabilities with $n = 4,\ 20,$ or 61, depending on whether the alphabet is taken from biological encodings (BEs) DNA, protein, or codons, respectively. It is easy to see that Q is a function of the data – owing to the presence of $\boldsymbol{q}$ – and of the pre-definition of R, and is not dependent on evolutionary time. Constraints (1), (2) and (3) ensure that background probabilities sum to one, and that elements in each row of the product of R and $\boldsymbol{q}$ also sum to one before the I matrix is subtracted so that elements in each row of Q sum to zero.

To illustrate how Q is implemented, I shall use a simple numerical example.

Let the substitution model be $R^{(JC)}$, that is, the Jukes-Cantor which I showed earlier. Also, let $\boldsymbol{q} = (0.40, 0.30, 0.14, 0.16)$. Substitution into 2.5 gives Q as

$$Q = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \left(\left(\begin{bmatrix} 0.40 \\ 0.30 \\ 0.14 \\ 0.16 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \right) \bullet \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

$$- \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -1.000 & 0.500 & 0.233 & 0.267 \\ 0.571 & -1.000 & 0.200 & 0.229 \\ 0.465 & 0.349 & -1.000 & 0.186 \\ 0.476 & 0.357 & 0.167 & -1.000 \end{bmatrix}$$

(Cell entries for the computed Q are rounded to three decimal places for the purpose of this illustration.)

Now I deal with the matrix $P(t)$, and to do so I appeal to HASEGAWA *et al.* (1985). The probability $P_{ij}(t)$ is designed to capture the event in evolution whereby a letter i changes to become letter j over some amount of evolutionary time Δt, after evolution had been taking place over an arbitrary time t following the start of divergence. Since the matrix Q represents times that are *transient*, we can represent these Markovian transitions as

$$P(\Delta t) = I + Q\Delta t, \tag{2.6}$$

$$P(t + \Delta t) = P(t)(I + Q\Delta t), \tag{2.7}$$

$$\lim_{\Delta t \to 0} \left\{ \frac{P(t + \Delta t) - P(t)}{\Delta t} \right\} = \lim_{\Delta t \to 0} \left\{ \frac{P(t)Q\Delta t}{\Delta t} \right\}, \tag{2.8}$$

$$\frac{dP(t)}{dt} = P(t)Q,$$

$$\int \frac{1}{P(t)} dP(t) = \int Q dt + K,$$

$$Qt = \log_e P(t) + K', \tag{2.9}$$

$$P(t) = e^{Qt}. \tag{2.10}$$

2.7 is an application of the Chapman-Kolmogorov equation. 2.8 is the standard form of the limiting theory of differential calculus. In 2.9, it should be easy to see that constant K' is zero since $P(0) = I$. 2.10 is equivalent to $M^{(c)} = e^{Q\hat{t}^{(c)}}$ in VEERASSAMY *et al.* (2003), although these authors use a slightly different notation, and they call $M^{(c)}$ the mutability matrix.

To avoid notational ambiguities, from now on I shall use the following notation:

1. R : substitution (or replacement[3]) model matrix,
2. $X(t)$: divergence matrix at time t,
3. Q : instantaneous rate matrix as a function of R and of $\boldsymbol{q}$,
4. $P(t)$: evolutionary matrix at time t.

2.1.1.2 The Protein Replacement Model

In Section 2.1.1, I showed how VEERASSAMY *et al.* (2003) constructed the estimator $\hat{t}^{(c)}$ for any of the elements in the set C. Substituting these estimators into 2.10 leads to a corresponding set of Q matrices, where $Q^{(c)} = \log_e M^{(c)}/\hat{t}^{(c)}$. To obtain a "universal" Q matrix for proteins, one can solve the non-linear program

[3]By convention, we use the term *substitution* for DNA and codon biological encodings (BEs) and the term *replacement* for protein BE.

$$\text{minimise}: \quad \sum_c \frac{\|e^{Q\hat{t}^{(c)}} - M^{(c)}\|}{\|M^{(c)}\|},$$

$$\text{subject to}: \quad Q = \sum_c w^{(c)} Q^{(c)}, \tag{1}$$

$$\sum_c w^{(c)} = 1, \tag{2}$$

$$0 \leqslant w^{(c)} \leqslant 1. \tag{3}$$

The R matrix of VEERASSAMY *et al.* (2003) is displayed in Appendix B.2, with each element entered to four decimal places. Note that this matrix is derived from 2.5 after the optimised Q had been computed. This R matrix is practical during implementation because unlike score matrices I can easily incorporate it as a "plug-in" within my modelling, along with other R matrices.

2.2 The Goldman-Yang Model

The PMB instantaneous rate matrix $Q^{(PMB)}$ uses the amino acid as its unit of data. It is also constructed from the Blocks databases (HENIKOFF and HENIKOFF, 1992), in which a large amount of evolutionary information is summarised. This means that $Q^{(PMB)}$ can deal with sequence pairs that have a wide range of evolutionary times. In Chapter 3, I show how I constructed a protein data set by random sampling of curated alignments stored in the BAliBASE database (THOMPSON *et al.*, 1999b) using $Q^{(PMB)}$. These pairs have relative evolutionary times (measured as the average number of replacements per amino acid) that vary widely between 0.25 to 1.25.

The $Q^{(PMB)}$ matrix, however, does not account for several biological factors, namely, (1) the dependence of intra-codon nucleotides, (2) the difference among the substitution rates of intra-codon nucleotides, (3) the transition-transversion rate ratio, and (4) the nonsynonymous-synonymous rate ratio.

$$Q_{ij}^{(GY94)} = \begin{cases} 0 & \text{for two codons that differ at more than one position} \\ q_j & \text{for synonymous transversion} \\ \kappa q_j & \text{for synonymous transition} \\ \omega q_j & \text{for nonsynonymous transversion} \\ \omega\kappa q_j & \text{for nonsynonymous transition} \end{cases} \tag{2.11}$$

To investigate phylogenetic trees, where selective constraints among lineages at the molecular level are of particular interest, GOLDMAN and YANG (1994) and YANG (1998) developed the codon-based model $Q^{(GY94)}$ shown in Equation 2.11.

While $Q^{(PMB)}$ is an empirical model, $Q^{(GY94)}$ is a mechanistic model with two parameters (not including the $\boldsymbol{q}$ vector of background probabilities). One of these is the parameter κ which is designed to capture information on the ratio between the rate of transitions and the rate of transversions that accumulate between two species experiencing increasing divergence over evolutionary time. The other parameter is ω whose role requires an understanding of the genetic code.

2.2.1 The Standard Genetic Code

TTT	→	Phe	TCT	→	Ser	TAT	→	Tyr	TGT	→	Cys
TTC	→	Phe	TCC	→	Ser	TAC	→	Tyr	TGC	→	Cys
TTA	→	Leu	TCA	→	Ser	TAA	→	Stp	TGA	→	Stp
TTG	→	Leu	TCG	→	Ser	TAG	→	Stp	TGG	→	Trp
CTT	→	Leu	CCT	→	Pro	CAT	→	His	CGT	→	Arg
CTC	→	Leu	CCC	→	Pro	CAC	→	His	CGC	→	Arg
CTA	→	Leu	CCA	→	Pro	CAA	→	Gln	CGA	→	Arg
CTG	→	Leu	CCG	→	Pro	CAG	→	Gln	CGG	→	Arg
ATT	→	Ile	ACT	→	Thr	AAT	→	Asn	AGT	→	Ser
ATC	→	Ile	ACC	→	Thr	AAC	→	Asn	AGC	→	Ser
ATA	→	Ile	ACA	→	Thr	AAA	→	Lys	AGA	→	Arg
ATG	→	Met	ACG	→	Thr	AAG	→	Lys	AGG	→	Arg
GTT	→	Val	GCT	→	Ala	GAT	→	Asp	GGT	→	Gly
GTC	→	Val	GCC	→	Ala	GAC	→	Asp	GGC	→	Gly
GTA	→	Val	GCA	→	Ala	GAA	→	Glu	GGA	→	Gly
GTG	→	Val	GCG	→	Ala	GAG	→	Glu	GGG	→	Gly

Table 2.1: Illustrating the redundancy property of the Standard Genetic Code.

Table 2.1 shows the Standard Genetic Code as tabulated by the NCBI web site. Several other codes are also tabulated by NCBI, but in the present work only the standard code will be used.

The first thing to note about this table is that no matter which three letter combination we choose from a total of 64, the combination will always code for one of the 20 amino acids, with three exceptions, namely, TAA, TAG and TGA. These three are stop codons which, if present somewhere along the protein and not at the end, they would produce a truncated protein. Such a truncation would most likely be selectively deleterious. For this reason, a codon substitution model would, in general, impose the constraint that stop codons do not occur in a protein.

The next thing to note is that very often when we change the third letter, the new codon still codes for the same amino acid. For example, when we take a codon that codes for valine (Val), no matter how we change the third letter, the new codon still keeps coding for valine so long as we keep the first two letters intact. This redundancy is also true to a small extent for the first letter, but it is never true for the middle letter. If we were to assume that nucleotides are equally distributed across the genome, and that nucleotide substitutions occur randomly under a uniform distribution, we could then construct Table 2.2.

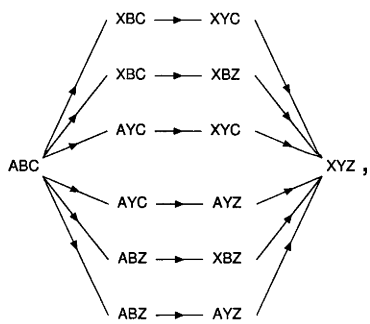
Position	# Synonymous	# Nonsynonymous	% Synonymous	% Nonsynonymous
1	8	166	4.6	95.4
2	0	176	0	100
3	126	50	72	28

Table 2.2: Variation of synonymous and nonsynonymous counts at the three codon positions.

From this table we can see that on average no amino acid replacement results from 8 possible changes in the first position, but for the third position this number rises to 126. It is abundantly clear that we would expect the rate of nucleotide substitution to be highly variable among these two positions, and it is for this reason that the ω parameter plays an important role.

Before the ω parameter was introduced, NEI and GOJOBORI (1986) and other workers had developed approximate methods to model codon substitutions. The

NEI and GOJOBORI (1986) method is centred on *counting* the number of synonymous sites S, and the number of synonymous differences S_d, in a codons pairwise alignment with gaps removed. S is a summation of constants for each codon in the alignment. For example, it is easy to see from Table 2.1 that codon TTA which codes for leucine (Leu) can change synonymously in only two different ways. To compute S_d, first we have to determine by how many nucleotide sites each codon pair of the alignment differ, that is, $0, 1, 2$, or 3. If the difference d is zero, then S_{dm} is zero for codon pair m (where $m = 1, 2, \ldots, M$ and M is the length of the alignment) since no change occurred. If d is one, then S_{dm} is one if the change is synonymous, and is zero otherwise. If d is two, then we would have two possible pathways and a different transit codon in each path. In this case, S_{dm} is the total of synonymous changes from a total of four possible changes multiplied by $\frac{1}{2}$ (assuming that pathways have equal probability). When the difference is three, we would then enumerate the six possible pathways with six different transit codons as shown in the following illustration using fictitious codons,



to similarly compute S_{dm}. S_d is then determined by summing over m. The number of nonsynonymous sites N and nonsynonymous differences N_d are computed in a complementary way. Two statistics could then be estimated using the standard Jukes-Cantor formula (which takes into account multiple hits per site), that is, $d_S = -\frac{3}{4}\log_e\left(1 - \frac{4}{3}\frac{S_d}{S}\right)$ and $d_N = -\frac{3}{4}\log_e\left(1 - \frac{4}{3}\frac{N_d}{N}\right)$ (NEI and GOJOBORI, 1986). I show estimates for d_S and d_N in the first two rows of Table 2.3 for ten alignments of ten coding DNA sequence pairs randomly selected from BAliBASE for the purpose of this exercise.

The method of NEI and GOJOBORI (1986) provides me with a precursor of

Aln Num	1	2	3	4	5	6	7	8	9	10
d_S	∞	∞	1.179	∞	∞	2.737	∞	1.783	0.676	∞
d_N	0.265	0.256	0.185	0.307	0.270	0.198	0.205	0.147	0.205	0.339
$\hat{t}_{(H_o)}$	0.257	0.236	0.172	0.275	0.269	0.202	0.198	0.165	0.155	0.294
$\hat{t}_{(H_{a_1})}$	0.263	0.251	0.172	0.283	0.270	0.207	0.204	0.164	0.156	0.293
$\hat{\kappa}_{(H_{a_1})}$	2.511	3.614	1.824	2.302	1.364	2.853	2.938	1.538	2.055	1.111
$p-value_{(H_{a_1})}$	0.001	0.000	0.070	0.009	0.406	0.000	0.000	0.440	0.014	0.810
$\hat{t}_{(H_{a_2})}$	0.387	0.318	0.187	0.408	0.502	0.313	0.313	0.202	0.169	0.415
$\hat{\omega}_{(H_{a_2})}$	0.069	0.020	0.069	0.032	0.039	0.028	0.019	0.035	0.069	0.018
$p-value_{(H_{a_2})}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 2.3: d_S values, which depend on the numbers of synonymous substitutions, are chaotic. This shows that it is very difficult to measure synonymous substitutions with the method of NEI and GOJOBORI (1986). d_N values, which depend on the numbers of nonsynonymous substitutions are stable and amenable to estimation. Evolutionary time is captured by $\hat{t}$ using the GOLDMAN and YANG (1994) model, and is highly comparable to d_N. $\hat{\kappa}$ is significant *only* when it is greater than two, and does not impact on $\hat{t}$. Contrariwise, $\hat{\omega}$ clearly shows signs of *innovation*, impacting greatly on $\hat{t}$ in most of the alignments.

how the two distinct substitutions differ from each other. On the one hand, nonsynonymous substitutions shown in row two are markedly stable. They also compare very well with the ML estimators shown in row three which are the average substitution rates computed using the GY94 model. The inference is that when we measure the average substitution rate in a pairwise alignment, what we could be measuring are those molecular changes that are well controlled (or constrained) by phenotypic dependencies, and hence also by natural selection. On the other hand, synonymous substitutions shown in row one are disparate, and most of them remain unmeasurable by this method. These type of substitutions do not generate amino acid replacements, and hence it appears that their behaviour is "erratic".

Questions that I posit here are, how do I develop a method that could give me better measurements of these erratic changes? Would it be possible to locate them positionally along the alignment and see how they cluster? If I could achieve this differentiation with statistical significance, would I then be able to estimate the transition-transvertion rate ratio κ and the nonsynonymous-synonymous rate ratio ω parameters within these non-conserved regions? How would these estimators differ from corresponding estimators within conserved regions?

Traditionally, the substitution rate has been studied as an average across the alignment. If we were to analyse the pairwise alignment purely as a biological

device and not as an exercise in letter patterning, then we should incorporate secondary structure in our methods. Endogenously extracting knowledge on secondary structure can help us identify regions along the alignment that exhibit significantly higher substitution rates. In Chapter 3, I propose a two-region model that attempts to address these questions.

2.2.1.1 The Codon Substitution Model

The GY94 model is described in detail by YANG and NIELSEN (2000) and in other places, (see for example GOLDMAN and YANG (1994) and YANG (1998)). This model is parametric and requires the Maximum Likelihood (ML) method. It consists of a 61×61 instantaneous rate matrix Q (2.11). The diagonal entries of Q are scaled so that rows sum to zero, and

$$-\sum_{i=1}^{61} q_i Q_{ii} = \sum_{\substack{j=1 \\ i \neq j}}^{61} q_i Q_{ij} = 1.$$

This formulation ensures that evolutionary time (captured by the estimator $\hat{t}$) is measured as *the expected number of nucleotide substitutions per codon.* What this means is that we have to divide $\hat{t}$ by three when we compare rates with those obtained using the PMB model.

In order to compute the ML, we require an explicit function that we can maximise using a numerical method such as simulated annealing. This function is defined as

$$\log \mathcal{L}(\boldsymbol{\theta}|\mathcal{A}) = \sum_{i,j=1}^{61} \log_e \left[q_i P_{ij}(t) \right], \tag{2.12}$$

where $\mathcal{A}$ is the pairwise alignment, and $P(t)$ is the evolutionary matrix at time t. $\boldsymbol{\theta}$ is the vector of parameters (t, κ, ω) that we want to maximise. For experimental purposes, we may not want to maximise all the parameters at once. Hence, I employ a binary vector, namely, $\mathfrak{b} = (\mathfrak{b}_1, \mathfrak{b}_2, \mathfrak{b}_3)$ which allows me to select which parameters I want to keep fixed during optimisation in accordance with some specified hypothesis.

Hence, to compute the estimates shown in row three of Table 2.3 under the

null hypothesis, I set $\mathfrak{b} = (1, 0, 0)$. This means that the optimisation routine will vary t only while keeping the other two parameters fixed to 1.0 during its search for the global optimum. For rows four and five $\mathfrak{b}$ is set to $(1, 1, 0)$, and for rows six and seven to $(1, 0, 1)$, under alternative hypotheses one and two, respectively. To test for significance of $\hat{\kappa}$ and $\hat{\omega}$, the χ^2 distribution is used with one degree of freedom.

From Table 2.3 we can see that the mutation parameter κ gains in significance as it rises above 2.0 without, however, greatly affecting evolutionary time. On the other hand, the selection parameter ω is always highly significant even though it is very small across these ten alignments. At the same time, it impacts greatly on evolutionary time, indicating that we can view ω as an *innovation* parameter.

One remaining issue to be raised here is that $\boldsymbol{\theta}$ is estimated in 2.12 with $\mathcal{A}$ as given. This is too common in the literature. My contention is that a thorough analysis of a pairwise alignment should not assume that the given alignment is *ex cathedra*. What I mean is that the alignment itself is unobservable, that is, it too is a stochastic quantity and should, therefore, be part of the maximisation process using ML. In my formulation of the one-region model later in this chapter I show how alignment optimisation is incorporated in the ML procedure.

2.2.1.2 Computing $\hat{d}_S$ and $\hat{d}_N$

It remains to show how we can use the GY94 model to obtain ML estimators $\hat{d}_S$ and $\hat{d}_N$, and then compare these with corresponding estimators obtained earlier using the method of NEI and GOJOBORI (1986).

The vectors $(\hat{t}, \hat{\kappa}, \omega = 1)$ for all the ten alignments are shown in rows four and five of Table 2.3. These are estimators obtained *before* natural selection had a chance to operate at the amino acid level (YANG and NIELSEN, 2000). We also require the corresponding vectors $(\hat{t}^*, \hat{\kappa}^*, \hat{\omega}^*)$ which are shown in Table 2.4. These estimators capture the effects of the *innovation* property of ω.

We can use (e.g. YANG and NIELSEN, 2000, p. 34)

$$\hat{d}_S = \frac{\hat{t}^* \sum_{\substack{i,j=1 \\ i \neq j}}^{61} \delta(i,j) q_i Q_{ij}(\hat{\kappa}^*, \omega^*)}{3 \sum_{\substack{i,j=1 \\ i \neq j}}^{61} \delta(i,j) q_i Q_{ij}(\hat{\kappa}, \omega = 1)}, \tag{2.13}$$

where $\delta(i,j)$ is one if (i,j) yield a synonymous change, and is zero otherwise. $\hat{d}_N$ can be computed likewise after reversing the value of $\delta(i,j)$.

Aln Num	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
d_S	∞	∞	1.179	∞	∞	2.737	∞	1.783	0.676	∞
d_N	0.265	0.256	0.185	0.307	0.270	0.198	0.205	0.147	0.205	0.339
d_N/d_S	0.000	0.000	0.157	0.000	0.000	0.072	0.000	0.082	0.303	0.000
$\hat{t}^*_{(H_{a_3})}$	0.405	0.316	0.187	0.414	0.515	0.316	0.313	0.203	0.181	0.414
$\hat{\kappa}^*_{(H_{a_3})}$	1.556	1.502	1.449	0.654	1.131	1.179	0.890	1.384	2.850	0.905
$\hat{\omega}^*_{(H_{a_3})}$	0.075	0.023	0.071	0.028	0.038	0.029	0.018	0.036	0.038	0.018
$p-value_{(H_{a_3})}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\hat{d}_S$	2.194	1.970	0.954	2.213	3.159	1.839	2.118	1.282	0.996	3.268
$\hat{d}_N$	0.333	0.299	0.237	0.415	0.346	0.256	0.255	0.213	0.214	0.377
$\hat{d}_N/\hat{d}_S$	0.152	0.152	0.248	0.188	0.110	0.139	0.120	0.166	0.215	0.115

Table 2.4: $\hat{d}_S$ and $\hat{d}_N$ were computed using the GY94 model. p-values were computed with two degrees of freedom.

Estimators $\hat{d}_S$ and $\hat{d}_N$ are shown in Table 2.4. They were obtained using the GY94 and maximum likelihood (ML). These estimators are asymptotically efficient under regulatory conditions owing to the ML property of *invariance* (e.g. GREENE, 1997, p. 133).

From my ten randomly selected alignments, it can be seen once more that $\hat{d}_S$ ($\hat{\sigma}_{\hat{d}_S} = 0.75$) is much more variable than $\hat{d}_N$ ($\hat{\sigma}_{\hat{d}_N} = 0.07$). Consistent with YANG and NIELSEN (2000), the ratios of $\hat{d}_N/\hat{d}_S$ are underestimated when using the NEI and GOJOBORI (1986) method, with the only exception being alignment nine. This can be attributed to the high variability of d_S. Finally, note the discrepancies between $\hat{\omega}^*$ and $\hat{d}_N/\hat{d}_S$. This reflects the fact that alignments consist of two

sequences of finite length.

2.3 The Hasegawa-Kishino-Yano Model

The substitution model matrix R in HASEGAWA *et al.* (1985) takes the following form

$$R(\alpha,\beta) = \begin{array}{c} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{A} \\ \mathrm{G} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T} & \mathrm{C} & \mathrm{A} & \mathrm{G} \end{array} \\ \begin{bmatrix} 0.0 & \alpha & \beta & \beta \\ \alpha & 0.0 & \beta & \beta \\ \beta & \beta & 0.0 & \alpha \\ \beta & \beta & \alpha & 0.0 \end{bmatrix} \end{array},$$

where α and β are non-negative parameters designed to measure the transition and transversion substitution rates, respectively. It is common, however, to implement the model with only one parameter as follows

$$R(\kappa) = \begin{array}{c} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{A} \\ \mathrm{G} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T} & \mathrm{C} & \mathrm{A} & \mathrm{G} \end{array} \\ \begin{bmatrix} 0.0 & \kappa & 1.0 & 1.0 \\ \kappa & 0.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 0.0 & \kappa \\ 1.0 & 1.0 & \kappa & 0.0 \end{bmatrix} \end{array},$$

where κ is the transition-transversion rate ratio parameter. The vector $\boldsymbol{q}$ of background probabilities for this model is estimated from the data using 2.4.

The HKY model in HASEGAWA *et al.* (1985) was designed to deal with the relation between the rate of transition and the rate of transversion in Mitochondrial DNA taken from Hominoidea and from corresponding regions of bovine and mouse. The authors identified the transversion rate as the variable which, unlike the transition rate, could explain evolutionary time regardless of codon position. They employed their method with a multiple alignment of seven sequences – with gaps removed – to estimate times of divergence at each bifurcation of the phylogenetic tree. In the following I give an outline of this method. For this purpose I assume a pairwise alignment for simplicity.

$$Q^{(HKY)} = \begin{array}{c} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{A} \\ \mathrm{G} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T} & \mathrm{C} & \mathrm{A} & \mathrm{G} \end{array} \\ \begin{bmatrix} -(\alpha q_C + \beta q_A + \beta q_G) & \alpha q_C & \beta q_A & \beta q_G \\ \alpha q_T & -(\alpha q_T + \beta q_A + \beta q_G) & \beta q_A & \beta q_G \\ \beta q_T & \beta q_C & -(\alpha q_C + \beta q_T + \beta q_C) & \alpha q_G \\ \beta q_T & \beta q_C & \alpha q_A & -(\alpha q_A + \beta q_T + \beta q_C) \end{bmatrix} \end{array}$$

Figure 2.2: The Q matrix of HASEGAWA *et al.* (1985), which I denote as $Q^{(HKY)}$.

Figure 2.2 shows the HKY rate matrix $Q^{(HKY)}$ as employed by the authors. Note that they chose to retain α and β in their formulation, and not reduce these two parameters to one, namely, κ. They did this for the reason explained earlier, and they examined the transversion rate patameter β rather than α when maximising the likelihood.

$$P(t) = \begin{array}{c} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{A} \\ \mathrm{G} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T} & \mathrm{C} & \mathrm{A} & \mathrm{G} \end{array} \\ \begin{bmatrix} 1 & q_Y^{-1} & 0 & q_C q_Y^{-1} \\ 1 & q_Y^{-1} & 0 & -q_T q_Y^{-1} \\ 1 & -q_R^{-1} & q_G q_R^{-1} & 0 \\ 1 & -q_R^{-1} & -q_A q_R^{-1} & 0 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T} & \mathrm{C} & \mathrm{A} & \mathrm{G} \end{array} \\ \begin{bmatrix} e^{t\lambda_1} & 0 & 0 & 0 \\ 0 & e^{t\lambda_2} & 0 & 0 \\ 0 & 0 & e^{t\lambda_3} & 0 \\ 0 & 0 & 0 & e^{t\lambda_4} \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T} & \mathrm{C} & \mathrm{A} & \mathrm{G} \end{array} \\ \begin{bmatrix} q_T & q_C & q_A & q_G \\ q_R q_T & q_R q_C & -q_Y q_A & -q_Y q_G \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \end{array}$$

Figure 2.3: The spectral decomposition of the right hand side of 2.6 after applying the limiting theory of calculus and replacing Q with $Q^{(HKY)}$.

It can be shown that with this formulation the exponentiation of $tQ^{(HKY)}$ can be spectrally decomposed as in Figure 2.3. These three matrices can be multiplied to obtain the matrix in Figure 2.4. It can also be shown that the λ's in this matrix are as follows: $\lambda_1 = 0$, $\lambda_2 = -\beta$, $\lambda_3 = -q_Y\beta - q_R\alpha$, and $\lambda_4 = -q_Y\alpha - q_R\beta$, where $q_Y = q_T + q_C$ and $q_R = q_A + q_G$.

The aim is to construct analytically a likelihood function that can be maximised in order to obtain the ML estimator for the parameter vector $\boldsymbol{\theta}$, given the pairwise alignment $\mathcal{A}$.

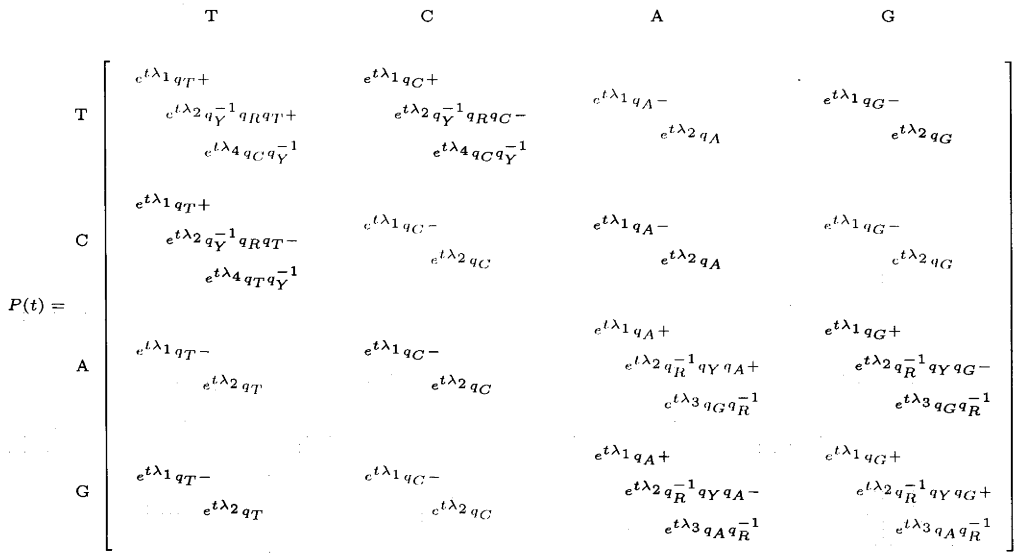
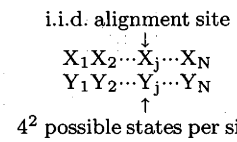


Figure 2.4: Product of the spectral decomposition of $e^{tQ^{(HKY)}}$.

Consider an alignment, with gaps removed, constructed from two sequences X and Y that belong to two species which diverged t million years ago. The alignment has length N, and can be considered large enough to allow averages to be representative of the population means. Each site $j \in \{1, 2, \ldots, N\}$ is assumed to be independently and identically distributed (i.i.d.) and having a polynomial distribution with 4^2 possible states, as illustrated below



Note that in this illustration the superscript is 2 because I am considering a pairwise alignment. Had it been an alignment with three sequences, the polynomial distribution would then have a total of 64 possible states, and so on. Denote the state at time t at site j by $x_j(t)$ in sequence X and by $y_j(t)$ in sequence Y, where x and y can take any letter of the alphabet $\xi = \{T, C, A, G\}$. It is also assumed that the process is stationary Markovian and reversible. Hence, for two extant characters

$a, b \in \{T, C, A, G\}$

$$P\big(x_j(t) = a, y_j(t) = b\big) = P\big(x_j(t) = a, y_j(t) = b \,|\, \Xi_j\big) P\big(\Xi_j\big), \;\; a \neq b,$$

$$= \nu \sum_{c \in \xi} q_c P_{ca}(t) P_{cb}(t), \;\; c \text{ is ancestral},$$

$$= \nu q_a \sum_{c \in \xi} P_{ac}(t) P_{cb}(t), \tag{2.14}$$

$$= \nu q_a P_{ab}(2t), \tag{2.15}$$

where Ξ_j means that site j mutates,
ν is the probability that a given site mutates,
2.14 is the Chapman-Kolmogorov equation, and
$\xi = \{T, C, A, G\}$.

To construct the likelihood function, we need to compute counts V_n, S_m, averages $\overline{V}(t), \overline{S}(t)$, variances σ_V^2, σ_S^2 and covariances σ_{VS}, σ_{SV} of the number of mismatches in the alignment which are transversions (V) and transitions (S). Counts are obtained using $V_n = \sum_{j=1}^{N} \delta_j(a, b)$, where $\delta_j(a, b)$ is one if the letters a, b on the site j yield a transversion, and zero otherwise. S_m is similarly computed by setting the value of $\delta_j(a, b)$ to be one if the letters a, b on the site j yield a transition.

To compute averages, we appeal to 2.15 and to the elements of the matrix in Figure 2.4. For the average number of transversions $\overline{V}(t)$, we start by collecting the transversion terms from the matrix $P(2t)$ and multiply each of these terms by νq_a, with $a \in \xi$, as follows

$$\begin{aligned}
&\nu q_T(e^{2t\lambda_1} q_A - e^{2t\lambda_2} q_A), && \nu q_T(e^{2t\lambda_1} q_G - e^{2t\lambda_2} q_G),\\
&\nu q_C(e^{2t\lambda_1} q_A - e^{2t\lambda_2} q_A), && \nu q_C(e^{2t\lambda_1} q_G - e^{2t\lambda_2} q_G),\\
&\nu q_A(e^{2t\lambda_1} q_T - e^{2t\lambda_2} q_T), && \nu q_A(e^{2t\lambda_1} q_C - e^{2t\lambda_2} q_C),\\
&\nu q_G(e^{2t\lambda_1} q_T - e^{2t\lambda_2} q_T), && \nu q_G(e^{2t\lambda_1} q_C - e^{2t\lambda_2} q_C).
\end{aligned}$$

Adding these terms, recalling that $\lambda_1 = 0$ and $\lambda_2 = -\beta$, and that we have defined $q_Y = q_T + q_C$ and $q_R = q_A + q_G$, we obtain

$$\overline{V}(t) = 2\nu N q_Y q_R[1 - e^{-2\beta t}].$$

Similarly, for the average number of transitions $\overline{S}(t)$, we collect the transition terms from the matrix $P(2t)$ and multiply each of these terms by νq_a, $a \in \xi$, as before and obtain

$$\begin{aligned}
&\nu q_T(e^{t\lambda_1} q_C + e^{t\lambda_2} q_Y^{-1} q_R q_C - e^{t\lambda_4} q_C q_Y^{-1}),\\
&\nu q_C(e^{t\lambda_1} q_T + e^{t\lambda_2} q_Y^{-1} q_R q_T - e^{t\lambda_4} q_T q_Y^{-1}),\\
&\nu q_A(e^{t\lambda_1} q_G + e^{t\lambda_2} q_R^{-1} q_Y q_G - e^{t\lambda_3} q_G q_R^{-1}),\\
&\nu q_G(e^{t\lambda_1} q_A + e^{t\lambda_2} q_R^{-1} q_Y q_A - e^{t\lambda_3} q_A q_R^{-1}).
\end{aligned}$$

Adding these terms, and recalling that $\lambda_3 = -q_Y\beta - q_R\alpha$ and $\lambda_4 = -q_Y\alpha - q_R\beta$, we obtain

$$\begin{aligned}
\overline{S}(t) = 2\nu N\Big[&q_T q_C + q_A q_G\\
&+\Big(q_T q_C \frac{q_R}{q_Y} + q_A q_G \frac{q_Y}{q_R}\Big) e^{-2t\beta}\\
&-\frac{q_A q_G}{q_R} e^{-2t(q_Y\beta + q_R\alpha)} - \frac{q_T q_C}{q_Y} e^{-2t(q_Y\alpha + q_R\beta)}\Big].
\end{aligned}$$

To compute variances and covariances, we use standard equations as follows

$$\sigma_V^2 = \overline{V}(t)\Big(1 - \frac{\overline{V}(t)}{N}\Big),$$

$$\sigma_S^2 = \overline{S}(t)\Big(1 - \frac{\overline{S}(t)}{N}\Big),$$

$$\sigma_{VS} = \sigma_{SV} = -\frac{\overline{V}(t)\overline{S}(t)}{N}.$$

We can now define the following terms before constructing the likelihood function

$$\boldsymbol{D} = \begin{pmatrix} V_n \\ S_m \end{pmatrix}, \quad \overline{\boldsymbol{D}} = \begin{pmatrix} \overline{V}(t) \\ \overline{S}(t) \end{pmatrix}, \text{ and } \quad \boldsymbol{\Omega} = \begin{pmatrix} \sigma_V^2 & \sigma_{VS} \\ \sigma_{SV} & \sigma_S^2 \end{pmatrix}.$$

Assuming $\boldsymbol{D} \sim \mathcal{N}(\overline{\boldsymbol{D}}, \boldsymbol{\Omega})$, the likelihood function can be written as

$$\mathcal{L}(\boldsymbol{\theta} \mid \mathcal{A}) = (2\pi \det|\boldsymbol{\Omega}|)^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{D} - \overline{\boldsymbol{D}})' \boldsymbol{\Omega}^{-1} (\boldsymbol{D} - \overline{\boldsymbol{D}}) \Big\}. \tag{2.16}$$

The parameter vector $\boldsymbol{\theta}$ has the elements (t, ν, α, β), and 2.16 can be maximised by minimising $(\boldsymbol{D} - \overline{\boldsymbol{D}})' \boldsymbol{\Omega}^{-1} (\boldsymbol{D} - \overline{\boldsymbol{D}})$.

Here it was possible for the authors to derive an optimisation criterion analytically. This approach, however, is not practical when using HMMs to model heterogeneity of evolutionary rates with several parameters. For this reason, I had to consider other approaches for optimising the likelihood function. One such approach was the Baum-Welch algorithm, which I described in Chapter 1. This method, together with numerical approximation, allowed me to deal with a complex optimisation function which is not amenable to an analytical formulation.

2.4 Insertions and Deletions

The three substitution models that I have described, namely, the PMB, GY94 and HKY85, are all built around a sound mathematical structure. They are also amenable to exponentiation and to maximum likelihood so that they allow the evolutionary time parameter to be computed as an asymptotically consistent estimator along a continuum while taking into account potential multiple hits on a single site. An important feature that they also share is versatility when applied to data sets that have a wide range of pairwise evolutionary times.

They have, however, a serious limitation. Each of these models is oblivious to insertions and deletions (indels). Intuitively, the greater the evolutionary time between two sequences, the more indels we can expect to encounter, and the substitution model becomes less and less effective in comparative analysis. A substitution model, therefore, would require additional modelling to capture the effects of indels on the "true" alignment. We would want to consider a parameter that can capture

the average rate of occurrence of indels, and a second parameter that can model the average length of these indels.

PASCARELLA and ARGOS (1992) studied indels in homologous pairs constructed from a collection of 32 protein structural families stored in the Brookhaven Protein Data Bank. The results of these workers provide a strong motivation for the design of my one region model which I shall describe in this chapter. Here I give an overview of the relevant methods and results of PASCARELLA and ARGOS (1992).

The authors considered all possible pairings of the aligned tertiary structures stored in the database at the time. The corresponding pairs of primary sequences were grouped according to the percentage identity c exhibited by each pair, with each c falling within a 5% interval. This grouping led to a histogram showing that most of the data had c values ranging from 5% to 35%, meaning that paired structures had accumulated a great amount of evolution. Within the context of this characteristic in the data, the following features emerged:

1. Once the length of an indel exceeded just one site in an alignment, the number of indels longer than one dropped sharply. This was evidence that the general assumption of species exercising great economy when mutating through deletions and insertions of nucleotides would generally hold true. We would not, therefore, expect alignment models to produce excessively long gaps.

2. The behaviour of the average indel length ($\bar{\ell}$) for c values between 5 and 60, remained at about 2 for most of evolutionary time, and not until c dropped below 25% that $\bar{\ell}$ started to rise sharply. Even when $\bar{\ell}$ increased, it did not reach more than 5, at which length it appeared to remain fixed. PASCARELLA and ARGOS (1992) stated that

   > The tendency then, once indel sites are established, is to reach an equilibrium length such that residues are inserted or deleted in a balanced manner with time. Furthermore, there is a limit, in general, to the size of an indel, around five.

   This feature suggests that to model gaps, a discrete probability distribution which could rapidly reduce the probability of additional unit gaps occurring –

once a gap length had stabilised to about 2 – would be suitable. For example, with the geometric distribution having parameter a set to 0.2, one could use the model

$$P(\text{Indel Length} = \bar{\ell}) = (1-a)a^{\bar{\ell}-1} \tag{2.17}$$

which would give a high probability of around 0.8 if $\bar{\ell}$ was just 1. This probability would drop to just 0.15 if $\bar{\ell}$ was 2, and to virtually zero if $\bar{\ell}$ reached 5.

3. An extrapolation after c had reached 4% showed that non-gapped sites of alignments converged to a length of about 8 sites. The inference here was that as the residue identity tends towards zero, the smallest average length of aligned residues is about eight, and this is approximately equal to the average length of an α-helix and a β-strand. Here, therefore, was a strong indication that insertions and deletions do not target secondary structural elements. PASCARELLA and ARGOS (1992) stated that

   > ...indels mostly intrude in turn and coil structures, and rarely encroach upon helices and strands...

   It is, therefore, very important to incorporate secondary structure in pairwise alignment modelling.

4. The indel rate was shown to saturate at around 15% residue identity, or $c = 15$, where the rate was just over 5 insertions per 100 non-gapped sites. At lower c values, it was unclear how the rate of insertions actually occurred. This appears to be the reason why it becomes increasingly difficult, if not impractical, to align sequence pairs that expressed large divergence. Nevertheless, PASCARELLA and ARGOS (1992) made the observation that as c tended to approach zero, we would need on average seven non-gapped segments – each having 8 aligned sites – in order to have six insertions of 5 residues each. That makes a total of 86 sites, which is very close to 100 and is consistent with the previous features.

For modelling purposes, the indel rate stays at around 1 per 100 aligned residues until $c \approx 65\%$, after which it rises steeply, suggesting an exponential behaviour. Hence, in general, one could consider the model

$$\text{Indel Rate} = k_1 e^{k_2(0.65 - c/100)} \quad \text{indels/aligned residue}, \tag{2.18}$$

where k_1 and k_2 are some suitable constants.

5. From among the 20 proteins, Glycine was the most frequent that flanked insertions, while Isoleucine was more likely to be located away from gaps. The proteins D, G, K, N, P, R, S and T, which are hydrophilic, were more likely to appear on the flanks of indels than other proteins.

   This feature has been useful in deterministic modelling (THOMPSON *et al.*, 1994). For the purpose of probability modelling, I have divided background probabilities into two sets, namely, the hydrophilic set $\mathcal{H}$ and the nonhydrophilic set $\overline{\mathcal{H}}$. By introducing the hydrophilicity parameter h, I can re-estimate from the data the vector of background probabilites $\boldsymbol{q}$ using

   $$\hat{\boldsymbol{q}}' = k\big(h\mathcal{H} \cup (1-h)\,\overline{\mathcal{H}}\big), \tag{2.19}$$

   where k is a suitable scalar so that the new elements in $\hat{\boldsymbol{q}}'$ still sum to one.

   The h parameter allows me to investigate whether there is a clear demarcation between hydrophilic and non-hydrophilic regions. It also allows me to test whether there exists a significant correlation, positionally in primary structure, between regions which are solvent and regions which exhibit faster (or slower) rate of evolution. Should I find that such a correlation exists, it would then be useful to know whether faster (or slower) rates of evolution are more likely to occur in hydrophilic regions of the molecule. This can shed light on evolutionary processes in coding DNA segments of the genome.

2.5 The Pair Hidden Markov Model

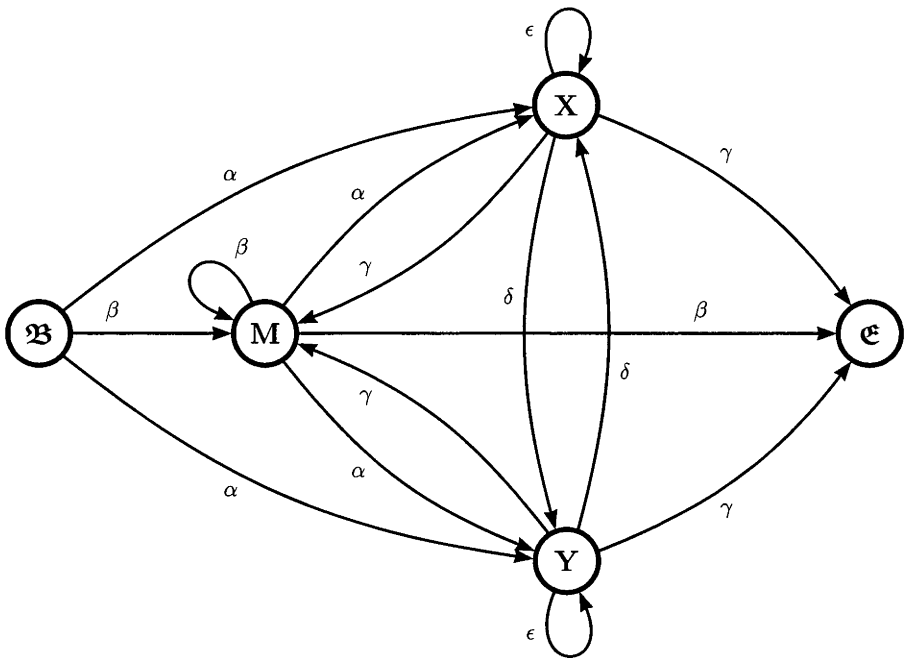


Figure 2.5: The PHMM with begin $\mathfrak{B}$ state, end $\mathfrak{E}$ state, align M state, delete X state and insert Y state. Transition probabilities are computed using the Knudsen-Miyamoto equations for specified parameters, namely, evolutionary time t, indel length a, and indel rate r. Note that probabilities α and β emanating from $\mathfrak{B}$ are the same as those emanating from M, and probabilities β and γ entering $\mathfrak{E}$ are the same as those entering M. States $\mathfrak{B}$ and $\mathfrak{E}$ are therefore redundant. However, $\mathfrak{B}$ is useful when introducing starting values, and $\mathfrak{E}$ makes it easier to deal with the fact that sequences have finite and different lengths.

The pair hidden Markov model (PHMM) was formally introduced by DURBIN *et al.* (1998) to address the issue of indels in probability modelling of pairwise alignments. Their formulation requires the estimation of a parameter vector that has five elements, namely, $(\alpha, \beta, \delta, \epsilon, \gamma)$ as shown in Figure 2.5. None of these five elements on its own can address directly the issues of indel length, indel rate and evolutionary time. To address these issues more directly and economically, KNUDSEN and MIYAMOTO (2003) proposed the theory that leads to a set of equations which I refer to as the Knudsen-Miyamoto (KM) equations. These equations reduce the number of parameters that need to be optimised down to three, including the average indel length a and the average indel rate r. Together, these two parameters describe the indel component of the evolutionary process which had

been studied empirically by PASCARELLA and ARGOS (1992) as I described earlier. I summarise the KM equations as follows:

$$\alpha = \frac{1}{2}p_1\left[1 - \frac{1}{2}p_2\left(\frac{1}{2} - p_4\right)\right], \qquad \beta = 1 - p_1\left(1 - \frac{1}{4}p_2p_3\right),$$

$$\gamma = \frac{\frac{1}{2}ap_2p_3 - \left(\frac{7}{8}p_1 - 1\right)(1-a)}{1 + \frac{1}{2}ap_2\left(\frac{1}{1-a}\right)}, \qquad \delta = \frac{\frac{1}{2}(1-a)ap_2p_4 + \frac{3}{8}p_1(1-a)^2}{1 - a + \frac{1}{2}ap_2},$$

$$\epsilon = \frac{\frac{1}{2}ap_2\left(p_4 + \frac{a}{1-a}\right) + \frac{1}{2}p_1(1-a) + a}{1 + \frac{1}{2}ap_2\left(\frac{1}{1-a}\right)},$$

where the probabilities p_1, p_2, p_3, and p_4 are defined as

$$p_1 = 1 - e^{-2rt}, \quad p_2 = 1 - \frac{p_1}{2rt}, \quad p_3 = \frac{1-a}{1+a}, \quad p_4 = \frac{a}{1+a}.$$

Probabilities p_3 and p_4 use the geometric distribution to model the indel length through the parameter a, while probabilities p_1 and p_2 use the exponential distribution to model the indel rate through the parameter r. This is compatible with features 2 and 4, respectively, in the PASCARELLA and ARGOS (1992) study discussed earlier. The evolutionary time is measured in units of expected substitutions or replacements (KNUDSEN and MIYAMOTO, 2003) through the parameter t.

2.5.1 Transition Probabilities Matrix

The transition probabilities are computed according to a Markov process represented by the matrix

$$T = \begin{array}{c} \\ \mathfrak{B} \\ \mathrm{M} \\ \mathrm{X} \\ \mathrm{Y} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{M} & \mathrm{X} & \mathrm{Y} & \mathfrak{E} \end{array} \\ \begin{bmatrix} \beta & \alpha & \alpha & 0 \\ \beta & \alpha & \alpha & \beta \\ \gamma & \epsilon & \delta & \gamma \\ \gamma & \delta & \epsilon & \gamma \end{bmatrix} \end{array}.$$

From the KM equations we can observe that each transition probability (with the

exception of the $\mathfrak{B}$ to $\mathfrak{E}$ transition, which is set to zero since this transition is not interesting) depends on both the indel process and on evolutionary time between the two sequences. The PHMM, therefore, provides the additional modelling which is lacking in the substitution model. That is, it provides the explicit modelling of the insertion/deletion process through parameters a and r. Through p_2, it also models instances when an indel event is followed by a second indel event at the same site of the alignment. This feature is important because the modelling of double indel events allow us to deal with sequence pairs that have a lower residue identity. The KM equation for the δ transition probability also ensures that modelling for double events does not lead to fragmentation of gaps. KNUDSEN and MIYAMOTO (2003) stated that

> ...for a given evolutionary history, there is not always a unique alignment corresponding to [that history].

For this reason they argue that it makes biological sense that we choose the alignment from all possible alignments that does not exhibit a high dispersion of gaps.

2.5.2 Emission Probabilities Matrix

The emission probabilities of the PHMM are constructed in accordance with GONNET and BENNER (1996), using (1) the emission matrices of the two sequences S_1 and S_2 being aligned, (2) the evolutionary matrix $P(t)$, and (3) the vector $\boldsymbol{q}$ of background probabilities.

The emission matrix of a sequence would normally have elements that are zeros and ones. Due to machine error, however, some of the ones may have to be broken down to fractions wherever there is uncertainty in the identification of a nucleotide during the sequencing procedure. For example, let S_1 and S_2 be the nucleotide sequences CTCGA and ASTCGT with emission matrices W and Z, respectively. Note that the first sequence will have an emission matrix with only zeros and ones. This will not be the case with the second sequence since one of the letters does not belong to the DNA alphabet. There is sequencing uncertainty at the second position which has been designated as S. Hence, the emission matrices of S_1 and S_2 would take the following form

$$W = \begin{array}{c} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{A} \\ \mathrm{G} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{C} & \mathrm{T} & \mathrm{C} & \mathrm{G} & \mathrm{A} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}, \qquad Z = \begin{array}{c} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{A} \\ \mathrm{G} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathrm{A} & \mathrm{S} & \mathrm{T} & \mathrm{C} & \mathrm{G} & \mathrm{T} \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & \frac{1}{2} & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

Note how the weight has been split into two for the alphabet letters C and G that correspond to the sequenced letter S in the second matrix.

Once the emission matrices are constructed, the matrix of emission probabilities can be computed from the following matrix manipulations

$$\mathrm{E}_{\mathrm{M}_{w\times z}} = (PW)'\Big(\big(\boldsymbol{q}\otimes \mathbf{1}_n'\big)\bullet I_{n\times n}\Big)PZ, \tag{2.20}$$

$$\mathrm{E}_{\mathrm{X}_{w\times 1}} = (PW)'\boldsymbol{q}, \tag{2.21}$$

$$\mathrm{E}_{\mathrm{Y}_{1\times z}} = \boldsymbol{q}'PZ, \tag{2.22}$$

where n is the number of letters in the alphabet,

w and z are the lengths of S_1 and S_2, respectively,

the n elements of vector $\boldsymbol{q}$ sum to one,

the n elements of vector $\mathbf{1}$ are all 1's,

$(\otimes)$ and $(\bullet)$ are the Kronecker and Hadamard products, respectively, and

t is omitted for notational convenience.

To simplify implementation, the three matrices $\mathrm{E}_{\mathrm{M}_{w\times z}}$, $\mathrm{E}_{\mathrm{X}_{w\times 1}}$ and $\mathrm{E}_{\mathrm{Y}_{1\times z}}$ are grouped together in one composite emission probabilities matrix as follows

$$E = \left[\begin{array}{c|c} \mathrm{E}_{\mathrm{M}_{w\times z}} & \mathrm{E}_{\mathrm{X}_{w\times 1}} \\ \hline \mathrm{E}_{\mathrm{Y}_{1\times z}} & \mathbf{0} \end{array}\right].$$

For the simple example of S_1 and S_2, E would take the form shown in Figure 2.6.

$$E = \begin{array}{c} \\ \mathrm{C} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{G} \\ \mathrm{A} \\ \\ \mathrm{Y} \end{array} \begin{array}{c} \begin{array}{cccccccc} \mathrm{M} & \mathrm{A} & \mathrm{S} & \mathrm{T} & \mathrm{C} & \mathrm{G} & \mathrm{T} & \mathrm{X} \end{array} \\ \left[\begin{array}{cccccc:c} \mathrm{E_{CA}} & \mathrm{E_{CS}} & \mathrm{E_{CT}} & \mathrm{E_{CC}} & \mathrm{E_{CG}} & \mathrm{E_{CT}} & \mathrm{E_{C-}} \\ \mathrm{E_{TA}} & \mathrm{E_{TS}} & \mathrm{E_{TT}} & \mathrm{E_{TC}} & \mathrm{E_{TG}} & \mathrm{E_{TT}} & \mathrm{E_{T-}} \\ \mathrm{E_{CA}} & \mathrm{E_{CS}} & \mathrm{E_{CT}} & \mathrm{E_{CC}} & \mathrm{E_{CG}} & \mathrm{E_{CT}} & \mathrm{E_{C-}} \\ \mathrm{E_{GA}} & \mathrm{E_{GS}} & \mathrm{E_{GT}} & \mathrm{E_{GC}} & \mathrm{E_{GG}} & \mathrm{E_{GT}} & \mathrm{E_{G-}} \\ \mathrm{E_{AA}} & \mathrm{E_{AS}} & \mathrm{E_{AT}} & \mathrm{E_{AC}} & \mathrm{E_{AG}} & \mathrm{E_{AT}} & \mathrm{E_{A-}} \\ \hdashline \mathrm{E_{-A}} & \mathrm{E_{-S}} & \mathrm{E_{-T}} & \mathrm{E_{-C}} & \mathrm{E_{-G}} & \mathrm{E_{-T}} & 0 \end{array}\right] \end{array}$$

Figure 2.6: In the $(w+1) \times (z+1)$ emission matrix E, the top-left quadrant holds emission probabilities for aligned positions of the pairwise alignment. The emission probabilities of every unique letter-pair combination sum to one. The top-right quadrant holds emission probabilities for positions with deleted characters, while the bottom-left quadrant holds emission probabilities for positions with inserted characters. In each case, probabilities corresponding to unique characters sum to one. The bottom-right quadrant is set to zero since it is not interesting. M, X, and Y are the three states of the PHMM, with each state emitting its respective matrix of emission probabilities that sum to one.

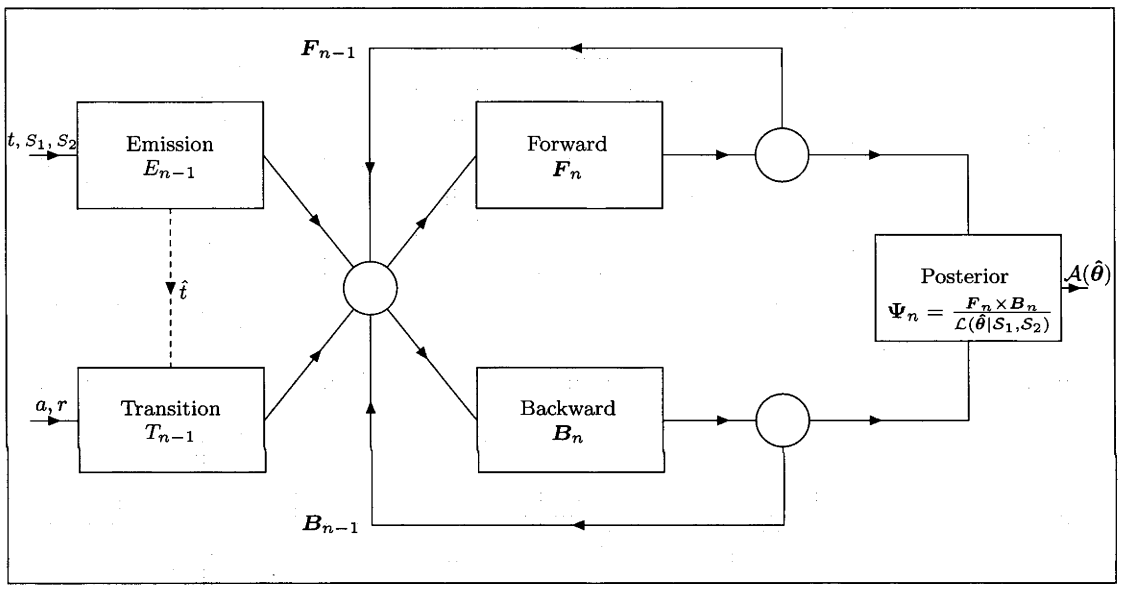


Figure 2.7: The one-region model takes three parameters, namely, evolutionary time t, indel length a, and indel rate r, along with the data consisting of the two sequences S_1, S_2 to be aligned. $\hat{t}$ is the estimator for the expected number of substitutions or replacements between the two sequences S_1, S_2. It is also part of the PHMM where $\hat{a}$ and $\hat{r}$ are estimated by the Markov chain according to the transition probabilities matrix T. This matrix, together with emission matrix E, is used with standard dynamic programs to compute the set of forward probabilities matrices $\boldsymbol{F}$ and the set of backward probabilities matrices $\boldsymbol{B}$. From these, the corresponding set of posterior probabilities matrices $\boldsymbol{\Psi}$ are also computed. The latter matrices are used with a standard trace-back procedure to produce the alignment $\mathcal{A}(\hat{\boldsymbol{\theta}})$. The index n is incremented with each position of the alignment of sequences S_1, S_2. $\hat{\boldsymbol{\theta}}$ is the vector of the ML estimators $\hat{t}, \hat{a}$ and $\hat{r}$. There is also an additional parameter, namely, the sequence length s parameter, which optimises for finite and unequal sequences. This parameter plays a relatively minor role and is not shown.

2.5.3 One-Region Modelling

Matrices T and E of transition and emission probabilities, respectively, are used with standard dynamic programs to compute the set of matrices $\boldsymbol{F}$ and the

set of matrices $\boldsymbol{B}$ of forward and backward probabilities, respectively. These computations are carried out iteratively as shown in Figure 2.7. The summation of the forward probabilities across all possible alignments at each iteration is used for maximising the following likelihood function

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{S}_1, \mathcal{S}_2) = \sum_{n=1}^{N} \prod_{\substack{i \in \mathcal{S}_1 \\ j \in \mathcal{S}_2}} \mathcal{A}_{ijn}(\boldsymbol{\theta}), \tag{2.23}$$

where $\mathcal{A}_{ijn}(\boldsymbol{\theta})$ is the contribution of letters i and j of sequences $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, of alignment n,
N is the number of all possible alignments, and
$\boldsymbol{\theta}$ is the parameter vector to be optimised.

Following maximisation, the ML estimator $\hat{\boldsymbol{\theta}}$ is used to compute the set of tables $\boldsymbol{\Psi}$ of posterior probabilities. A standard trace-back procedure is then applied to construct the alignment $\mathcal{A}$ from these tables, and to compute a posterior probability for each position of the alignment. Each of these posterior probabilities provides a measure of how strong the alignment is between the corresponding letter-pair at that position.

2.5.3.1 The Trace-Back Procedure

The trace-back procedure produces $\mathcal{A}$ from the three posterior probabilities tables Ψ_M, Ψ_X, and Ψ_Y. It produces n_M aligned character-pairs, n_X inserts in S_1, and n_Y deletes in S_1 from the three tables, respectively. The expectations $E(n_M)$, $E(n_X)$, and $E(n_Y)$ of these three numbers is computed by summing all probabilities in each respective posterior probabilities table. By the property of invariance of ML, $\mathcal{A}(\hat{\boldsymbol{\theta}})$ would – asymptotically and under regulatory conditions – also be the ML estimator (e.g. GREENE, 1997, p. 133) of the true alignment if $E(n_M) = n_M$. $\left|E(n_M) - n_M\right|$, however, is a random variable whose distribution depends on the accuracy of the trace-back procedure. This procedure does not guarantee that this random variable is always not statistically different from zero. For this reason, $\mathcal{A}(\hat{\boldsymbol{\theta}})$ is not strictly the ML estimator of the true alignment.

CHAPTER 3
The Two-Region Model

3.1 Modelling Heterogeneity in Molecular Evolution

Molecular evolutionists have long been aware that different segments of biological sequences had been evolving at different rates (FELSENSTEIN and CHURCHILL, 1996). Consider, for example, the argument that all morphological characters are ultimately controlled by DNA (NEI, 2005). These characters had been exposed to environmental changes over evolutionary time. It would follow, therefore, that substitution rates of DNA segments that control these characters would have evolved at different rates in order to allow corresponding parts of the phenotype to adapt to these environmental changes.

MARGOLIASH and SMITH (1965) and ZUCKERKANDL and PAULING (1965) had also argued that amino acid replacements are slower than neutral in regions of the protein that are functionally more important. That is, change in these regions had been suppressed to ensure that the species would not be adversely affected. On the other hand, in regions that are not critical to function, rates of replacement are faster than neutral. In this case, replacements are not detrimental to the species. They can also be slightly positively selected (NEI, 2005) because they provide better chances of survival to those individuals that acquire these variants.

It is therefore of interest to consider a pairwise aligner that takes into account potential heterogeneity of substitution rates along the two DNA homologues. It is not known, however, how this heterogeneity can best be stratified for the purpose of modelling. A reasonable starting point would be to divide heterogeneity into two broad regions, namely, the slow and the fast rates of substitutions. This would be compatible with secondary structure composition whereby α-helices and β-sheets would constitute the conserved region at the core of the molecule, while loops, coils and turns would form the non-conserved region that is present on the hydrophilic surface. Given this setting, one would then expect substitutions to be scarce in the former region, while more common in the latter. From the empirical study of

PASCARELLA and ARGOS (1992), for example, we know that indel processes are more likely to be active in the hydrophilic region which is the point of attack for this component of evolution.

THORNE *et al.* (1992) pioneered the two-region pairwise aligner which assumes regional heterogeneity of substitution rates. This model treats regions along the evolving DNA as consisting of a variety of fragments, where each fragment evolves at its own rate and with a stochastic length drawn from some common probability distribution with one parameter. For the purpose of tractability, the authors categorise these fragments into two broad classes, namely, those that express a fast rate p and those that express a slow rate $1-p$. They also introduce a parameter k to relate these two regional substitution rates. From simulation results that they present, k appears to deviate greatly from the true value, and exhibits large variances. The parameter k may in fact be unnecessary since substitution rates in a region are likely to be dependent on secondary structure, and ultimately on phenotypic requirements related to that region, but not on the substitution rates of the other region. The authors also show concern that their model does not account for possible increases in the indel rate with increases in p. A more serious concern is, however, that the substitution rate p in the fast rate region is tied to the substitution rate $1-p$ in the slow rate region. In reality, one would expect that the two rates are independent from each other; that is, one does not increase (or decrease) at the expense (or the benefit) of the other.

Independence between the two rates is necessary for the fact that slow and fast rates serve two unrelated purposes. The slow rate ensures that selection is more rigorous and precise, and hence needs more time to mature. On the other hand, the fast rate ensures that change does occur, even if the outcome of this change may not always be exactly what was needed. In fact, the outcome from the latter could even turn out to be slightly harmful, but still much better than if no change had occurred. Assuming that this premise is true, it should then follow that the two rates exist in parallel but separately. One would also not expect that the two putatively independent substitution rates, slow and fast, would be random variables necessarily drawn from the same probability distribution along the same stretch of

DNA. One could consider, for example, that slow rates are normally distributed across species, while fast rates are drawn from some other non-normal distribution.

Independence between the fast and slow rate regions also lead me to reasonably hypothesise that there is a correlation between regional evolutionary rates and clearly defined parts of secondary structure. This correlation would be similar to the correlation suggested by GOLDMAN *et al.* (1996), whereby species in a phylogeny are positioned in accordance with a hierarchy of evolutionary rates. Here, evolutionary rates in different branches of the tree are non-independent, but they are also "averaged". Secondary structure in the data from which the phylogeny is derived cannot be observed, but it can be estimated by employing a three state HMM, namely, a state for α-helices (α), a state for β-sheets (β), and another state for everything else (L) (GOLDMAN *et al.*, 1996). This approach leads to the modelling of evolution which takes place through amino acid replacements that are described by three phylogenetic trees rather than just one. Rates within each of these three trees are non-independent, but there is nothing to suggest that they are not independent between any two of the three distinct phylogenetic trees.

Focusing on these processes that are likely to have generated the data, I can now visualise a novel device to model a pairwise alignment. This device consists of two-tiered HMMs. The first layer seeks to tease out the true substitution and indel processes as in DURBIN *et al.* (1998) and in KNUDSEN and MIYAMOTO (2003), while the second layer attempts to exploit the correlation of the aligned sites with different parts of secondary structure, as I shall demonstrate later in this chapter and which we showed in SAMMUT *et al.* (2006).

FELSENSTEIN and CHURCHILL (1996) employ an HMM to allocate, to each site, a rate selected (that is, "emitted") from a category of pre-defined and finite number of rates. The category, here, represents the region along the molecule which is experiencing a slow or a fast rate of substitution. Thus, a region would consist of a "cluster" of rates contiguously emitted from the same category. These clusters are in turn correlated through a parameter λ, which is equivalent to the parameter p in the THORNE *et al.* (1992) model. λ is the probability that a rate from a category is followed by a rate from the same category. Note here that unlike in the

THORNE *et al.* (1992) model, regions, and rates from within the same region (that is, category), are assumed to be correlated, but no correlation is assumed between rates drawn from different categories. This seems to be the preferred specification, although rates in the FELSENSTEIN and CHURCHILL (1996) model are not estimated from the data, making this model a naive one-layered HMM, and hence is limited in what it can do.

Incidentally, at time of writing, I became aware of LÖYTYNOJA and GOLDMAN (2008) who use the same two-tiered model with exactly the same HMM-PHMM topology as in SAMMUT *et al.* (2006). [4] These authors, however, parametrize their model using affine gap penalties for each of the two PHMMs (lower level), and probabilities which are predefined and fixed to switch between the two PHMMs (upper level). At both levels, parameters are estimated from training data, namely, biological sequences for parameters within each PHMM, and biological structure classes for parameters that switch between the two PHMMs. LÖYTYNOJA and GOLDMAN (2008) aim to model multiple structure classes in multiple alignment settings.

3.2 The Two-Tiered HMM-PHMM Topology

The PHMM has been formally presented by DURBIN *et al.* (1998). My approach in applying this device for the purpose of aligning two biological sequences is based on the theory proposed by KNUDSEN and MIYAMOTO (2003). The motivation stems from the knowledge that the substitution rate is not homogeneous along the DNA (CHURCHILL, 1989). There also appears to be a correlation between substitution rates and secondary structure (GOLDMAN *et al.*, 1996), and between indel processes and hydrophilic regions of the molecule (PASCARELLA and ARGOS, 1992). My approach uses more than one PHMM to model different substitution and indel rates in different regions. It is not known what the optimal number of PHMMs should be, or whether such an optimal number exists. However, as I discussed earlier, it is reasonable to assume that heterogeneity can be categorised into

[4] LÖYTYNOJA and GOLDMAN (2008) have not cited SAMMUT *et al.* (2006). I presented the topology and the results at the poster session of the 11th International Congress of Human Genetics held at the Brisbane Convention & Exhibition Centre, Brisbane, Australia between August 6 - 10, 2006. The abstract is published on-line courtesy ICMS who also hold a copy of the accompanying PDF file.

two broad regions, namely, the fast and slow rates. Hence, my initial model is as in SAMMUT *et al.* (2006). That is, it consists of two PHMMs conjoined by a silent state denoted by $\mathfrak{S}$ as shown in Figure 3.1.

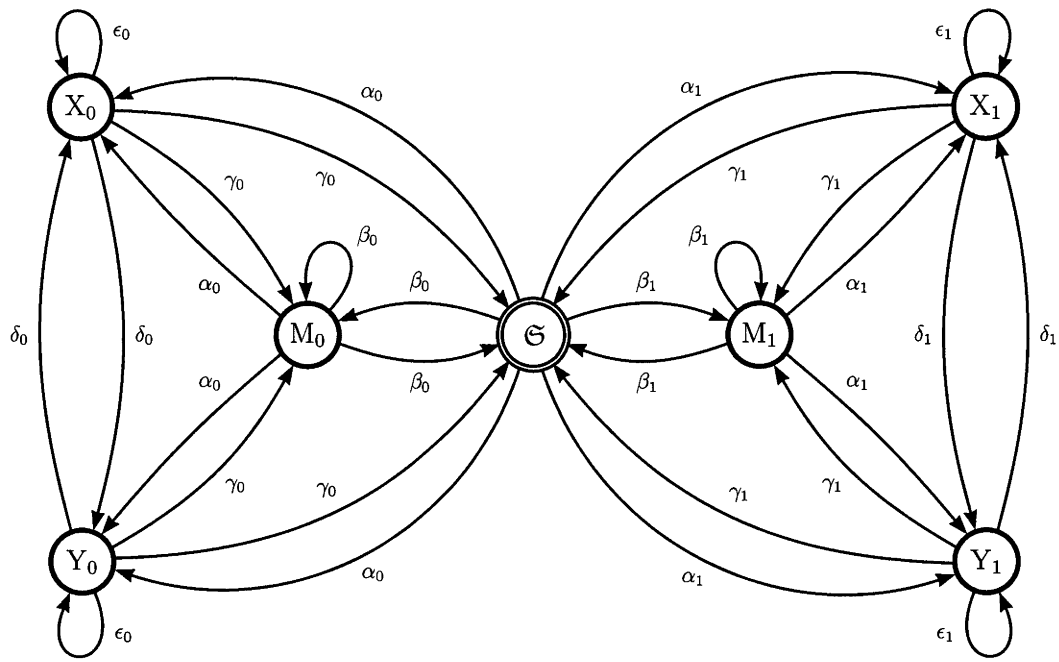


Figure 3.1: The double PHMM topology (SAMMUT *et al.*, 2006) has one silent $\mathfrak{S}$ state. The silent state conceptually replaces the begin and the end states of the one-region model. It allows one to join two PHMMs in such a way whereby each PHMM is allowed to operate *independently* from the other. Thus, when a switch occurs from $PHMM_0$ to $PHMM_1$, $\mathfrak{S}$ behaves as the end state of $PHMM_0$ and as the begin state of $PHMM_1$. Similarly, when a switch occurs from $PHMM_1$ to $PHMM_0$, $\mathfrak{S}$ behaves as the end state of $PHMM_1$ and as the begin state of $PHMM_0$. An important property of the silent state is that it allows the KM equations to be applied independently to each of the two PHMMs. This greatly simplifies parameter ML estimation during optimisation. More importantly, however, it models the two separate parts of secondary structure, namely, the conserved and the non-conserved regions.

3.2.1 The Two-Region Transition Matrix

The aim of the two-region model shown in Figure 3.1 is to capture each of the two broad categories of evolutionary rates by each of the two PHMMs, one with its own set of parameters optimised for slow rates and the other also with its own set of parameters optimised for fast rates. These two sets of parameters lead to two sets of transition probabilities as shown in the composite transition matrix in Figure 3.2. In this matrix, one set of probabilities is indexed zero to signify that they belong to

$PHMM_0$ that models one region, while the other set is indexed one to signify that they belong to $PHMM_1$ that models the other region.

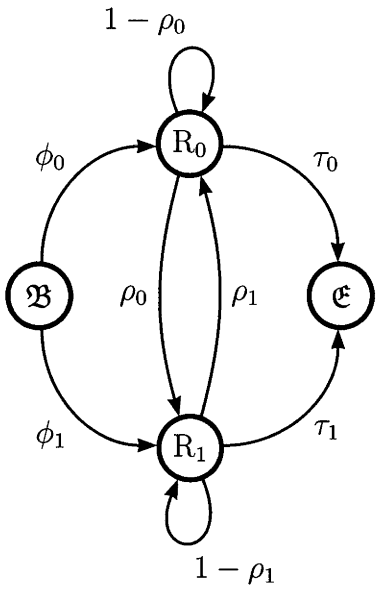
$$\mathbf{T} = \begin{array}{c} \\ \mathfrak{S} \\ \mathrm{M_0} \\ \mathrm{X_0} \\ \mathrm{Y_0} \\ \mathrm{M_1} \\ \mathrm{X_1} \\ \mathrm{Y_1} \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathrm{M_0} & \mathrm{X_0} & \mathrm{Y_0} & \mathrm{M_1} & \mathrm{X_1} & \mathrm{Y_1} & \mathfrak{S} \end{array} \\ \begin{bmatrix} \beta_0 & \alpha_0 & \alpha_0 & \beta_1 & \alpha_1 & \alpha_1 & 0 \\ \beta_0 & \alpha_0 & \alpha_0 & 0 & 0 & 0 & \beta_0 \\ \gamma_0 & \epsilon_0 & \delta_0 & 0 & 0 & 0 & \gamma_0 \\ \gamma_0 & \delta_0 & \epsilon_0 & 0 & 0 & 0 & \gamma_0 \\ 0 & 0 & 0 & \beta_1 & \alpha_1 & \alpha_1 & \beta_1 \\ 0 & 0 & 0 & \gamma_1 & \epsilon_1 & \delta_1 & \gamma_1 \\ 0 & 0 & 0 & \gamma_1 & \delta_1 & \epsilon_1 & \gamma_1 \end{bmatrix} \end{array}$$

Figure 3.2: The conceptual transition matrix **T** of the two-region model is a composite of two 3×3 transition matrices, one from each of the two PHMMs in this model. A transition within one of the PHMMs is similar to that of the one-region model. However, a transition from one PHMM to the other PHMM has to be channelled through the silent state $\mathfrak{S}$. Note that during an inter-region transition, the silent state acts as the end state of the source PHMM and as the begin state of the sink PHMM, simultaneously.

The topology in Figure 3.1 constitutes the first layer of the two-tiered HMM-PHMM model. A second layer is needed to capture the alternating regions of slow and fast evolutionary rates along the DNA. This alternate behaviour of rate heterogeneity can be assumed to be a two-state Markov process which I model by a two-state HMM (SAMMUT *et al.*, 2006) as shown in Figure 3.3 and which has a 2×2 transition matrix. Transition probabilities ρ_0 and ρ_1 can be viewed as region switching probabilities. Each determine when the flow in the current PHMM should traverse to the other PHMM via the silent state. CHURCHILL (1989) showed that under stable DNA heterogeneity, these probabilities would normally be small. That is, I can expect heterogeneity not to be fragmented.

The begin state $\mathfrak{B}$ plays a role only during the first step of the alignment process. The starting probability from $\mathfrak{B}$ to the PHMMs is multiplied by the stationary probabilities ϕ_j, $j \in \{0, 1\}$, to average out the initial uncertainty of the forward dynamic program. Similarly, the end state $\mathfrak{E}$ plays a role only during the last step. The last probability from each of the forward tables to $\mathfrak{E}$ is multiplied by the sequence length parameter τ_i, $i \in \{0, 1\}$, in order to take into account the fact that the sequences being aligned do not have infinite length.

Transiting from one state of $PHMM_\eta$, $\eta \in \{0, 1\}$, to another state of the same PHMM has the same probability as that which is computed from the KM equations



$$
\begin{array}{c} \\ \mathfrak{B} \\ R_0 \\ R_1 \end{array}
\begin{array}{c}
\begin{array}{ccc} R_0 & R_1 & \mathfrak{E} \end{array} \\
\begin{bmatrix} \phi_0 & \phi_1 & 0 \\ 1-\rho_0 & \rho_0 & \tau_0 \\ \rho_1 & 1-\rho_1 & \tau_1 \end{bmatrix}
\end{array}
\begin{array}{c}
\begin{array}{cc} PHMM_0 & PHMM_1 \end{array} \\
\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{array}
$$

Figure 3.3: The two-state HMM models the two regions of the molecular secondary structure, namely, the conserved and non-conserved regions. States R_0 and R_1 emit $PHMM_0$ and $PHMM_1$, respectively with probability 1.0. $\mathfrak{B}$ and $\mathfrak{E}$ are silent states which are replaced by a single silent state $\mathfrak{S}$ in the conceptual transition matrix $\mathbf{T}$ of the two-region model.

except that this probability is now multiplied by $1-\rho_\eta$. Transiting from one state of $PHMM_\eta$, $\eta \in \{0,1\}$ to another state of $PHMM_{1-\eta}$ would require two probabilities computed from the KM equations, and both are multiplied by ρ_η. For example, a transition from state M_0 to state Y_1 would be the product of β_0, α_1, and ρ_0.

$$
\mathbf{T} = \begin{array}{c}
\begin{array}{ccccccc} M_0 & X_0 & Y_0 & M_1 & X_1 & Y_1 & \mathfrak{E} \end{array} \\
\begin{bmatrix}
\beta_0\phi_0 & \alpha_0\phi_0 & \alpha_0\phi_0 & \beta_1\phi_1 & \alpha_1\phi_1 & \alpha_1\phi_1 & 0 \\
\beta_0(1-\rho_0) & \alpha_0(1-\rho_0) & \alpha_0(1-\rho_0) & \beta_0\beta_1\rho_0 & \beta_0\alpha_1\rho_0 & \beta_0\alpha_1\rho_0 & \beta_0\tau_0 \\
\gamma_0(1-\rho_0) & \epsilon_0(1-\rho_0) & \delta_0(1-\rho_0) & \gamma_0\beta_1\rho_0 & \gamma_0\alpha_1\rho_0 & \gamma_0\alpha_1\rho_0 & \gamma_0\tau_0 \\
\gamma_0(1-\rho_0) & \delta_0(1-\rho_0) & \epsilon_0(1-\rho_0) & \gamma_0\beta_1\rho_0 & \gamma_0\alpha_1\rho_0 & \gamma_0\alpha_1\rho_0 & \gamma_0\tau_0 \\
\beta_1\beta_0\rho_1 & \beta_1\alpha_0\rho_1 & \beta_1\alpha_0\rho_1 & \beta_1(1-\rho_1) & \alpha_1(1-\rho_1) & \alpha_1(1-\rho_1) & \beta_1\tau_1 \\
\gamma_1\beta_0\rho_1 & \gamma_1\alpha_0\rho_1 & \gamma_1\alpha_0\rho_1 & \gamma_1(1-\rho_1) & \epsilon_1(1-\rho_1) & \delta_1(1-\rho_1) & \gamma_1\tau_1 \\
\gamma_1\beta_0\rho_1 & \gamma_1\alpha_0\rho_1 & \gamma_1\alpha_0\rho_1 & \gamma_1(1-\rho_1) & \delta_1(1-\rho_1) & \epsilon_1(1-\rho_1) & \gamma_1\tau_1
\end{bmatrix}
\end{array}
\begin{array}{c} \\ \mathfrak{B} \\ M_0 \\ X_0 \\ Y_0 \\ M_1 \\ X_1 \\ Y_1 \end{array}
$$

Figure 3.4: The implementation of the two-region transition matrix $\mathbf{T}$ has transition probabilities consisting of products of probabilities taken both from the conceptual transition matrix and from the two-state region HMM. Each row is normalised to make $\mathbf{T}$ row stochastic. Note also that the silent state $\mathfrak{S}$ is now replaced by the begin state $\mathfrak{B}$ in the first row and by the end state $\mathfrak{E}$ in the last column.

Figure 3.4 shows all the probability transformations that produce the two-region transition matrix that can be implemented in dynamic programming after

each row had been normalised and made to sum to one. Note also that the begin state $\mathfrak{B}$ and the end state $\mathfrak{E}$ are also restored as a result of this transformation.

3.2.2 Emission Matrices for the Two-Region Model

$$\mathbf{E} = \begin{array}{c} \\ \mathrm{C} \\ \mathrm{T} \\ \mathrm{C} \\ \mathrm{G} \\ \mathrm{A} \\ \mathrm{Y}_\eta \end{array} \begin{array}{c} \begin{array}{cccccc} \mathrm{M}_\eta & \mathrm{A} & \mathrm{S} & \mathrm{T} & \mathrm{C} & \mathrm{G} & \mathrm{T} \end{array} \\ \left[\begin{array}{cccccc:c} \mathrm{E}_{\eta,\mathrm{CA}} & \mathrm{E}_{\eta,\mathrm{CS}} & \mathrm{E}_{\eta,\mathrm{CT}} & \mathrm{E}_{\eta,\mathrm{CC}} & \mathrm{E}_{\eta,\mathrm{CG}} & \mathrm{E}_{\eta,\mathrm{CT}} & \mathrm{E}_{\eta,\mathrm{C}-} \\ \mathrm{E}_{\eta,\mathrm{TA}} & \mathrm{E}_{\eta,\mathrm{TS}} & \mathrm{E}_{\eta,\mathrm{TT}} & \mathrm{E}_{\eta,\mathrm{TC}} & \mathrm{E}_{\eta,\mathrm{TG}} & \mathrm{E}_{\eta,\mathrm{TT}} & \mathrm{E}_{\eta,\mathrm{T}-} \\ \mathrm{E}_{\eta,\mathrm{CA}} & \mathrm{E}_{\eta,\mathrm{CS}} & \mathrm{E}_{\eta,\mathrm{CT}} & \mathrm{E}_{\eta,\mathrm{CC}} & \mathrm{E}_{\eta,\mathrm{CG}} & \mathrm{E}_{\eta,\mathrm{CT}} & \mathrm{E}_{\eta,\mathrm{C}-} \\ \mathrm{E}_{\eta,\mathrm{GA}} & \mathrm{E}_{\eta,\mathrm{GS}} & \mathrm{E}_{\eta,\mathrm{GT}} & \mathrm{E}_{\eta,\mathrm{GC}} & \mathrm{E}_{\eta,\mathrm{GG}} & \mathrm{E}_{\eta,\mathrm{GT}} & \mathrm{E}_{\eta,\mathrm{G}-} \\ \mathrm{E}_{\eta,\mathrm{AA}} & \mathrm{E}_{\eta,\mathrm{AS}} & \mathrm{E}_{\eta,\mathrm{AT}} & \mathrm{E}_{\eta,\mathrm{AC}} & \mathrm{E}_{\eta,\mathrm{AG}} & \mathrm{E}_{\eta,\mathrm{AT}} & \mathrm{E}_{\eta,\mathrm{A}-} \\ \hdashline \mathrm{E}_{\eta,-\mathrm{A}} & \mathrm{E}_{\eta,-\mathrm{S}} & \mathrm{E}_{\eta,-\mathrm{T}} & \mathrm{E}_{\eta,-\mathrm{C}} & \mathrm{E}_{\eta,-\mathrm{G}} & \mathrm{E}_{\eta,-\mathrm{T}} & 0 \end{array}\right] \end{array} \begin{array}{c} \\ \mathrm{X}_\eta \end{array}$$

Figure 3.5: The emission matrix **E** of Figure 2.6 is extended to cater for two PHMMs by introducing the index $\eta \in \{0, 1\}$. **E** is now a vector of emission matrices with number of elements equal to number of regions.

The emission matrix of each PHMM in the two-region model is the same as that specified for the one-region model described in Chapter 2. Each matrix is indexed according to the regional PHMM it is assigned to. Emission matrix **E** shown in Figure 3.5 is therefore a vector of matrices whose elements are ordered, starting from region zero, and the number of these elements is equal to the number of regions being modelled. Each element is indexed by η and inherits the same set of parameters assigned to the region η it belongs to, and receives corresponding estimators determined by ML during optimisation.

3.2.3 Two-Region Modelling

Figure 3.6 shows the two-region model in schematic form. The PHMM in each region is represented by the corresponding transition and emission matrices. As in the one-region model, these two matrices share the same substitution (or replacement) rate parameter t_η, $\eta \in \{0, 1\}$, which is assigned to its region independently of the other parameter $t_{1-\eta}$ assigned to the other region. Likewise, parameters a_η and r_η are assigned to corresponding transition matrices.

The two-region model differs from the one-region model in three important ways. First, the transition matrix of each PHMM now has an additional parameter, namely, ρ_η. This parameter is also estimated from the data. Its estimator decides

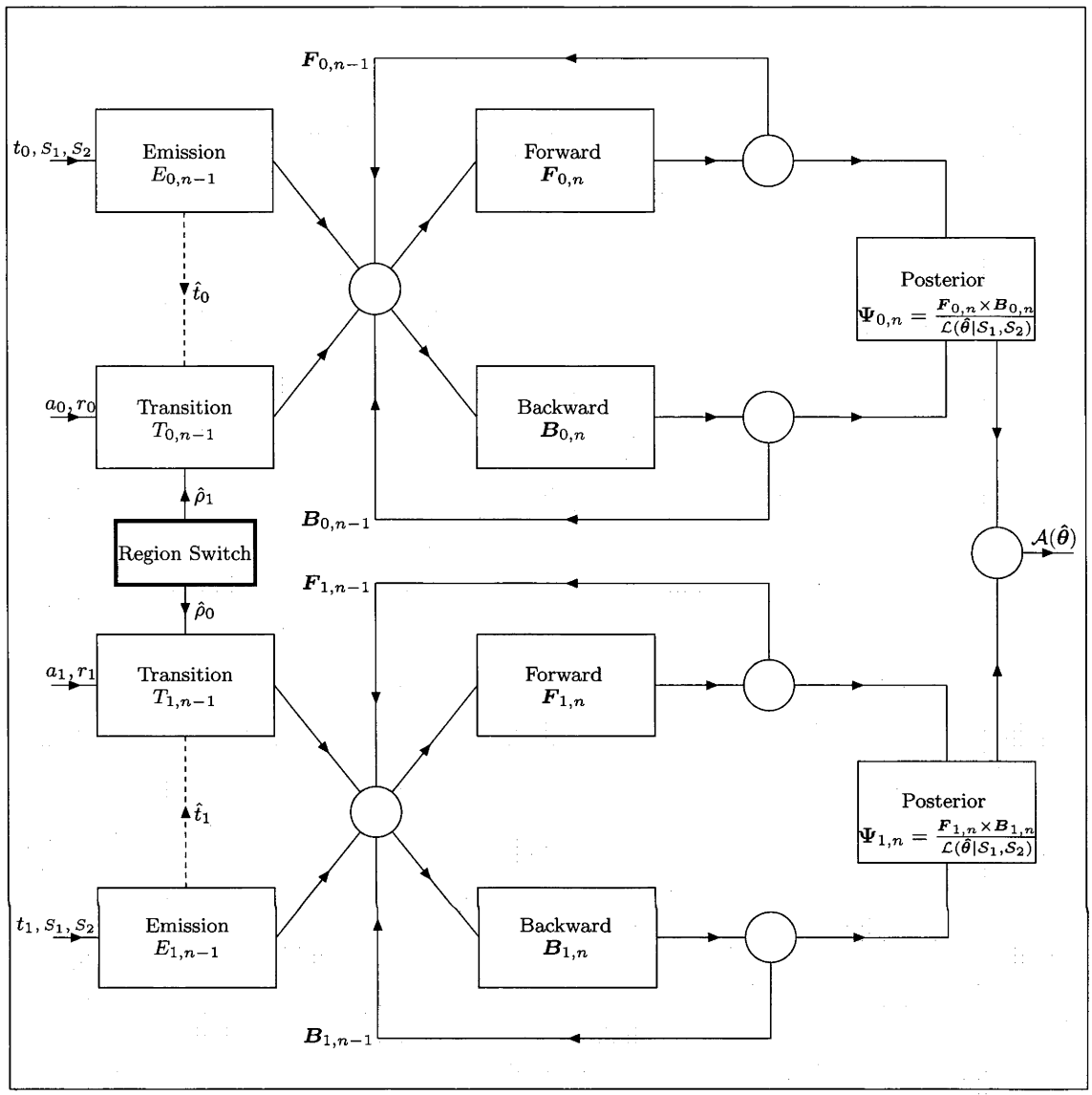


Figure 3.6: In the two-region model, each region is identified by the index number of the input parameters t_η, a_η, and r_η, and of the matrices E_η, T_η, $\boldsymbol{F}_\eta$, $\boldsymbol{B}_\eta$, and $\boldsymbol{\Psi}_\eta$, of each $PHMM_\eta$, $\eta \in \{0, 1\}$. The region switch connects directly to the transition matrix of each PHMM, and switches from $PHMM_0$ to $PHMM_1$ through parameter ρ_0, and from $PHMM_1$ to $PHMM_0$ through parameter ρ_1. The dynamic program in each PHMM steps forward and backward through the index $\{n\colon n = 1, 2, \ldots, N\}$, where N is the number of all possible alignments. The numerical optimiser reads from the three forward matrices in $\boldsymbol{F}_0$ of region one and from the three forward matrices in $\boldsymbol{F}_1$ of region two to compute the maximum likelihood $\mathcal{L}$ (2.23). Likewise, the trace-back procedure reads from the three posterior matrices in $\boldsymbol{\Psi}_0$ of region one and from the three posterior tables in $\boldsymbol{\Psi}_1$ of region two to produce the alignment $\mathcal{A}(\boldsymbol{\theta})$.

which region each alignment site should belong to in order to achieve a better likelihood. Thus, ρ_η would switch the HMM signal from $PHMM_\eta$ to $PHMM_{1-\eta}$ if this would cause the likelihood to rise, otherwise, $\rho_{1-\eta}$ would switch the signal

from $PHMM_{1-\eta}$ to $PHMM_{\eta}$. Second, the likelihood function $\mathcal{L}(\boldsymbol{\theta}|\mathcal{S}_1, \mathcal{S}_2)$ is now dependent on a parameter vector $\boldsymbol{\theta}$ which is a collection of parameters sourced from two PHMMs rather than just one during optimisation. Nevertheless, the optimiser remains oblivious to the fact that more than one region are being modelled. It simply continues to receive values that are additive from the forward dynamic program of each corresponding PHMM. Finally, the trace-back procedure reads from six rather than three posterior probabilities matrices; that is, three tables belonging to region $\eta \in \{0, 1\}$ are collected in $\boldsymbol{\Psi}_{\eta}$, whereby each of this set of three tables had been computed through the forward and backward probability tables of the corresponding region.

3.3 Model Testing

Figure 3.7 shows simulation results consisting of one-sample t-tests and of two-factor ANOVA tables for each biological encoding (BE), namely, protein, codon, and DNA.

KNUDSEN and MIYAMOTO (2003) tested the parameterisation of the PHMM. My aim here was to test the double PHMM topology as a two-region model by varying one parameter at a time across two regions. To carry out the test, I simulated sets of 12 alignments for each BE under the corresponding regimes of parameter values, as shown in the three tables on the left of Figure 3.7. Each cell in these tables, for each combination of parameter values, gives a p-value obtained from a one-sample t-test. Each of these p-values is computed from a data set consisting of 24 point estimators. These were obtained by optimising the likelihood over two parameters across the 12 alignments of the corresponding set.

3.3.1 Simulations

The same 24 point estimators were used to carry out two-factor ANOVA analyses with $K_{ij} = 24$ (i.e. DEVORE, 1990, p. 413). Factor A is made up of the three evolutionary distances shown in the first row of the three tables on the left. Factor B is made up of parameter settings shown in column five of the same tables.

The p-values obtained from the one-sample t-tests show that H_o: $x = x_o$ is

retained for all parameter combinations for the protein and DNA BEs at the 5% level of significance. For the codon BE, however, H_o is rejected once at $t_1 = 0.4$ and four times at $t_1 > 0.4$. This can be expected owing to the fact that the GY94 model is designed for close homologues, where the distance is assumed to be less than 0.4.

The p-values obtained from the ANOVAs provide a similar picture at the 5% level of significance. First, Factor AB is not significant in all three tables. That is, there is no significant interaction between distance t point estimators and all other point estimators. This means that I can interpret directly the effects of Factor A and Factor B on model performance.

Second, for all three BEs, Factor B has no significance on model performance. That is, model performance can be expected to be the same regardless over which parameter, other than the distance t parameter, I am optimising the likelihood.

Finally, for the protein BE, Factor A has no significance on model performance. That is, I can align sequence pairs with evolutionary distances of at least 0.8 without compromising alignment quality. This is not the case, however, with the codon and DNA BEs. Once again, this is expected since both the GY94 and the HKY85 were designed for close homologues. On the other hand, as I discussed in chapter 2, the PMB model is based on the HENIKOFF and HENIKOFF (1992) BLOSUM matrices and is *linearly* informative on a wide range of replacement rates.

3.4 Data Sets

The main aim throughout this work was to investigate the effect of secondary structure elements (such as (1) α-helices, β-sheets, and coils in protein polypeptide chains, and (2) base-pair helices and bulges in RNA strands) on the substitution rates between biological sequence pairs, whereby each pair is considered to have diverted independently and over evolutionary time. To carry out the investigation, protein sequence pairs were sourced from the BAliBASE 3.0 database (THOMPSON *et al.*, 2005), and RNA sequence pairs were sourced from the European ribosomal RNA database (WUYTS *et al.*, 2004).

Protein Model		One-Sample t-test		
$t_1 = 0.4$	$t_1 = 0.6$	$t_1 = 0.8$	**Parameters**	**Settings**
0.1309	0.1865	0.5543	Substitution rate (t_2)	0.20
0.5462	0.2572	0.3318	Indel lengths (a_1, a_2)	0.20, 0.30
0.1585	0.6143	0.8128	Indel rates (r_1, r_2)	0.30, 0.05
0.5202	0.0488	0.7110	Hydrophilicities (h_1, h_2)	0.40, 0.70

Codon Model		One-Sample t-test		
$t_1 = 0.4$	$t_1 = 0.6$	$t_1 = 0.8$	**Parameters**	**Settings**
0.0311	0.1679	0.2265	Substitution rate (t_2)	0.20
0.9150	0.4097	0.0333	Indel lengths (a_1, a_2)	0.20, 0.30
0.0904	0.0196	0.0154	Indel rates (r_1, r_2)	0.03, 0.05
0.0257	0.0743	0.4985	Transition-transversion ratios (κ_1, κ_2)	0.80, 1.20
0.0154	0.1416	0.3043	Darwinian selection ratios (ω_1, ω_2)	0.80, 1.20
0.1620	0.0030	0.0338	Hydrophilicities (h_1, h_2)	0.40, 0.70

DNA Model		One-Sample t-test		
$t_1 = 0.2$	$t_1 = 0.3$	$t_1 = 0.4$	**Parameters**	**Settings**
0.1581	0.1040	0.0799	Substitution rate (t_2)	0.10
0.0945	0.2458	0.9795	Indel lengths (a_1, a_2)	0.30, 0.50
0.4818	0.8004	0.9394	Indel rates (r_1, r_2)	0.05, 0.08
0.8609	0.5089	0.1404	Transition rates (κ_1, κ_2)	1.50, 1.20

Protein Model		Two-Factor ANOVA with $K_{ij} = 24$			
Source of variation	**d.f.**	**Sum of squares**	**Mean square**	**f-test**	**p-value**
FactorA	3	2.42	0.81	1.1969	0.3113
FactorB	2	0.69	0.35	0.5119	0.5999
FactorAB	6	2.71	0.45	0.6701	0.6739
Error	276	186.29	0.68		
Total	287	192.12			

Codon Model		Two-Factor ANOVA with $K_{ij} = 24$			
Source of variation	**d.f.**	**Sum of squares**	**Mean square**	**f-test**	**p-value**
FactorA	5	1035.82	207.16	5.7651	0.0000
FactorB	2	198.68	99.34	2.7644	0.0642
FactorAB	10	395.21	39.52	1.0998	0.3607
Error	414	14876.75	35.93		
Total	431	16506.46			

DNA Model		Two-Factor ANOVA with $K_{ij} = 24$			
Source of variation	**d.f.**	**Sum of squares**	**Mean square**	**f-test**	**p-value**
FactorA	3	93.62	31.21	3.3933	0.0184
FactorB	2	33.41	16.71	1.8165	0.1645
FactorAB	6	68.69	11.45	1.2448	0.2835
Error	276	2538.18	9.20		
Total	287	2733.89			

Figure 3.7: Sets of 12 alignments each were simulated for each biological encoding (BE), namely, protein, codon, and DNA. Parameter values for evolutionary distances (t_1, t_2) were set to $(0.2, 0.4)$, $(0.2, 0.6)$, and $(0.2, 0.8)$ for protein and codon BEs, and to $(0.1, 0.2)$, $(0.1, 0.3)$, and $(0.1, 0.4)$ for DNA BE. All other parameters were set to some sensible value as shown in the three tables on the left. One-sample t-tests were used to test model performance by testing H_o: $x = x_o$ versus H_a: $x \neq x_o$ for each parameter. Two-Factor ANOVAS with $K_{ij} = 24$ (after allowing one parameter at a time to vary across two regions, while all other parameters were held fixed) were used to test for interaction between estimators (FactorAB), model performance across different distances (FactorA), and model performance across all other parameters (FactorB).

From each of these two databases, multiple sequence alignments that were deemed suitable for my experiments were downloaded. From each of these alignments, a phylogeny was constructed in two steps. First, pairwise evolutionary distances were computed using the standard Neighbour-Joining method. Second, Maximum Likelihood was employed to optimise the likelihood function of the tree that had been obtained from the first step.

To each phylogeny, a post-order traversal was applied in order to extract sequence pairs that do not share the same immediate ancestor within the same phylogeny. Figure 3.8 shows an example of a phylogeny constructed from a multiple alignment taken from the BAliBASE database after searching under the unique key BBS12002. Following maximisation of the likelihood function of this tree, a post-order traversal yielded three sequence pairs, namely, (1) RL1_HALVO and RL1_BUCAP with an evolutionary distance of 1.647 and sharing ancestor number 1, (2) R10A_TRYBR and R10A_ENTHI with an evolutionary distance of 0.987 and sharing ancestor number 2, and (3) 1cjs_A and 1mzp_A with an evolutionary distance of 1.121 and sharing ancestor number 3. The distance between each of these three pairs is indicated by the bolded line sections within the phylogeny in Figure 3.8. The important feature of each of these three bolded lines is that each is separate from the other two, and each is bifurcated by a different ancestor. That is, for experimental purposes I can assume that each of the three pairs never shared evolutionary events with either of the other two, and evolved independently over evolutionary time. Hence, I call these units phylogenetically independent pairs (PIPs). PIPs that had been randomly sampled can also be assumed to be independently and identically distributed (i.i.d.), and hence they constitute a data set which is amenable to standard inferential techniques that are based on the Central Limit Theorem and Maximum Likelihood.

3.4.1 The Protein Data Set

A total of 808 protein PIPs were extracted from Reference Sets 1, 2, 3, and 5 of the BAliBASE 3.0 database. I excluded Reference Set 4 because this set contains alignments with very long extensions. This characteristic could be problematic when searching for a "good" alignment. For the same reason, I used only BAliBASE

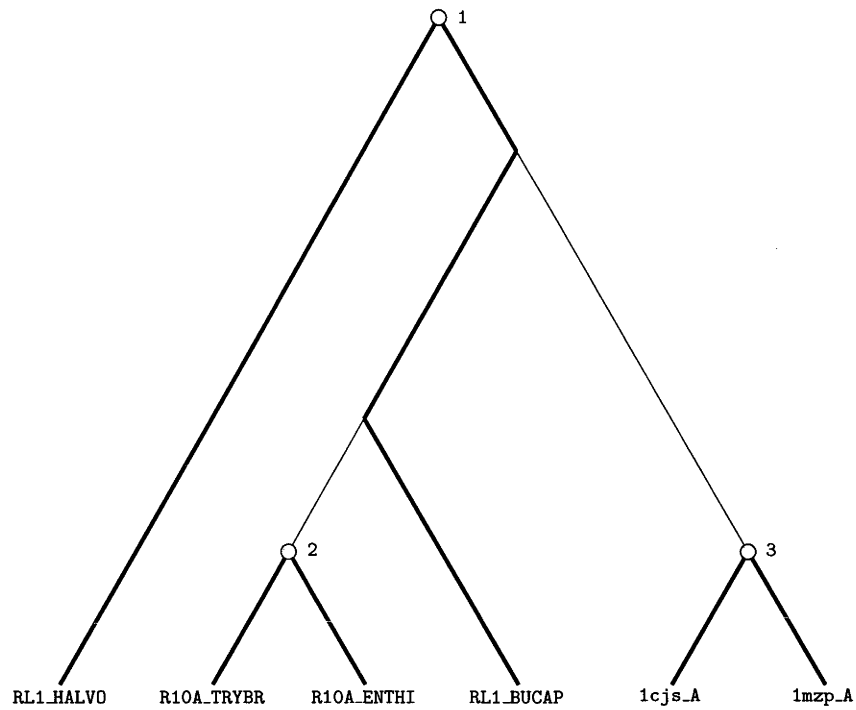


Figure 3.8: The phylogeny is constructed from a multiple alignment stored in the BAliBASE database under the unique key BBS12002. The alignment consists of six protein sequences resulting in six tips. A post-order traversal classified these tips into three pairs whereby each pair does not share the same immediate ancestor, and hence can be considered as having diverged independently for experimental purposes.

truncated alignments under search keys prefixed by BBS in order to construct PIPs. This approach avoids effects due to large end gaps and increases the chance of the optimiser finding global optima, rather than mere local optima, by exploiting homologous domains (THOMPSON *et al.*, 2005). Figure 3.9 shows a boxplot of the relative evolutionary times – or distances – of the 808 PIPs. For the purpose of my experiments, the spread of these distances was too wide, ranging from 0.03 to 2.36 with numerous mild and severe outliers all located at the upper end and hence heavily skewing the data set. The lower and upper fourths show that most PIPs have distances concentrated around a median of 0.71 with a spread of about 0.4 on either side. I decided, therefore, that a suitable range of distances for my experiments would lie approximately between 0.3 and 1.2.

In designing a data set, a good strategy was to subdivide the sample into

bins with a bandwidth of 0.25. A narrow bandwidth would not contain enough PIPs to sample from effectively, while a wide bandwidth could lead to non-uniform sampling. Table 3.1 shows nine bins, starting from the smallest distance. It is clear that bins 2 to 5 (with shaded colour) contain PIPs whose statistics are better in that differences in the means are precisely 0.25, and differences in the standard deviations are uniform and small. At the same time, each of these four bins turn out to have sizes large enough to allow random sampling of 30 PIPs per bin that lead to a large number of potential unique experimental samples. Any of these samples can be constructed in order to allow experiments to be repeated. Figure 3.10 shows boxplots for each of these four bins, whereby each boxplot reveals a wide fourth spread and no outliers in each bin, hence eliminating skewness in each.

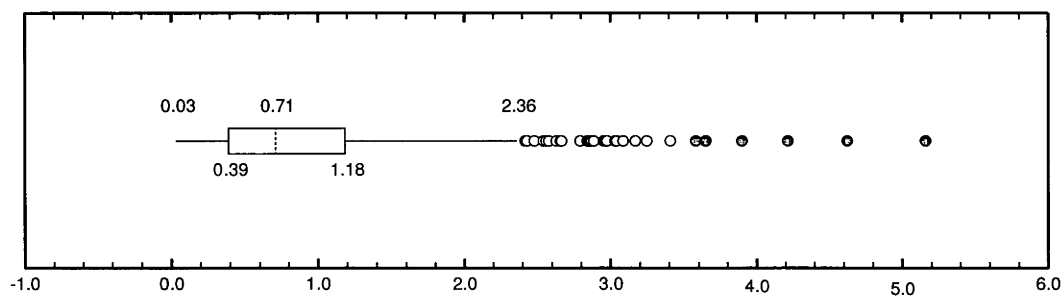


Figure 3.9: The boxplot shows that PIPs extracted from the BAliBASE database have a large fourth spread and a large number of outliers. Lower and upper fourths indicate that a suitable data set would consist of PIPs with distances ranging between 0.25 and 1.25.

3.4.2 The Codon Data Set

Part of the objective in this work was to repeat experiments carried out using protein sequence pairs with codon equivalent sequences. For this purpose I obtained the DNA equivalents of the 808 protein PIPs using a Python script written by Peter Maxwell. Using the script, I accessed the NCBI Protein and Nucleotide sequence databases. Protein records for all truncated sequences extracted from BAliBASE were fetched using protein codes. Cross-references from each protein record were used to identify the corresponding DNA sequence. The coding sequence, identified using the feature table of the DNA record, was extracted. After removing the terminal stop codon, it was then stored in a fasta formatted file. Using this procedure,

Bin	Size	Min	Max	Median	Mean	Diff-M	SD	Diff-SD
1	99	0.027	0.249	0.196	0.176	–	0.061	–
2	185	0.250	0.498	0.378	0.372	0.196	0.075	0.014
3	140	0.502	0.749	0.607	0.618	0.246	0.071	-0.004
4	126	0.750	0.995	0.875	0.871	0.254	0.068	-0.002
5	78	1.004	1.249	1.109	1.124	0.253	0.071	0.003
6	64	1.251	1.497	1.374	1.373	0.248	0.077	0.006
7	40	1.500	1.734	1.611	1.617	0.244	0.072	-0.005
8	25	1.761	1.993	1.839	1.869	0.252	0.077	0.005
9	14	2.009	2.153	2.071	2.078	0.209	0.047	-0.030

Table 3.1: The BAliBASE sample of 808 PIPs yields bins of varying sizes. Nine bins are constructed on the basis of distances, with a bandwidth of 0.25 per bin. Bins 2 to 5 are suitable for experimental purposes since they cover the range of interest, namely, 0.25-1.25. They also present inter-mean differences of precisely 0.25 and present also the smallest inter-standard deviation differences of just ± 0.003 approximately. The size of these four bins also allows a large number of unique experimental samples to be drawn, with 30 PIPs per bin, and hence provide the means for repeating experiments and replicating results.

I collected a total of 1187 DNA sequence equivalents as compared to the 3359 protein truncated sequences stored in BAliBASE 3.0. The final data set of 808 PIPs was then derived from the intersection of these DNA and protein sequence pools following ML maximisation of the protein trees as described in 3.4.1.

3.4.3 The RNA Data Set

Figure 3.11 shows two boxplots of the RNA data set. The first boxplot shows the spread of two sets of PIPs extracted from two RNA trees that had been downloaded from the European ribosomal RNA database. The first tree yielded 74 PIPs while the second yielded 151 PIPs, a total of 225 unique PIPs. However, the spread of the distances of these PIPs was very narrow, just $0.00 - 0.25$ approximately, with severe outliers at each end.

To increase the spread, and to reduce outliers, I decided to prune several times the second tree (since this was larger than the first tree), collecting PIPs with larger distances at each pass. This procedure yielded 105 unique PIPs which, together with the 74 PIPs from the first tree produced the second boxplot. From this boxplot it can be seen that the spread of distances increased to $0.00 - 0.44$ with no outliers. From the 105 PIPs from the second tree, 25 were randomly selected, and together with the 74 from the first tree, an RNA experimental sample of 99 PIPs was constructed.

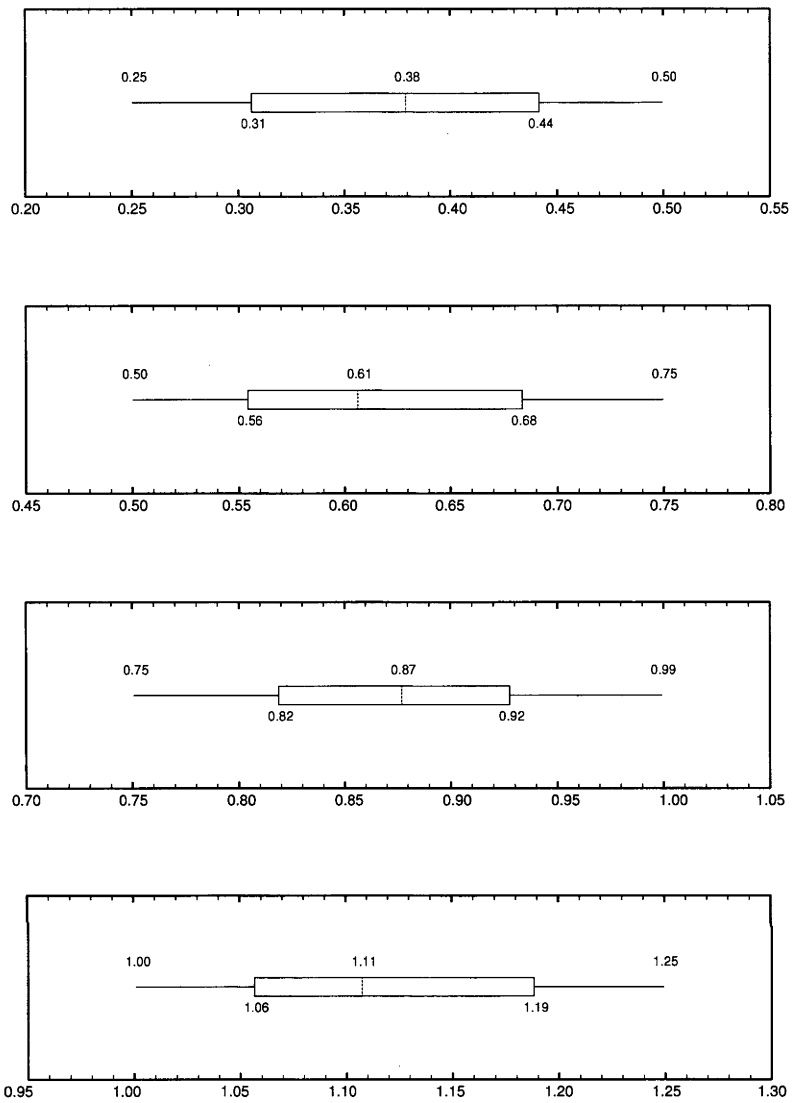


Figure 3.10: Boxplots for bins 2 to 5 from the nine bins shown in Table 3.1. All four bins exhibit no outliers, and exhibit also reasonably large fourth spreads. These properties make these bins suitable for sampling, with 30 randomly selected PIPs per bin.

3.5 The Experimental Setting

The data sets were constructed to carry out experiments for the purpose of testing hypotheses. Some of these hypotheses address the question as to whether there exist two broad classifications, along the DNA or protein, of some element of interest that contributes to evolutionary processes. The most important of these elements is the substitution (or replacement) rate t. Hence, a test can typically be

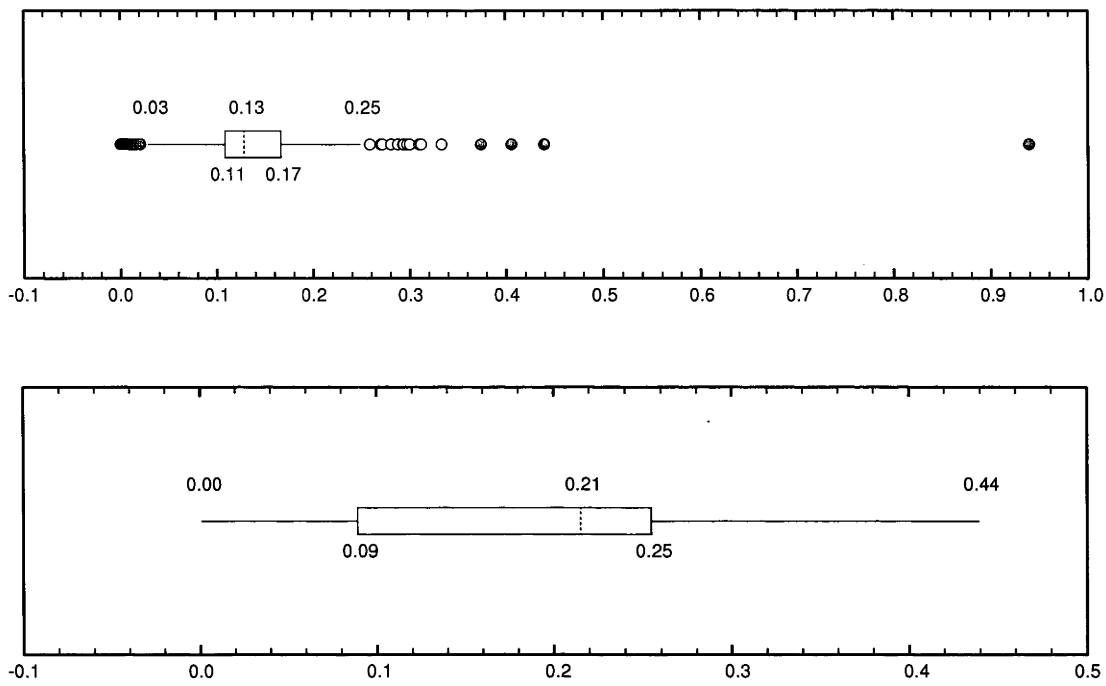


Figure 3.11: The first boxplot shows the spread of RNA PIPs distances before pruning, while the second shows the spread after pruning. The pruning procedure increased the range while eliminating all outliers. Equally important, it moved the median closer to the centre, thus greatly reducing skew in the experimental data.

set as follows. In order to test whether there is a significant difference between t_1 in region one and t_2 in region two, of the two sequences being aligned, define

$$H_o\text{: } t_1 = t_2, \qquad \text{versus} \tag{3.1}$$
$$H_a\text{: } t_1 \neq t_2.$$

To test 3.1, I would maximise the likelihood function 2.23 twice. For protein sequences, first use the vector $\boldsymbol{\theta_o} = (t_1 = t_2, a_1 = a_2, r_1 = r_2, h_1 = h_2, \rho_1 = \rho_2 = 0.5)$ under the null[5]. Then use the vector $\boldsymbol{\theta_a} = (t_1, t_2, a_1 = a_2, r_1 = r_2, h_1 = h_2, \rho_1, \rho_2)$ under the alternative. I set the level of significance at 5%, and compute a p-value with 3 degrees of freedom. The latter is 3 because under H_a, the parameter t is allowed to vary as two independent parts, one in each region, instead of one, thus

[5]Note that under the null the topology is equivalent to a one-region model. Hence, ρ_1 and ρ_2 can also be allowed to vary freely without affecting the degrees of freedom. This would not have any significant effect on the null but would considerably increase computational time.

increasing the degrees of freedom by one. At the same time, region switch parameters ρ_1 and ρ_2 are also relaxed in order to find the best region for the substitution (or replacement) rates at each site of the alignment. This relaxation increases the freedom by a further two degrees.

Let $\mathcal{L}_o$ be the log-likelihood obtained from the likelihood function under the null and $\mathcal{L}_a$ be the log-likelihood obtained under the alternative. It can be shown that a χ^2 statistic can be constructed as

$$\varphi = 2(\mathcal{L}_a - \mathcal{L}_o) \sim \chi^2_{(\alpha=0.05,\, 3)} \tag{3.2}$$

for each alignment.

Since alignments are made from PIPs which are assumed to be i.i.d. (as I explained in Section 3.4), a χ^2 statistic can also be constructed over a set ζ of n alignments whereby each alignment in the set has $\varphi > 0$. This condition would not be satisfied if the optimiser were not able to locate the global maximum for a particular alignment, as I explained in Chapter 2. In most cases, however, the two PHMMs in the two-region model shown in Figure 3.6 produce a smooth surface while maximising the likelihood function. This is due to the nature of the forward dynamic programs that sum over all possible alignments at each iteration. As a result of this property, only a few PIPs (if any) result in $\varphi \leq 0$ and would need to be removed from the set ζ. To test for 3.1 over the set ζ, the χ^2 statistic is now constructed as

$$\varphi_\Sigma = \sum_{i=1}^{n} \varphi_i \sim \chi^2_{(\alpha=0.05,\, 3n)}, \tag{3.3}$$

where n is the number of alignments that have $\varphi > 0$.

3.6 Results

3.6.1 Hypotheses Testing – Protein

Table 3.2 shows nine null hypotheses that I have tested against their corresponding alternatives using the two-region model in Figure 3.6, together with (1)

Test (1)	H_o (2)	H_a (3)	*d.f.* (4)	n (5)	m (6)	φ_Σ (7)	*d.f.*$_\Sigma$ (8)	*p−value*$_\Sigma$ (9)	Conc$_{H_o}$ (10)	Conc$_{H_a}$ (11)
1	$x_1{=}x_2$	$h_1{\neq}h_2$	3	114	10	390.12	342	3.71×10^{-2}	0.9085	0.9081
2	$x_1{=}x_2$	$t_1{\neq}t_2$	3	116	64	1304.99	348	1.30×10^{-110}	0.9425	0.9437
3	$x_1{=}x_2$	$a_1{\neq}a_2$, $r_1{\neq}r_2$	4	93	18	417.96	372	5.01×10^{-2}	0.8960	0.8843
4	$h_1{\neq}h_2$	$h_1{\neq}h_2$, $t_1{\neq}t_2$	1	110	78	1179.88	110	1.24×10^{-178}	0.9455	0.9497
5	$t_1{\neq}t_2$	$h_1{\neq}h_2$, $t_1{\neq}t_2$	1	102	37	353.14	102	2.11×10^{-29}	0.9576	0.9658
6	$h_1{\neq}h_2$	$h_1{\neq}h_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	2	99	29	408.13	198	1.09×10^{-16}	0.9187	0.9052
7	$a_1{\neq}a_2$, $r_1{\neq}r_2$	$h_1{\neq}h_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	1	113	40	384.62	113	2.58×10^{-31}	0.9476	0.9548
8	$t_1{\neq}t_2$	$t_1{\neq}t_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	2	73	4	136.68	146	6.98×10^{-1}	0.9544	0.9606
9	$a_1{\neq}a_2$, $r_1{\neq}r_2$	$t_1{\neq}t_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	1	92	54	776.08	92	8.93×10^{-109}	0.9326	0.9444

Table 3.2: Table shows results of the nine tests which use the protein data set. Column numbers are shown in brackets under each title heading. Columns 2 and 3 show the experimental setting under H_o and H_a, respectively for each test, where the notation is as described in the text. Column 4 shows the number of degrees of freedom applicable for each corresponding test. Column 5 shows the number n of alignments that had $\varphi > 0$. Column 6 shows the number m of alignments from the corresponding n alignments that were significant at the 5% level (p-values with *d.f.* degrees of freedom not shown). Column 7 shows the sum of φ (as per 3.3) across the corresponding n alignments. Column 8 is the product of the corresponding *d.f.* and n. Column 9 is the p-value computed for the χ^2 distributed statistic φ_Σ in column 7 with degrees of freedom equal to *d.f.*$_\Sigma$ in column 8. Columns 10 and 11 show the average concordances of the n alignments with corresponding curated alignments under H_o and H_a, respectively. (Note that these average concordances depend also on m. For example, Conc$_{H_o}$ is different for Tests 1, 2, and 3 because m is different for these three tests.)

the protein data set and (2) the experimental setting. I described the latter two in sections 3.4.1 and 3.5, respectively. $x_a = x_b$ means that, with the exception of $\rho_a = \rho_b = 0.5$, all corresponding parameters in regions a and b are restricted to vary equally in the two regions. A not-equal sign means that the two parameters indexed a and b, along with ρ_a and ρ_b, are allowed to vary freely and independently in regions a and b, respectively. An important aspect of this table lies in the n

values of column 5. These vary from 73 in Test 8 to 116 in Test 2. As I mentioned earlier, the optimiser does not find a global optimum in some cases, and this artifact required me to trim my sample size at each test. The average trim across the nine tests was 15.6% which I consider to be reasonable since, throughout my experiments, (1) I always used the same initial values to reduce computational time, and (2) I always retained the optimiser "default values" to ensure there was no subjective manipulation among alignments. Thus, although I had to prune a small number of alignments in each test, I also ensured that no optimisation bias was introduced among the remaining alignments.

Tests 1 to 3 show that while all other parameters in region one were set equal to their corresponding parameters in region two, the hydrophilicity parameter h in Test 1, the replacement rate parameter t in Test 2, and the indel parameters a and r in Test 3, contributed to a significantly higher likelihood when allowed to vary freely and independently in two regions at each test. From these three tests, I make the following inferences.

Test 1 : More hydrophilic content is present in one part of the molecule than in another. This is as I had expected because it is known that solvency exists more abundantly near the surface rather than at the core.

Test 2 : Replacement rates produce a very clear demarcation between slow and fast rates of replacement along the primary structure of amino acids. This result was also expected, but the extremely small p-value is notable, providing strong evidence that *the difference between rates of replacements in two regions of the molecule is unequivocal.*

Test 3 : Although the evidence is weak, there is also a significant difference between the joint effect of indel length and indel rate (i.e. $a \cup r$) in one region and in the other on the likelihood. This difference has not been quantified in the literature because similar models often omit gaps in the alignments under study. One exception is ClustalW whereby one of its main assumptions is that indels occur more frequently in hydrophilic regions as reported in PASCARELLA and ARGOS (1992). Even in ClustalW, however, quantification

is made indirectly and subjectively with the aim of "improving" the alignment (THOMPSON *et al.*, 1994).

These three tests show that the model components, namely, h, t, and $a \cup r$ play a different role in one region of the molecule than they do in another region of this molecule. They also show that the difference between the two roles is very strong for t but is somewhat weak for h and for $a \cup r$. These weaknesses could be attributed to the data set consisting of protein BE. What these tests do not show is whether the two roles are positionally concomitant along the primary structure of the protein among the three components. For example, I cannot infer whether a region of component h positionally coincides with a region of component $a \cup r$ along the polypeptide.

I designed the remaining tests to investigate collocation among these three components. I define collocation between any two components, whose parameters are allowed to vary freely and independently in two regions under the alternative, to exist if *the levels of the estimators for the two components are both high (or both low) in the same region.*

Test 4 : The number n of alignments that have $\varphi > 0$ decreases when h and t vary freely and independently in two regions simultaneously. This suggests that there is some degree of confounding between h and t in the model, making it harder for the optimiser to find the global maximum. While the optimisation performance is marginally reduced, however, the number m of significant alignments increases from 64 to 78. At the same time, p-value$_\Sigma$ is much lower than that obtained in Test 2. This suggests that collocation between $\hat{h}_\eta$ and $\hat{t}_\eta$, $\eta \in \{1, 2\}$, is most likely.

Test 5 : Again n decreases from 114 to 102, while m increases from just 10 to 37. The latter is a substantial increase, while p-value$_\Sigma$ drops sharply. These results further confirm that the evidence of collocation is strong. That is, *hydrophilicity and replacement rate need to be modelled together in two regions in order to test for a potentially better likelihood when using protein BE.*

Test 6 : Collocation also appears to exist between hydrophilicity and indels. Now, however, n increases from 93 to just 99, providing some indication that these two components do not seem to be confounded. That is, it is somewhat easier for the optimiser to find the maximum likelihood when h and $a \cup r$ vary freely and independently in two regions simultaneously. Furthermore, p-value$_\Sigma$ shows that in the presence of hydrophilicity, indels are easily identifiable, with m increasing from 18 to 29. This is an improvement on the result I obtained in Test 3 where I allowed $a \cup r$ to vary freely and independently while restricting all other components to vary equally in two regions. In Test 3, the evidence on the basis of the p-value$_\Sigma$ was weak.

Test 7 : Similarly, n remains essentially the same, while m increases from just 10 to 40 and p-value$_\Sigma$ drops sharply. The results obtained from Tests 6 and 7 are consistent with results reported in PASCARELLA and ARGOS (1992) where it is shown that indels are more likely to be found in solvent regions.

Tests 8 and 9 : In a similar vein, replacement rates and indels varying freely and independently in two regions are confounded – note the large drop in n from 116 in Test 2 down to 73 in Test 8. There is insufficient evidence here to suggest that collocation between t and $a \cup r$ exists.

3.6.2 Replacement Rates in Hydrophilic Regions

In this section I investigate further the hypothesised collocation between hydrophilicity and replacement rates. For this purpose I define a new test as follows

$$\begin{aligned} H_o&: x_1 = x_2, \qquad \text{versus} \\ H_a&: h_1 \neq h_2,\ t_1 \neq t_2. \end{aligned} \tag{3.4}$$

The results from test 3.4 were $n = 120$, $m = 63$, and p-value$_\Sigma = 4.39 \times 10^{-113}$ with $d.f._\Sigma = 480$. Furthermore, I counted the number of times, among the $m(= 63)$ alignments under H_a, the feature of interest, namely, *fast replacement rates and high hydrophilicity are collocated*, occurred. For notational convenience, I denote this feature by $\varsigma_{(h,t)}$.

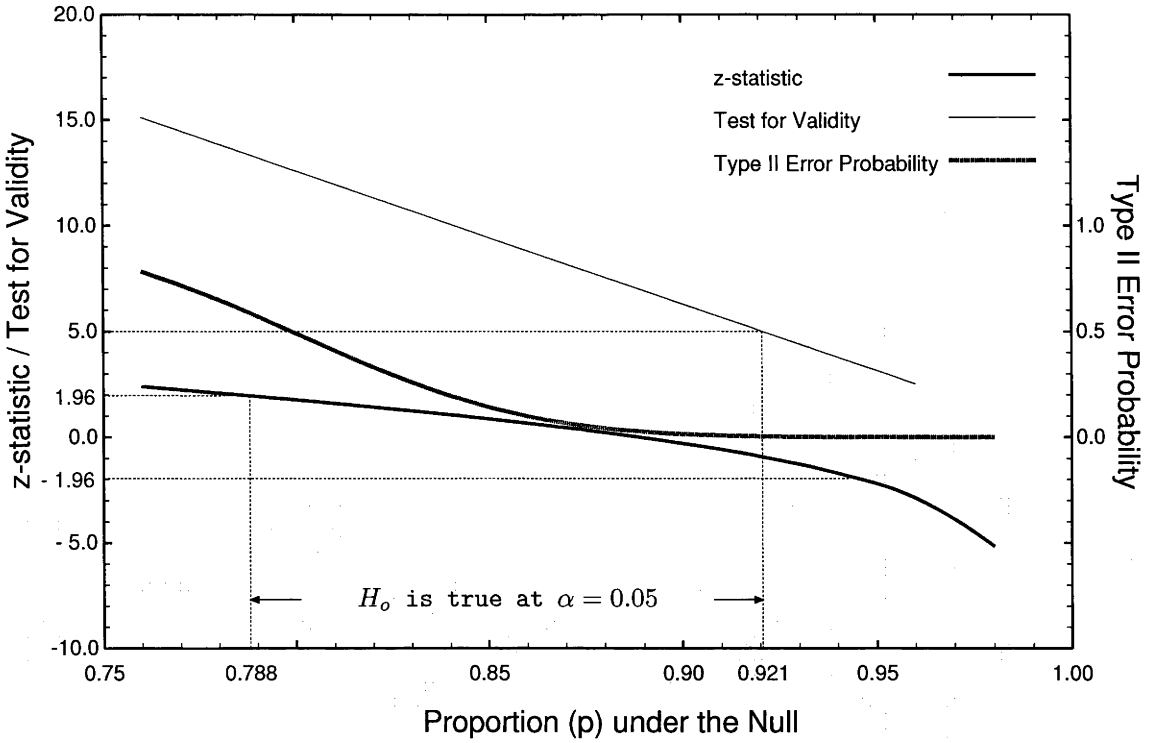


Figure 3.12: The graph illustrates a statistical test whereby the null, namely, the population proportion $p = p_o$ of PIPs possess a specified quality, is true when p lies between 79% and 92%. The validity rule (i.e. DEVORE, 1990, p. 308) is applied due to a reduced sample size after testing for significance of the substitution rate and of the hydrophilicity parameters during pairwise alignment. Type II error against the alternative hypothesis, namely H_a: $p = 0.70$, is also shown.

The graph in Figure 3.12 illustrates the statistical test concerning the population proportion p of protein PIPs that have the feature $\varsigma_{(h,t)}$. The null hypothesis is stated as $p_o\%$ *of PIPs have this feature*, and the alternative is stated as *The null is untrue.* From a random sample of $m = 63$ PIPs, $X = 56$ were found to have the feature $\varsigma_{(h,t)}$, where X is assumed to have approximately the binomial distribution. Considering that m is large, both X and $\hat{p} = X/m$ are also approximately normally distributed with $E(\hat{p}) = p$ and $\sigma_{\hat{p}} = \sqrt{p(1-p)/m}$. When H_o is true, $E(\hat{p}) = p_o$, $\sigma_{\hat{p}} = \sqrt{p_o(1-p_o)/m}$, and the test statistic is

$$z = \frac{\hat{p} - p_o}{\sqrt{p_o(1-p_o)/m}} \sim N(0,1),$$

(i.e. DEVORE, 1990, p. 308). The test is valid only if both mp_o and $m(1-p_o)$ are equal or greater than 5. As can be seen in Figure 3.12, the sample of 63 observations is not large enough to span the potential range of H_o not being rejected. The validity

rule slightly truncates the right hand side of the range of proportions that can be hypothesised to be true under the null. Nevertheless, the sample is large enough to allow me to infer that *conditional on the replacement rate and the hydrophilic content present in the molecule being statistically significant, a very high percentage of protein sequences – approximately between* 80% *to* 90% *– exhibit collocation of these two components that contribute to evolutionary processes.* That is,

> **under the assumption of regional heterogeneity of substitution rates, high substitution rates in coding DNA are mostly to be found on the surface of the molecule which is more amenable to water and furthest from the core.**

Type II error probabilities, (e.g. DEVORE, 1990, p. 309), after setting the alternative hypothesis to some reasonable level, say, H_a: $p = 70\%$, are also shown in Figure 3.12. My choice of 70% is conservative. I am safe to assume that if the null turned out to be untrue, the probability p being as small as 70% is very small, and the probability p being less that 70% would be even smaller still. Another alternative would be p greater than 92%, but for the purpose of my investigation, this range is not interesting.

3.6.3 Indels in Hydrophilic Regions

To investigate collocation between hydrophilicity and indels, I defined the following test

$$\begin{aligned} &H_o\text{: } x_1 = x_2, \qquad \text{versus} \\ &H_a\text{: } h_1 \neq h_2,\ a_1 \neq a_2,\ r_1 \neq r_2. \end{aligned} \tag{3.5}$$

The results from this test were $n = 114$, $m = 23$, and $p\text{-value}_\Sigma = 1.16 \times 10^{-8}$ with $d.f._\Sigma = 570$. That is, the percentage of significant alignments was just 20%, and this is much lower than that obtained from 3.4, which was just over 50%. Indels, therefore, are not as heterogeneous as replacement rates. It was interesting, therefore, to examine indel lengths and indel rates separately.

Among the 23 significant alignments, 20 had the feature $\varsigma_{(h,r)}$ and just 10 had the feature $\varsigma_{(h,a)}$. An upper-tailed sign test gave a p-value of 0.00024 for the

former feature, and 0.7976 for the latter. This means that *high indel rates, like high replacement rates, are collocated with the solvent regions of the molecule, but indel lengths, whether short or long, are not.*

These two results are consistent with two important findings reported in PASCARELLA and ARGOS (1992), and which I discussed in Section 2.4. The first was that indel lengths have a tendency to reach equilibrium. That is, the evolutionary process of insertions and deletions takes place in a balanced manner, irrespective of how much evolution had taken place. In the two-region context, this means that indel lengths would reach saturation whether they are located in regions experiencing slow, or in regions experiencing fast, replacement rates. This property makes it difficult for the optimiser to distinguish indels in slow regions – core – from indels in fast regions – solvent – of the pairwise alignment, thus resulting in less alignments possessing the feature $\varsigma_{(h,a)}$.

The second important result in PASCARELLA and ARGOS (1992) was that indels are more tolerated in those regions which are more solvent, that is, indels occur more often in regions that consist of turn and coil structures. This makes it easier for the optimiser to distinguish indel rates that are fast in one region from those that are slow in the other region, resulting in more alignments with the feature $\varsigma_{(h,r)}$.

To confirm these two results, I defined one last test, namely,

$$\begin{aligned} &H_o\colon x_1 = x_2, \qquad \text{versus} \\ &H_a\colon t_1 \neq t_2,\ a_1 \neq a_2,\ r_1 \neq r_2. \end{aligned} \tag{3.6}$$

The results from this test were $n = 101$, $m = 42$, and $p\text{-value}_\Sigma = 2.10 \times 10^{-57}$ with $d.f._\Sigma = 505$. Thus, of the 42 significant alignments, 31 had the feature $\varsigma_{(t,r)}$, and just 19 had the feature $\varsigma_{(t,a)}$. For the former, a sign test gives a p-value of 0.0014, and for the latter, a p-value of 0.322. These two results are expected since I had established that fast replacement rates are collocated with solvency. That is, *indel rates can be expected to behave similarly when observed in fast replacement regions and in hydrophilic regions of protein sequences.*

3.6.4 Hypotheses Testing – Codon

I repeated the protein experiments using (1) the same data set in codon BE and (2) replacing the PMB model with the GY94 model – the latter I described in Section 2.2. In order to reduce the number of degrees of freedom, I set $\omega = 1.0$ in the GY94 model throughout the codon experiments. I explain further on this setting and deal with ω varying freely and independently in two regions in Chapter 4.

The results obtained from the codon alignments are shown in Table 3.3. As in the protein experiments, all three components, namely, hydrophilicity, substitution rates, and indels, are significantly different between two regions. Note especially the zero measures of the p-values$_\Sigma$ in column 9 for $\hat{h}_\eta$ and for $\hat{t}_\eta$, $\eta \in \{1, 2\}$. Also, the significance of the joint effect of $\hat{a}_\eta$ and $\hat{r}_\eta$ varying freely and independently in two regions is now clearer. Finally, the transition-transversion rates ratio estimators $\hat{\kappa}_\eta$ have no statistical significance, and hence the κ parameter will not be considered further in a two-region context.

3.6.5 Collocations – Codon

Tests of hypotheses on collocation, using the codon data set, are summarised in Table 3.4. This table shows that with the GY94 model, only the feature $\mathbf{\varsigma}_{(t,h)}$ comes out as significant. This can be attributed to the fact that confounding is strong when using this model, as I mentioned earlier. This model has the disadvantage of not allowing the optimiser to detect collocation effectively. This point is further illustrated in Figure 3.13 which shows that the proportion of alignments that have the feature $\mathbf{\varsigma}_{(t,h)}$ varies over a range of percentages that are much lower than those shown in Figure 3.12. Note also that Type II error probabilities, with H_a: $p = 55\%$, rise rapidly as the range approaches the alternative.

3.6.6 Model Dependence

When nesting hypothesis, the codon data set yielded m:n ratios that were different from those that had been obtained with the protein data set. These ratios, together with the corresponding p-values$_\Sigma$ (shown in brackets), are summarised in Table 3.5 both for protein and codon BEs.

Test (1)	H_o (2)	H_a (3)	*d.f.* (4)	n (5)	m (6)	φ_Σ (7)	*d.f.*$_\Sigma$ (8)	$p-value_\Sigma$ (9)	Conc_{H_o} (10)	Conc_{H_a} (11)
1	$x_1{=}x_2$	$h_1{\neq}h_2$	3	119	117	8173.09	357	0.00	0.9327	0.9465
2	$x_1{=}x_2$	$t_1{\neq}t_2$	3	119	108	3630.86	357	0.00	0.9283	0.9319
3	$x_1{=}x_2$	$a_1{\neq}a_2$, $r_1{\neq}r_2$	4	103	29	738.04	412	7.88×10^{-21}	0.8712	0.8444
4	$x_1{=}x_2$	$\kappa_1{\neq}\kappa_2$	3	120	3	130.68	360	1.00	0.7341	0.7391
5	$h_1{\neq}h_2$	$h_1{\neq}h_2$, $t_1{\neq}t_2$	1	113	53	976.26	113	1.98×10^{-137}	0.9306	0.9487
6	$t_1{\neq}t_2$	$h_1{\neq}h_2$, $t_1{\neq}t_2$	1	118	109	5359.65	118	0.00	0.9352	0.9458
7	$h_1{\neq}h_2$	$h_1{\neq}h_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	2	94	5	259.53	188	4.24×10^{-4}	0.7670	0.8932
8	$a_1{\neq}a_2$, $r_1{\neq}r_2$	$h_1{\neq}h_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	1	117	103	6151.61	117	0.00	0.9312	0.9479
9	$t_1{\neq}t_2$	$t_1{\neq}t_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	2	98	4	140.01	196	9.99×10^{-1}	0.8729	0.8837
10	$a_1{\neq}a_2$, $r_1{\neq}r_2$	$t_1{\neq}t_2$, $a_1{\neq}a_2$, $r_1{\neq}r_2$	1	119	104	2769.99	119	0.00	0.9100	0.9315

Table 3.3: Table shows ten tests using the codon data set and the experimental setting. The notation is the same as in Table 3.2. Column numbers are shown in brackets under each title heading. ω was set to 1.0 in each test.

Ratios for protein BE increased each time I changed the nested hypotheses. For example, starting with the nested hypothesis H_{nested}: $x_1 = x_2$ shown in columns 2 and 4, I obtained m:n ratios 0.0877 and 0.5517, respectively. Each of these two ratios increased substantially when I changed the nested hypothesis. The first ratio increased from 0.0877 to 0.3627 when I changed the nested hypothesis to H_{nested}: $t_1 \neq t_2$ (column 3). The second ratio increased from 0.5517 to 0.7091 when I changed the nested hypothesis to H_{nested}: $h_1 \neq h_2$ (column 5). For the codon BE, however, the reverse is true. The two ratios in columns 3 and 5 can be seen to have decreased; the second one substantially. A similar behaviour was observed when I

Feature	m:ς	p-value
$\varsigma_{(t,h)}$	120:83	1.62×10^{-5}
$\varsigma_{(h,a)}$	111:35	9.99×10^{-1}
$\varsigma_{(h,r)}$	111:62	1.27×10^{-1}
$\varsigma_{(t,a)}$	96:40	9.59×10^{-1}
$\varsigma_{(t,r)}$	96:43	8.69×10^{-1}

Table 3.4: Collocations computed from the codon data set.

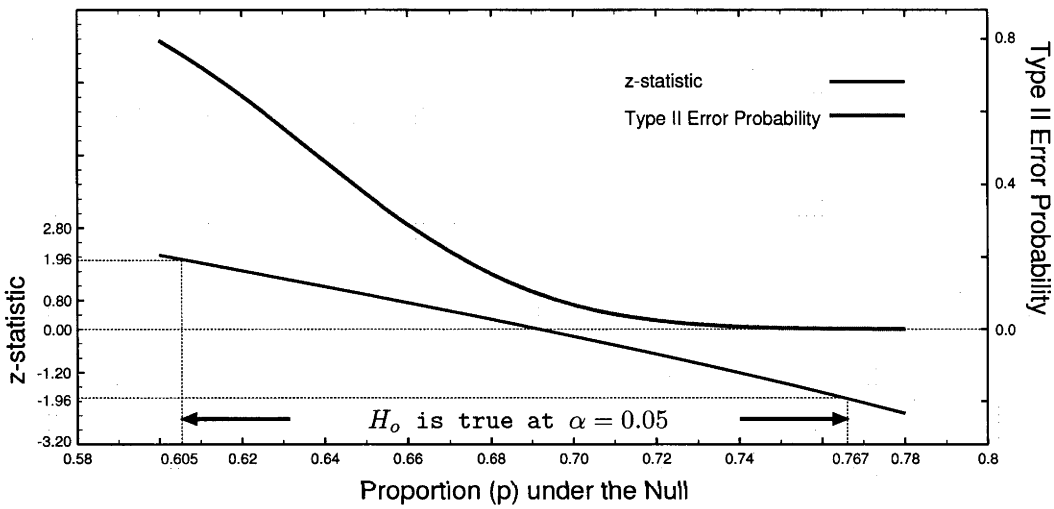


Figure 3.13: Statistical test concerning the population proportion p of protein PIPs that have the feature $\varsigma_{(h,t)}$, using the codon data set. Type II error against the alternative hypothesis, namely H_a: $p = 0.55$, is also shown. The validity rule is omitted because m =120 is a large sample and makes the rule unnecessary.

nested hypotheses using indel parameters (summaries not shown).

It is clear that inference drawn on parameters that are allowed to vary freely and independently in two regions is model dependent. This conclusion is based on the fact that I have used the same data set, and the same experimental setting, with each of the two models, namely, PMB and GY94. This model dependency is explained by the fact that the two models are structurally different as I explained in Chapter 2. The PMB model is in its great part an empirically derived model, using BLOSUM matrices, while the GY94 is a mechanistic model. The PMB model is based on averages that span a wide range of evolutionary distances and therefore

BE	$x_1=x_2$ $h_1\neq h_2$	$t_1\neq t_2$ $t_1\neq t_2, h_1\neq h_2$	$x_1=x_2$ $t_1\neq t_2$	$h_1\neq h_2$ $h_1\neq h_2, t_1\neq t_2$
(1)	(2)	(3)	(4)	(5)
Protein	0.0877 (3.71×10^{-2})	0.3627 (2.11×10^{-29})	0.5517 (1.30×10^{-110})	0.7091 (1.24×10^{-178})
Codon	0.9832 (0.00)	0.9237 (0.00)	0.9076 (0.00)	0.4690 (1.98×10^{-137})

Table 3.5: Table shows m:n ratios obtained from each of the four tests, using protein and codon BEs. Values in brackets are corresponding p-values$_\Sigma$. Column numbers are shown in brackets under each title heading.

it is less sensitive to model parameters. That is, information on evolution which the model extracts depends more on its own specification and less on parameter values. This feature makes it easier for the optimiser to find the best likelihood at each alignment since much of the information is provided *a priori*.

3.6.7 Hypotheses Testing – RNA

Table 3.6 shows the results of three hypotheses using the RNA data set. Test 1 shows once more a clear demarcation between slow and fast regions, with 91% of the alignments showing significance. p-value$_\Sigma$ of these alignments is zero, suggesting that the distinction between the two rates is unequivocal and essentially ubiquitous.

Test	H_o	H_a	*d.f.*	n	m	φ_Σ	*d.f.*$_\Sigma$	$p-value_\Sigma$	Conc_{H_o}	Conc_{H_a}
1	$x_1=x_2$	$t_1\neq t_2$	3	98	90	5085.40	294	0.00	0.9555	0.9639
2	$t_1\neq t_2$	$t_1\neq t_2$, $r_1\neq r_2$	1	87	7	140.23	87	2.62×10^{-4}	0.9721	0.9704
3	$x_1=x_2$	$t_1\neq t_2$, $r_1\neq r_2$	4	98	86	5163.10	392	1.30×10^{-110}	0.9543	0.9630

Table 3.6: Table shows three tests using the RNA data set and the experimental setting. The notation is the same as in Table 3.2.

Test 2 shows that there is confounding between indel rate and substitution rate when they are allowed to vary freely and independently in two regions simultaneously, with n dropping substantially from 98 to 87. Although p-value$_\Sigma$ shows clearly that indel rates varying freely and independently in two regions are distinct between the two regions, this distinction is not common among alignments since m

is just 7. That is, only 8% of the alignments that expressed $\varphi > 0$ were significant in this sample of 99 alignments.

In these tests, using the RNA data set, I did not test for the indel length varying freely and independently in two regions. The previous tests had shown that there is no evidence of collocation between the substitution rate and the indel length, and hence I decided to omit testing for whether indel lengths varying freely and independently in two regions contribute significantly to the likelihood. I also did not include $r_1 \neq r_2$ in H_o. The reason for this is that from Test 10 in Table 3.3 it is clearly shown that nesting $r_1 \neq r_2$ did not have any effect on the likelihood. Considering that r and t are multiplied together, or "coupled", in the KM equations, the impact is derived solely from the substitution rate, and further testing would be unnecessary.

The purpose of Test 3 was to test for collocation of fast substitution rates with fast indel rates. Of the 86 significant alignments ($m = 86$), only 41 had the feature $\varsigma_{(t,r)}$. A sign test gave a p-value of 0.705, thus showing clearly that the two rates are not collocated. I conclude, therefore, that *collocation seems to be a property solely of the hydrophilicity component in protein data.*

3.6.8 Concordances

For each test in Tables 3.2, 3.3, and 3.6, concordances were computed and averaged only for the m alignments that were significant at the 5% level. (This is the reason, for example, $Conc_{H_o}$ is different for Tests 1, 2, and 3 in Table 3.2.) Also, for each test, the average concordances were computed twice, that is, under H_o and under H_a, each time using corresponding estimators. This regime provided me with a practical measure of by how much additional parameters varying freely and independently in two regions may (or may not) improve alignment "quality". For example, from Tests 1, 5, and 7 in Table 3.2, I can reasonably assume that allowing component h to vary freely and independently in two regions on its own is not likely to improve quality (Test 1). However, quality appears to improve in the presence of component t (Test 5) and in the presence of components a and r (Test 7).

To compute the concordance of an alignment, the column score (CS) defined in THOMPSON *et al.* (1999a) was used. That is, $CS = \frac{1}{M}\sum_{i=1}^{M} C_i$, where M is the number of sites in the test alignment, and C_i is 1 if site i, $i = 1, 2, \ldots, M$, is the same as the corresponding reference site, else C_i is 0. In both Tables 3.2 and 3.3, the best average concordance occurs when the substitution rate and the hydrophilicity parameters are optimised simultaneously under the two-region assumption, namely, Test 5 under H_a. Here the average concordance is 95 to 96 %. This is a good result when compared to what is often reported in the literature. EDGAR (2004), for example, reported that multiple aligners MUSCLE, MUSCLE-p, T-Coffee, and ClustalW, all performed at about the 88% mark when benchmarked with BAliBASE curated alignments. The disadvantage of these aligners is that they are based on heuristics. My two-region model, on the other hand, is a dynamic programming algorithm that constructs the alignment on the basis of ML estimators which, under regulatory conditions, are asymptotically efficient (e.g. GREENE, 1997, p. 133). The only limitation of my two-region model lies in the trace-back procedure which selects a "good" alignment from among many possible good alignments but not necessarily the "true" alignment.

The true alignment is a random variable, and hence it is also unobservable. For this reason, concordance measures should be treated only as a guide. All alignments, including curated alignments such as those stored in BAliBASE, are statistics which only try to be as close as possible to the true alignment. We cannot know which of these alignments is the closest to the true alignment. My aim was to build a model based on maximum likelihood (ML). This approach allowed me to obtain ML estimators which, given that the biological sequences are long enough, can be expected to be efficient under the usual regulatory conditions (e.g. GREENE, 1997, p. 133). This feature is the linchpin of my alignments. I can assume, therefore, that my pairwise alignments are the best that one can possibly construct given the data. On the basis of this assumption, I can postulate that when using protein or codon BEs, a good approach to obtaining the best possible pairwise alignment would be as follows.

First, model either the substitution rate parameter or the hydrophilicity pa-

rameter to vary freely and independently in two regions, while each of the other parameters is forced to vary equally in these two regions. Define this alignment as the null alignment. Then model both the substitution rate parameter and the hydrophilicity parameter to vary freely and independently in the two regions simultaneously, while each of the other parameters is forced to vary equally in the two regions. If a likelihood ratio (LR) test shows that this alignment is significantly better than the null alignment, then consider this alignment as the "best" alignment under the two-region assumption.

From Table 3.6, the highest concordance was achieved in Test 2 under H_o. It should not be hard to see that this does not mean that alignments in Test 2 under H_o were the best alignments. What it actually means is that the curated alignments are further from the "true" alignments than the alignments obtained in Test 2 under H_a.

CHAPTER 4
Further Results

4.1 Evolutionary Rates Distributions

To investigate the distribution of substitution rates in each region, I plotted histograms of the slow and fast substitution rates point estimators obtained from the 108 significant PIPs of the codon experiment under Test 2 in Table 3.3.

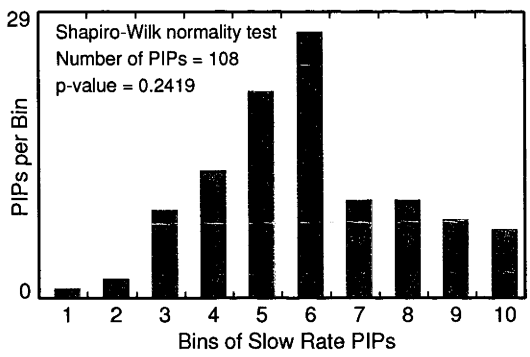


Figure 4.1: The histogram with ten bins of the 108 slow substitution rates point estimators obtained under Test 2 in Table 3.3.

The histogram of slow substitution rates point estimators is shown in Figure 4.1. This plot reveals that the distribution appears to be normal. A Shapiro-Wilk test showed that the null hypothesis should be retained, with a p-value of 0.2419. That is, there is evidence to suggest that slow substitution rates point estimators from my sample of 108 PIPs are normally distributed.

A similar plot (shown in panel 1 of Figure 4.2) of fast substitution rates point estimators reveals that the distribution is clearly not normal. A Shapiro-Wilk test now had a p-value of just 0.0138 after taking the natural logs, thus strongly rejecting the null. This was expected because fast substitution rates estimators were highly erratic throughout my experiments in Chapter 3. Another observation was that some of these estimators exceeded the upper limit of 50.0, which I had set arbitrarily during the experiment.

Panel 1 of Figure 4.2 shows the fourth test of a series of ten tests that I carried out for fast substitution rates estimators. The results from these ten tests

Test	Trim	PIPs	p-value
1	50	89	0.0002
2	45	84	0.0018
3	40	82	0.0034
4	35	79	0.0138
5	30	76	0.0723
6	25	72	0.5893
7	20	70	0.9191
8	15	67	0.9345
9	10	63	0.3773
10	5	43	0.0384

Table 4.1: Ten tests for normality of the natural log of fast substitution rates point estimators were carried out. PIPs were trimmed at each test, eliminating those that had fast substitution rates estimators higher than the pre-set level shown in column two. p-values were obtained using the Shapiro-Wilk test.

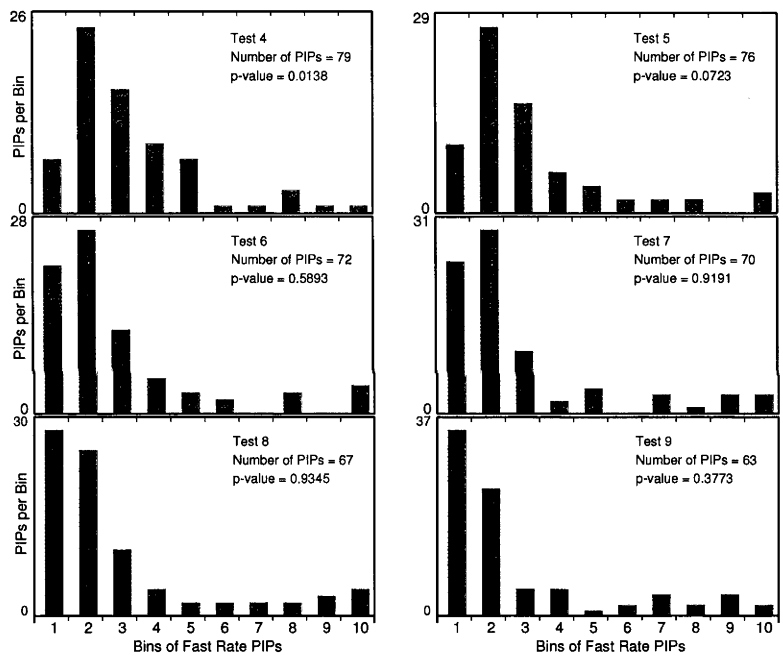


Figure 4.2: Ten tests were carried out to investigate the distribution of fast rates. The six panels show histograms for those among the ten tests, namely Tests 4 to 9, that had the highest Shapiro-Wilk test statistic in Table 4.1 with sufficiently high number of PIPs.

are listed in Table 4.1. At each test, I trimmed the PIP data set so that all PIPs remaining had fast substitution rates estimators that were not greater than a pre-set level. These pre-set levels are shown in column two of Table 4.1. An inspection of this table reveals that the p-values obtained from the Shapiro-Wilk statistic increase substantially under Tests 6 to 9. The corresponding histograms in Figure 4.2 appear

to take the shape of a lognormal distribution.

The largest p-value is obtained under Test 8. The corresponding panel (the fifth panel) in Figure 4.2 shows distinctively five bars – starting from the left – decreasing monotonically with about one half of the distribution concentrated in the first two bars. This strongly suggests a lognormal distribution with $\sigma \approx 1$. At this largest p-value, the pre-set level was set to 15.0.

Region	Distribution	$\hat{\mu}$	$\hat{\sigma}$
1	Normal	0.58	0.22
2	Lognormal	1.42	0.56

Table 4.2: Moments estimators for the Normal (Lognormal) distributions of slow (fast) rates point estimators. PIPs that had fast rates higher than 15.0 were removed from the final sample.

Considering that my analyses are based on point estimators and not on observed data, I cannot infer what the limiting distributions of slow and fast substitution rates would be. The correlation between the 108 slow and fast estimators is small, just 0.34, and a tentative conclusion would be that *the two distributions are independent (or weakly dependent), and normal (lognormal) for slow (fast) rates.* On the basis of my sample, these distributions have first and second moments estimated as shown in Table 4.2, after removing PIPs that had fast rates higher than 15.0. This is because I consider rates that are higher than this level to be non-informative, that is, they are artifacts of the optimiser searching along a flat surface.

4.2 Optimising ω in the Two-Region Model

During the initial development of my two-region model I discovered that changing the ω parameter in the GY94 substitution model along with the substitution rate parameter t, was leading to unexpectedly high estimators for ω. This compelled me to fix ω to 1.0 throughout the experiments that were based on the codon data set.

In hindsight, it is not hard to see that the ω estimator can be expected to have a distribution which is very different from that of the t estimator. The parameters ω and t were designed to deal with two very different evolutionary processes. I also suspected that these two processes are highly interdependent. Unlike the parameters

in the KM equations, I had no theory that could enable me to relate ω with t in my likelihood function. For this reason, it was meaningless to allow for these two parameters to vary together during the optimisation of this function.

MURPHY and TOPEL (1985) describe two-step estimation procedures that can overcome this type of problem. One of these procedures allows for both the auxiliary and the second-step models to be estimated by maximum likelihood. They propose that the marginal distributions of the two random vectors y_1 and y_2 (or, in my case, the two random alignments $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively) can be stated as $F_1(y_1;\theta_1)$ and $F_2(y_2;\theta_1,\theta_2)$, where θ_1 and θ_2 are to be estimated from the data. Under this formulation, the two-step procedure to maximise 2.23 can be stated as follows:

$$\text{Step One} \qquad \sum_{n=1}^{N} \frac{\delta \mathcal{L}_1(\mathcal{A}_{1n};\hat{\boldsymbol{\theta}}|\omega_1=\omega_2=1.0,\mathcal{S}_1,\mathcal{S}_2)}{\delta\theta} = 0, \tag{4.1}$$

$$\text{Step Two} \qquad \sum_{n=1}^{N} \frac{\delta \mathcal{L}_2(\mathcal{A}_{2n};\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\omega}}|\mathcal{S}_1,\mathcal{S}_2)}{\delta\omega} = 0. \tag{4.2}$$

In 4.1, I fix ω of the GY94 substitution model to 1.0, and hence ω is considered not to be part of the parameter set. In 4.2, ω is allowed to vary either in a one-region (the null) or in a two-region (the alternative) setting, with other parameters held fixed at their corresponding estimated levels computed in Step One.

Under the usual regulatory conditions, the Step One maximum likelihood $\hat{\boldsymbol{\theta}}$ is consistent (e.g. MURPHY and TOPEL, 1985, p. 377). It can also be shown that maximising $\frac{1}{N}\sum \mathcal{L}_2(\mathcal{A}_{2n};\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\omega}})$ with respect to ω is asymptotically equivalent to maximising $\frac{1}{N}\sum \mathcal{L}_2(\mathcal{A}_{2n};\hat{\boldsymbol{\theta}}^*,\hat{\boldsymbol{\omega}})$, where $\hat{\boldsymbol{\theta}}^*$ is the vector of ML estimators obtained from Step One and is held fixed during optimisation in Step Two. Asymptotically, therefore, $\hat{\boldsymbol{\omega}}$ is also consistent. By "asymptotically" here I mean that if the two sequences $\mathcal{S}_1$ and $\mathcal{S}_2$ of PIP_j, $j = 1,2,\ldots,108$ (under Test 2 in Table 3.3), are long enough, I can assume that $\hat{\boldsymbol{\omega}}$ is consistent without any adverse effect on inference.

4.2.1 Two-Step Estimation of ω

To implement the two-step estimation procedure for estimating $\boldsymbol{\omega}$, I re-used the 108 alignments that I had obtained from Test 2 under H_a in Table 3.3. ML estimators from each of these alignments constitute the vector $\hat{\boldsymbol{\theta}}$ for 4.1 of Tests 1 and 2 shown in Table 4.3. To carry out Step Two estimations, alignments were first re-estimated under Test 1, where ω was allowed to vary under H_a but was kept equal across the two regions. Alignments were then re-estimated again under Test 2 where ω was now allowed to vary freely and independently between the two regions under H_a. Under Test 1, 78 alignments were significant at the 5% level, whereby p-values were computed with one degree of freedom. Under Test 2, 76 alignments were significant at the 5% level, whereby p-values were computed with three degrees of freedom. These results are summarised in Table 4.3.

Test	H_o	H_a	*d.f.*	n	m	φ_Σ	dof_Σ	$p-value_\Sigma$
1	$\omega_1=\omega_2=1.0$	$\omega_1=\omega_2$	1	80	78	10870.0	80	0.00
2	$\omega_1=\omega_2$	$\omega_1\neq\omega_2$	3	78	76	4564.7	234	0.00

Table 4.3: Results from Step Two estimations of ω. From the 108 alignments (that were obtained under H_a of Test 2 in Table 3.3), 80 expressed a χ^2 statistic greater than zero under Test 1 of the table above. p-values were computed with one degree of freedom, and 78 of these were significant at the 5% level. Under Test 2, all of the remaining 78 alignments expressed a χ^2 statistic greater than zero, p-values were computed with three degrees of freedom, and 76 of these were significant at the 5% level.

This shows that in 76 alignments, out of 120 alignments of my codon sample, *the natural selection parameter ω played a statistically significant role in the slow and fast rate regions.* Of these 76 alignments, only two expressed an ω estimator with a level higher than 1.0, and both were located in the fast rate region, as shown in Table 4.4. In both alignments, the fast substitution rate point estimator reached the upper limit of 50.0 which I had pre-set during the experiment. As I had stated in Section 4.1, this high level is an artifact of the optimiser, and the actual substitution rate can be considered to be about 15.0.

In both alignments that had $\omega_2 > 1.0$, the level of the estimator was only slightly higher than one, while all other ω estimators in the 76 alignments were mostly very small, the highest level being just 0.4. It would be tempting to suggest,

Aln	t_1	t_2	ω_1	ω_2
88	0.6299	50.0	0.0517	1.1986
117	0.9700	50.0	0.0877	1.1000

Table 4.4: Two alignments expressed $\omega_2 > 1.0$ following a two-step estimation.

therefore, that positive selection was detected under weak selection in two cases. Whether these are true positives, however, is not the central issue here. My main aim in this experiment was to determine, for reasons I explained at the start of this section, whether ω could be estimated separately from all other parameters, using a two-step estimation approach. In addition, I wanted to ascertain whether ω varying freely and independently in two regions under H_a would increase the likelihood by a significant amount, hence showing that selection can be detected in a two-region context. My results here show that under the assumptions of the GY94 model at least, ω does play a significant role in this setting.

Admittedly, the GY94 is a purely *mechanistic* model (KOSIOL *et al.*, 2007), in the same way the PMB is a purely *empirical* model as I illustrated in Chapter 2. The ω parameter in the GY94 model is interpreted strictly as a rate ratio. That is, it represents the *absolute* nonsynonymous-synonymous rate ratio (KOSIOL *et al.*, 2007). What I would need to do to obtain better inference on selection (if any) – present in PIPs randomly drawn from BAliBASE – is to adopt the approach of KOSIOL *et al.* (2007), whereby ω would measure the *relative* rather than the absolute strength of selection. In this approach, I would then need to estimate the *average* level of selection strength which is implicit in BAliBASE and use this as the reference, (i.e. the expectation of ω under neutral selection).

Thus, the new ω would be the ratio $\frac{\rho_N/E(\rho_N)}{\rho_S/E(\rho_S)}$, where E is the expectation operator. The expectations are as derived in NEI and GOJOBORI (1986), and ρ_N and ρ_S are computed as in GOLDMAN and YANG (1994). In the latter case, however, q_{ij}, which I denote as $Q_{ij}^{(KHG07)}$, would now be specified as follows

$$Q_{ij}^{(KHG07)} = \begin{cases} 0 & \text{if } i \text{ or } j \text{ is a stop codon} \\ s_{ij}\pi_j\kappa(i,j) & \text{if } i \rightarrow j \text{ is a synonymous change} \\ s_{ij}\pi_j\kappa(i,j)\omega & \text{if } i \rightarrow j \text{ is a nonsynonymous change} \end{cases} \tag{4.3}$$

(KOSIOL *et al.*, 2007), where s_{ij} would be exchangeabilities estimated from multiple alignments stored in BAliBASE, and $\kappa(i,j)$ is now a function of the number of nucleotide changes as defined in (KOSIOL *et al.*, 2007). That is, transition-transversion bias is now modelled by several parameters to allow for double and triple nucleotide changes. This makes $Q_{ij}^{(KHG07)}$ a context dependent instantaneous rate matrix that can model biological mechanisms involving changes in 2 and 3 neighbouring nucleotides. The construction of $Q_{ij}^{(KHG07)}$ using BAliBASE data, together with re-estimation of my sample, could be part of future work using my two-region model.

4.3 Indel Analyses

In Appendix B.3, I summarise the scheme of PASCARELLA and ARGOS (1992). I shall use this summary for my analyses in this section.

4.3.1 Regional Indel Averages

From my results using the codon sample, I obtained $\bar{\ell}^{(R_1)} = 0$, $\bar{\ell}^{(R_2)} = 9.30$, $\bar{r}^{(R_1)} = 0$, and $\bar{r}^{(R_2)} = 0.00327$, where R_1 and R_2 represent portions of alignments that belong to slow and fast substitution rates, respectively. These measurements reflect the fact that indels in my codon alignments under Test 2 in Table 3.3 are concentrated in fast rate regions as I had expected. Furthermore, I applied the scheme strictly, whereby I discarded indels which overlapped the two regions. I also discarded alignments that did not yield at least one indel. To ensure that my regime was very strict, I applied a standard Runs test on each alignment to confirm that the regions' pattern was not random. This to allow for the fact that the trace-back procedure does not guarantee the true alignment, as I had explained in Section 2.5.3.1.

In all, I had remaining in my final sample 72 alignments that were interesting.

Between them they had a total of 25,070 aligned sites. Among aligned sites that were in fast rate regions, I had a total of 82 indels to work with. The two averages, namely, $\bar{\ell}^{(R_2)} = 9.30$ and $\bar{r}^{(R_2)} = 0.00327$ are based on these 82 indels in my final sample. The corresponding two averages of point estimators obtained across the 120 alignments were $\bar{\hat{a}}_{allPIPs} = 0.3819$ and $\bar{\hat{r}}_{allPIPs} = 0.00159$.

The first statistic, namely, $\bar{\hat{a}}_{allPIPs}$ means that the probability of no indels in an alignment is 0.62, while the probability of an indel of length one is 0.24. This is reasonable since several alignments did not have indels, while several others had indels one-gap long. Therefore, $\bar{\hat{a}}_{allPIPs}$ compares reasonably well with $\bar{\ell}^{(R_2)}$ since the lattar is measured only in the fast rate regions of the remaining 72 alignments that passed all the criteria. In a similar vein, $\bar{\hat{r}}_{allPIPs}$ was measured across the entire alignment of each of the 120 PIPs, while $\bar{r}^{(R_2)}$ is now measured across the fast rate regions only. In total, fast rate regions can be considered to be, overall, about half the length of each alignment. On average, therefore, *indels are likely (1) to be about nine times longer, and (2) to have twice the rate, in fast rate regions than when measured across the entire length of the alignment.*

4.3.2 Regional Codon Preference

Using B.10 and B.11, I computed the preference index p and the corresponding standard deviation for each codon in the 72 alignments, and listed these in Table C.1. I then sorted this table by p.

What becomes clear is that among the top 12 of the 61 codons in this table, only half code for hydrophilic amino acids. This is contrary to what was reported by PASCARELLA and ARGOS (1992). In their study using polypeptides, amino acids that flanked indels were mostly hydrophilic, and the authors found that hydrophilic residues are target points for indels. In my pairwise alignments using codon data, however, codons flanked indels randomly between codons that code for hydrophilic amino acids and those that do not.

The disparity could be attributed to the fact that PASCARELLA and ARGOS (1992) used data that consisted of tertiary structures. It appears that the preferential positioning of residues flanking indels in a 3-dimensional folding topology is

different from the preferential positioning of codons flanking indels in primary structures within fast rate regions. This unless bias was introduced when alignments were manually curated using tertiary structure as a guide.

4.3.3 Regional Codon Usage

To investigate codon usage as opposed to codon preference, I measured the frequency of each codon (and of gaps) in slow and fast rate regions of each of the 72 alignments. The results are tabulated in Table C.2 which is sorted by *slow÷fast* in the fifth column.

Codons that are used most in fast rate regions have a smaller slow:fast ratio, and hence are located further up this table. The interesting result here is that the top twelve positions, bar just one, are occupied by codons that code either for Serine or for Arginine. Neither of these two amino acids are in the top twelve positions of Table C.1.

It is clear from this result that *there is a demarcation between codons that are conducive to fast substitutions and codons that have a tendency for flanking positions.* Although fast substitutions and indels are mostly located in the same region, their chemical agents are mutually exclusive.

CHAPTER 5
Detecting Pyrosequencing Errors

5.1 Real-Time Sequencing

To investigate the possibility of a single base mutation in the HIV-1 *pol* gene, NYRÉN *et al.* (1993) performed a novel piece of DNA sequencing that required neither electrophoresis nor any radioactive materials. This pioneering procedure, called the ELIDA, consisted of a series of steps. Each step required only an enzymatic reaction, as illustrated in Figure 5.1, to complete a one nucleotide assaying process.

5.1.1 The ELIDA Concept

Step	Enzymatic Reaction
1	$(DNA)_n + dNTP \xrightarrow{DNA\ \text{Polymerase}} (DNA)_{n+1} + PPi$
2	$PPi + APS \xrightarrow{ATP\ \text{Polymerase}} ATP + SO_4^{2-}$
3	$ATP + \text{luciferin} + O_2 \xrightarrow{\text{Luciferase}} AMP + PPi + \text{oxyluciferin} + CO_2 + hv$

Figure 5.1: The diagram shows Nyrén's method of minisequencing, requiring just three enzymatic reactions and without the need for labels or electrophoresis. The three steps are reproduced from NYRÉN *et al.* (1993). Together they form the ELIDA.

The procedure can be described as follows. A single strand DNA template is first prepared, and then incubated with the necessary enzymes. Each of the four dNTPs, (deoxynucleoside triphosphates which target one of A, C, G, and T during the sequencing elongation), is added sequentially. With each addition, incorporation, if any, takes place by the catalysis of the DNA polymerase enzyme and the current dNTP. If the next base in the template is the complement to the current dNTP, an incorporation "event" is said to occur, resulting in the release of PPi (inorganic pyrophosphate). The quantity released is measured accurately, as this translates to a count on how many homopolymer bases have been incorporated during the current addition.

Accurate measurement is provided by the catalysis of the ATP sulfurylase enzyme and the APS (adenosine-5′-phosphosulphate). This reaction produces the ATP (adenosine-5′-triphosphate) substrate for the next reaction between the luciferase enzyme and luciferin, yielding the byproduct oxyluciferin. Detection of light generated by these reactions is by a luminometer whose peak is recorded by a potentiometer, and translated to a direct count of the homopolymer bases added. Thus, the homopolymer length is resolved – one or more bases – at the current incorporation, if any.

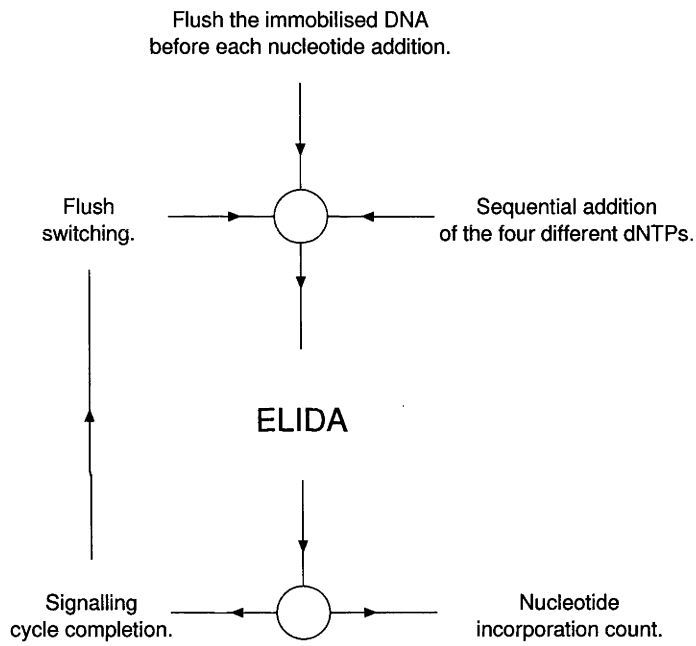


Figure 5.2: The ELIDA was automated to process DNA sequencing in real-time. The illustration is based on RONAGHI *et al.* (1996).

The method of DNA sequencing using an enzymatic luminometric inorganic pyrophosphate detection assay (ELIDA) was first employed by NYRÉN (1987) for reading a single DNA letter of interest. This method was suitable only for experimental purposes where bases had to be detected one at a time. RONAGHI *et al.* (1996) later worked on how the ELIDA could be enhanced, using a cyclical incubation-flush process. This led to the system illustrated in Figure 5.2.

Here, in effect, was the introduction of DNA sequencing in real time. This new technology was developed by Pyrosequencing AB in Uppsala, Sweden. This company was renamed Biotage, and in 2003 licensed the technology to 454 Life Sciences Corporation which is a subsidiary of CuraGen Corporation.

5.2 Massively Parallel Pyrosequencing

In less than twenty years after the introduction of the ELIDA by Nyrén, real-time DNA sequencing was to make the next leap forward following the work of MARGULIES *et al.* (2005).

In summary, the genome is first broken down into random fragments. Each fragment is captured in a separate bead, where it is cloned and amplified within an emulsion, and is turned into a template. Sequencing of templates is then performed by syntheses simultaneously in open wells of a fibre-optic slide. A slide typically contains 1 to 2 million wells, each well housing a template. The slide, in turn, is housed inside a flow chamber, with wells resting in a vertical position.

A second fibre-optic element makes contact with individual wells at the base, and this element channels photons to a sensor. Reagents flow by convection through wells inside which ELIDA like enzymatic reactions occur in parallel. This leads to base extensions – where the length of the homopolymer incorporated is proportional to quanta released by corresponding photons – on templates, with a very large economy of scale (a system now commonly termed *massively parallel pyrosequencing*). Following each extension, residue nucleotides are thoroughly flushed by means of the enzyme apyrase to ensure that prior nucleotides do not remain in wells before the next nucleotide is introduced.

Massively parallel pyrosequencing was developed by 454 Life Sciences and is marketed by Roche Diagnostics. Their introductory machine, the Roche GS 20, could generate reads of approximately 100 bps in length and at a rate of 25×10^6 bps per one four-hour run. Their latest machine which is being marketed presently, namely, the Roche GS FLX, can generate reads of between 200 and 300 bps in length and at a rate of 50×10^6 bps per one four-hour run. This means, for example, that with the FLX, operators can sequence the Human Genome over a ten-day continuous

active time consisting of 24-hour runs at 300×10^6 bps per run. In October 2008, Roche Diagnostics released the Genome Sequencer FLX Titanium Series reagents, which enable 1 million reads at 400 base pairs in length to be produced.

Over the next few years, 3^{rd} generation sequencing systems by Roche, Illumina, Applied BioSystems, and by other contenders who are expected to enter the market as early as 2010, are poised to challenge the scientific community and their funding agents. These systems will be based on the *single-molecule analysis* technology, and are being developed by VisiGen and Helicos (SCHUSTER, 2008).

In the face of these rapid advances, together with cost reductions, it is clear that there will be the need to develop DNA data modelling that can deal with this large data availability in a fast, effective, and practical way without compromising the mathematical structure around which this modelling is built. In the following section, I deal briefly with a specific inherent problem of pyrosequencing that has been acknowledged in many parts of the literature – for example, MARGULIES *et al.* (2005), MEYER *et al.* (2008), and SCHUSTER (2008).

5.3 The Homopolymer Problem

When operating the Roche GS 20, a chain of responses occurs with each incorporation inside wells. A chain starts with the release of inorganic pyrophosphate. This release produces quanta which translate to signal intensities that have to be separated from noise and then normalised. Signal levels following normalisation are equimolar to the number of nucleotide repeats that form a homopolymer, up to a length of eight bases. However, due to the physics inherent in the technology, this linearity property is not guaranteed, and the true length of the homopolymer may not always be accurately resolved, resulting in inadvertent overshooting (inserts) or incomplete extensions (deletes). MARGULIES *et al.* (2005) provide details on this homopolymer effect.

HUSE *et al.* (2007) conducted a study on error rates generated by Roche GS 20 pyrosequencing. 340,150 reads were generated using a PCR amplicon library prepared from 43 reference templates. Each of these templates contained a distinct ribosomal RNA gene – which included the V6 hyper-variable region – from a

collection of 43 divergent bacteria. The inclusion of the V6 region was important because it contains homopolymers which are neither long nor frequent. Overall, the percentages of homopolymers in these reference sequences were composed of 45% and 55% of A/T and C/G, respectively.

The authors constructed a separate multiple alignment for each read against the 43 reference sequences. This enabled them to identify the reference sequence that had the best mapping with the corresponding read, thus forming a sequence pair, namely, the test sequence and its reference. To compute error rates for each pair, they used the Needleman-Wunsch algorithm with optimised settings of gap opening penalty of 5.75 and of gap extension penalty of 2.75.

From a total of 32,801,420 bases in their data set of 340,150 pairs, 159,981 bases were miscalled – a total error rate of 4.877×10^{-3}. A portion of this error rate was attributed to the homopolymer effect, with overshooting being the most prominent, having a net error rate of 1.756×10^{-3} due to inserts of one or more bases. A high percentage of 86% of the reads contained no errors, and those reads which between them constituted 50% of all errors all expressed a percentage identity of less than 95%.

5.3.1 The Homopolymer Effect – Experimental Setting

Here I describe how I have tested for the presence (or absence) of the homopolymer effect that results in an overshoot of *exactly one* base, namely, *monoinserts*. HUSE *et al.* (2007) reported monoinserts due to the homopolymer effect to be the most common among homopolymer inserts in sequences that had been produced from the same sequencer run.

5.3.1.1 The Experimental Data Set

I randomly sampled pyrosequenced reads from the data set of HUSE *et al.* (2007) which consists of 340,150 reads. I aligned each of these reads on the fly with each of the 43 cognate reference sequences, using ClustalW and its standard in-build DNA evolutionary model. I retained the first 100 of these pairwise alignments that had a percentage identity of between 0.75 and 0.96, and discarded the rest. I converted these alignments to corresponding 100 non-gapped sequence pairs to

produce my experimental data set. In each pair, I made the first sequence to be the pyrosequenced read and the second sequence to be the cognate DNA reference.

5.3.1.2 The Three-Region Model

To re-align the 100 pairs in my experimental data set, I constructed a three region model which consists of three PHMMs conjoined by one silent state, as shown in Figure 5.3. This construction is similar to that shown in Figure 2.5, except that now I have added a third PHMM. To each I assigned the parameter vector $(\alpha_\eta, \beta_\eta, \delta_\eta, \epsilon_\eta, \gamma_\eta)$, $\eta \in \{1, 2, 3\}$, and constructed the initial 10×10 transition matrix (not shown) similar to that shown in Figure 3.2.

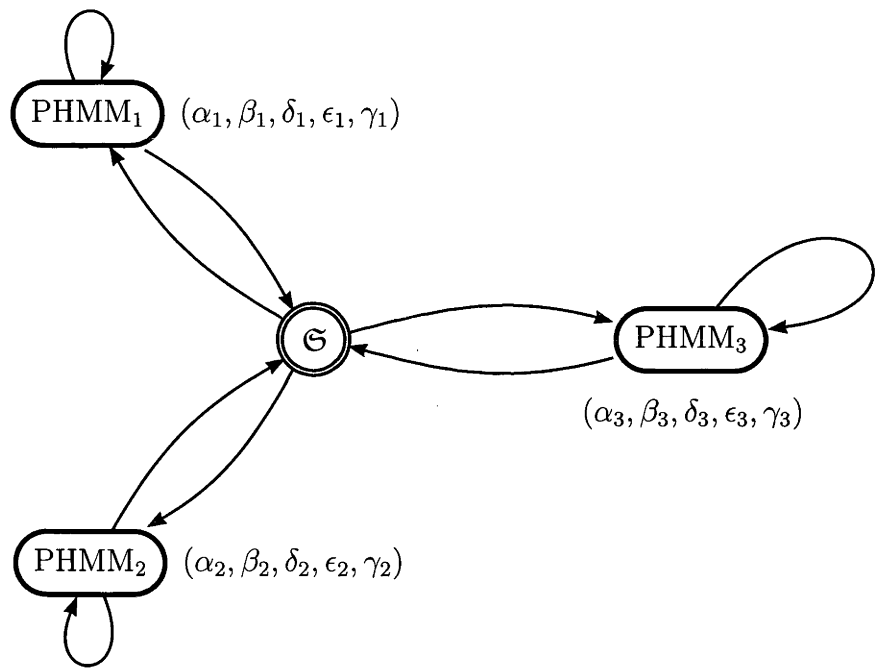


Figure 5.3: The three-region model has three PHMMs, each having the parameter vector $(\alpha_\eta, \beta_\eta, \delta_\eta, \epsilon_\eta, \gamma_\eta), \eta \in \{1, 2, 3\}$, and they are conjoined by one silent state $\mathfrak{S}$ in a similar way as in Figure 2.5. This conceptual topology forms the lower layer of the three-region model.

After applying the Knudsen-Miyamoto (KM) equations to each of the three PHMMs, factoring out the transition probabilities associated with the silent state, and adding a second HMM layer that has three emitting states, the topology is

enhanced to a two-tiered HMM-PHMM model as shown in Figure 5.4. Each of the three emitting states of the upper HMM layer emits one of the three PHMMs in the lower layer with probability $1 - \rho_\eta$, $\eta \in \{1, 2, 3\}$. Each PHMM is parametrised with the KM parameter vector (t_η, a_η, r_η), $\eta \in \{1, 2, 3\}$ where, as in Section 2.5, t_η, a_η, and r_η capture the substitution rate, the indel length, and the indel rate, respectively, in region η.

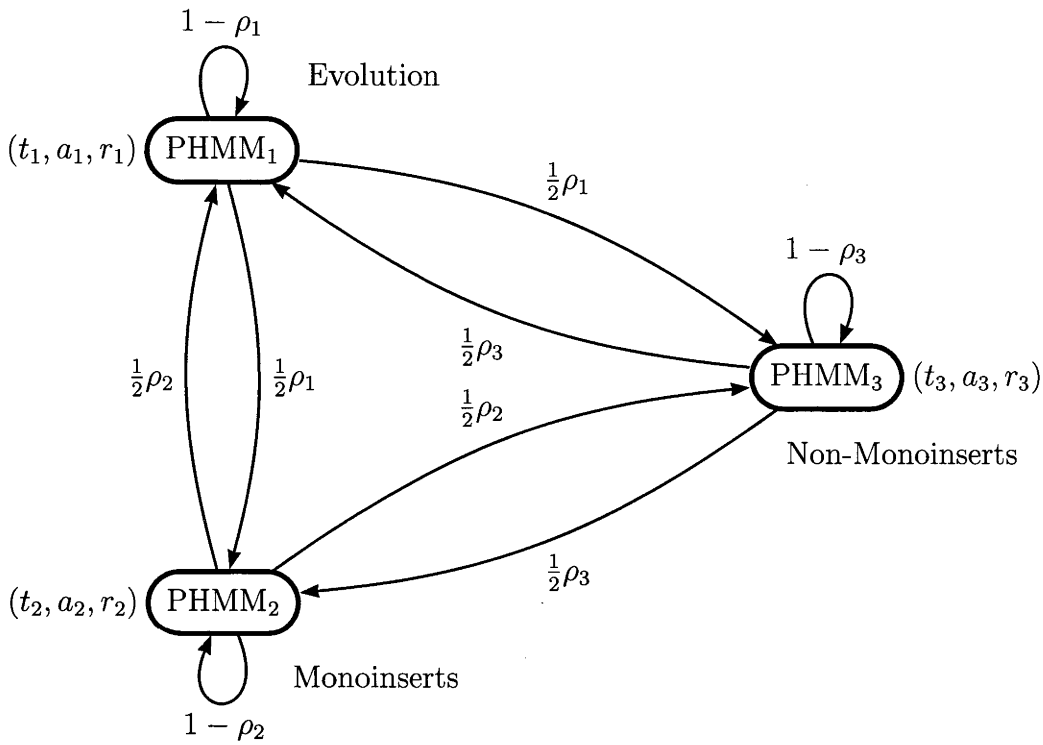


Figure 5.4: The Knudsen-Miyamoto (KM) equations listed in Section 2.5 are applied in order to derive the parameter vector $(t_\eta, a_\eta, r_\eta, \rho_\eta), \eta \in \{1, 2, 3\}$, for each $PHMM_\eta$. $PHMM_1$ is designed to model indels which are due to evolutionary processes. $PHMM_2$ is designed to model *monoinserts* which are due to the homopolymer effect. $PHMM_3$ is designed to model all other homopolymer effects. For region one, the probability of modelling in region one is $1 - \rho_1$, and the probability of leaving region one in order to model in either of the other two regions is $\frac{1}{2}\rho_1$; and similarly for regions two and three. The begin $\mathfrak{B}$ and end $\mathfrak{E}$ states are not shown.

With a two-tiered three-region topology, my aim was to model (1) evolutionary processes with the PHMM in region one, (2) monoinserts with the PHMM in region two, and (3) everything else with the PHMM in region three. The idea here is that during the pairwise alignment of a pyrosequenced read and its cognate DNA, the model would remain mostly in region one. However, it would not be unreasonable

to expect that upon encountering a machine error, the model would switch to either region two if it encounters a machine error consisting of a monoinsert or region three if the machine error is otherwise.

5.3.1.3 Second Order Markov Chain

In designing the emission matrices for the three-region model, my aim was to spatially target monoinsert patterns – in the pairwise alignments – made of adenine, cytosine, guanine, or thymine, as illustrated in the following table:

Base	Pattern 1	Pattern 2
Adenine	A A A -	A A - A
Cytosine	C C C -	C C - C
Guanine	G G G -	G G - G
Thymine	T T T -	T T - T

,

where the single gap due to the homopolymer effect, if present, is always located in the second sequence as a result of the corresponding monoinsert in the first sequence. To achieve "monoinsert targeting", I employed a second order Markov chain for the construction of the emission matrices $E_{M_{w\times z}}$ in 2.20, $E_{X_{w\times 1}}$ in 2.21, and $E_{Y_{1\times z}}$ in 2.22. For this purpose I needed an alphabet with 16 symbols which are

$$AA,\ AC,\ AG,\ AT,\ CA,\ CC,\ CG,\ CT,\ GA,\ GC,\ GG,\ GT,\ TA,\ TC,\ TG,\ TT.$$

To construct the evolutionary rate matrix $P(t)$ for this alphabet, I also needed a 16×16 instantaneous rate matrix Q. Hence I put a 16×16 Jukes-Cantor substitution model R and the vector of uniformly distributed background probabilities $\boldsymbol{q}$ into equation 2.5, giving (to three decimal places)

$$Q = \begin{array}{c} \\ AA \\ AC \\ AG \\ AT \\ CA \\ CC \\ CG \\ CT \\ GA \\ GC \\ GG \\ GT \\ TA \\ TC \\ TG \\ TT \end{array} \begin{array}{c} \begin{array}{cccccccccccccccc} AA & AC & AG & AT & CA & CC & CG & CT & GA & GC & GG & GT & TA & TC & TG & TT \end{array} \\ \begin{bmatrix} -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 & 0.067 \\ 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & 0.067 & -1 \end{bmatrix} \end{array},$$

before computing the exponentiation $P(t) = e^{Qt}$ and substituting into equations 2.20, 2.21, and 2.22.

This choice of R and $\boldsymbol{q}$ were suitable for the purpose of this experiment where my aim was not to tease out features of evolutionary processes as in Chapters 3 and 4 but to "target" nucleotide patterns in the DNA pairwise alignment caused by machine and not by evolution. Obviously, evolutionary processes were not interesting in this setting, and hence it was justifiable to provide a level playing field to the four nucleotides. My interest here was in how well my model specification could differentiate between naturally occurring indels in the pairwise alignment and monoinserts in the first sequence. This experiment was, essentially, about pattern recognition using an HMM technique whereby everything was to be averaged except for the pattern of interest.

5.3.1.4 Emission Probabilities of Monoinserts

The method I use to code the DNA sequences emission matrices W and Z in equations 2.20, 2.21, and 2.22 is illustrated in Figure 5.5. Recall that matrix W codes the first sequence of the pairwise alignment while matrix Z codes the second sequence. For illustration purposes, I shall use a short fictitious sequence, namely, *AGAACGTTAC*, to represent the first sequence of a typical DNA sequence pair in my data set. Hence, the matrices shown in Figure 5.5 are all designated W, and I have three of these matrices; one for each PHMM in my three-region model. (The illustration also applies to the second sequence except that the resulting matrices would be designated Z.)

Each of the 10 letters in this sequence serves as a column heading in each of

the three sequence emission matrices shown in Figure 5.5. Similarly, each of the 16 emission symbols in the alphabet of the second order Markov chain serves as a row heading in each of these three sequence emission matrices.

$W_{Region1}$ =

	A	G	A	A	C	G	T	T	A	C
AA	0.0625	0	0	1	0	0	0	0	0	0
AC	0.0625	0	0	0	1	0	0	0	0	1
AG	0.0625	1	0	0	0	0	0	0	0	0
AT	0.0625	0	0	0	0	0	0	0	0	0
CA	0.0625	0	0	0	0	0	0	0	0	0
CC	0.0625	0	0	0	0	0	0	0	0	0
CG	0.0625	0	0	0	0	1	0	0	0	0
CT	0.0625	0	0	0	0	0	0	0	0	0
GA	0.0625	0	1	0	0	0	0	0	0	0
GC	0.0625	0	0	0	0	0	0	0	0	0
GG	0.0625	0	0	0	0	0	0	0	0	0
GT	0.0625	0	0	0	0	0	1	0	0	0
TA	0.0625	0	0	0	0	0	0	0	1	0
TC	0.0625	0	0	0	0	0	0	0	0	0
TG	0.0625	0	0	0	0	0	0	0	0	0
TT	0.0625	0	0	0	0	0	0	1	0	0

$W_{Region2}$ =

	A	G	A	A	C	G	T	T	A	C
AA	0.0625	0.0625	0.0625	1	0.0625	0.0625	0.0625	0	0.0625	0.0625
AC	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
AG	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
AT	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
CA	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
CC	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
CG	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
CT	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
GA	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
GC	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
GG	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
GT	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
TA	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
TC	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
TG	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	0	0.0625	0.0625
TT	0.0625	0.0625	0.0625	0	0.0625	0.0625	0.0625	1	0.0625	0.0625

$W_{Region3}$ =

	A	G	A	A	C	G	T	T	A	C
AA	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
AC	0.0625	0	0	0.0625	1	0	0	0.0625	0	1
AG	0.0625	1	0	0.0625	0	0	0	0.0625	0	0
AT	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
CA	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
CC	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
CG	0.0625	0	0	0.0625	0	1	0	0.0625	0	0
CT	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
GA	0.0625	0	1	0.0625	0	0	0	0.0625	0	0
GC	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
GG	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
GT	0.0625	0	0	0.0625	0	0	1	0.0625	0	0
TA	0.0625	0	0	0.0625	0	0	0	0.0625	1	0
TC	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
TG	0.0625	0	0	0.0625	0	0	0	0.0625	0	0
TT	0.0625	0	0	0.0625	0	0	0	0.0625	0	0

Figure 5.5: A second order Markov process is used to code the three sequence emission matrices shown here for the fictitious sequence *AGAACGTTAC*. The first matrix is designed to spatially capture evolutionary processes, the second matrix is designed to spatially capture errors due to homopolymer effects with exactly one insertion, and the third matrix is designed to spatially capture all other machine errors. Each design is coded by assigning probabilities at two letters at a time in each row, as explained in the text.

In each of the three matrices, the first column probabilities are always uni-

formly distributed because this column by itself cannot differentiate two contiguous nucleotides from other nucleotides, as the succeeding columns do. The first matrix is assigned to region one, and is designed to spatially model processes due solely to evolution. Hence, every column – following the first column – in this matrix is assigned a 1 at the position corresponding to the current and the preceding nucleotide, while all other positions are assigned a 0. For example, in the second column whose heading is G, and whose preceding heading is A, is assigned a probability 1 at the row with heading AG, while all other positions in this second column are assigned probability 0.

The second matrix is assigned to region two, and is designed to spatially model monoinserts that are due solely to the homopolymer effect that results in exactly one insertion. This is achieved by differentiating contiguous like nucleotides, two at a time, from all other nucleotides. Hence, if the current heading is different from the preceding heading, probabilities in the current column are uniformly distributed. If, on the other hand, the two were the same, a probability 1 is then assigned at the position whose row heading is the same as these two contiguous like nucleotides. For example, heading of the second column is different from the preceding column heading, and hence this column has uniformly distributed probabilities. So does the third column. The fourth column, however, has A as the heading which is the same as the preceding heading. Hence, probability 1 is assigned to this column at the position whose row heading is AA, and 0 in all other positions.

The third matrix is assigned to region three, and is designed to spatially model all other machine errors which are not modelled by the second matrix. In fact, this matrix is the "contrast" of the second matrix, that is, it operates in exactly the opposite way of the second matrix. Thus, for example, because the heading of the second column of the third matrix is different from the preceding column heading, this column is now assigned probability 1 at the position where the row heading is AG, and probability 0 in all other positions. Similarly, the third column has probability 1 assigned at the position where the row heading is GA since the heading of this column is A and the heading of the preceding column is G. However, the probabilities of the fourth column are now uniformly distributed since the heading

of this column and the heading of the preceding column are the same. In this way, each two contiguous sites of the alignment, whose alignment is neither due to evolutionary processes nor due to the homopolymer effect with exactly one insertion, will be spatially modelled by this matrix.

Owing to their specific formulation, these three matrices produce three different sets of emission probabilities, that is, one set for each region in a three-region model. Each region also has its own 3×3 transition matrix as in the two-region model. That is, states M, X, and Y in each of the three regions still follow a first order Markov process. However, to switch between three regions, now I needed a 3×3 region switching matrix as shown in Figure 5.7. This matrix has three switching parameters, namely, ρ_1, ρ_2, and ρ_3. Each of these allows the model to exit the current region and to enter one of the other two regions with equal probability, as shown in Figure 5.7. Under this regime, I would expect that single gaps that are due to the homopolymer effect with exactly one insertion will be best predicted, on average, by ρ_2. This is because the set of emission probabilities of region two will spatially "target" those nucleotides that yield these gaps when these are present since each of these gaps in the second sequence corresponds to two like contiguous nucleotides in the first sequence. The converse applies to ρ_3, while ρ_1 would keep the model in region one at those sites which are aligned according to evolutionary processes and not due to machine errors.

5.3.2 The Homopolymer Effect – Hypothesis Testing

To carry out a test for each pairwise alignment, I defined H_o and H_a as shown below. That is, under the null, there are no sequencing errors in the pairwise alignment that are due to the homopolymer effect with exactly one insertion. Hence the model is equivalent to a two-region model under the null, namely, the region of evolutionary processes and the region of sequencing errors. I needed to test this hypothesis against the alternative hypothesis, namely, the null is untrue. That is, the pairwise alignment has gaps in the second sequence that are due to the homopolymer effect with exactly one insertion. Under the alternative, therefore, the model has three regions.

The hypotheses are

$$H_o : x_1 \neq x_2 = x_3,\ \rho_1 \neq \rho_2 = \rho_3, \qquad \text{versus}$$
$$H_a : t_1 \neq t_2 = t_3,\ a_1 \neq a_2 \neq a_3,\ r_1 \neq r_2 \neq r_3,\ \rho_1 \neq \rho_2 \neq \rho_3.$$

where the subscripts refer to the region number. x_η means all the parameters in region η, $\eta \in \{1, 2, 3\}$, while t, a, and r are the substitution rate, the indel length and the indel rate parameters, respectively, as in Chapter 3.

Under the null, region one models evolutionary processes with its own set of parameters. At the same time, parameters in region two and corresponding parameters in region three are forced to be equal. This means that regions two and three are equivalent to one region with its own set of parameters and models all types of errors with the same expectation since it assumes that there are no significant monoinserts.

Under the alternative, region one retains the same set of parameters, and again models evolutionary processes independently from regions two and three. However, all parameters, except the substitution rate parameters, are now relaxed under the alternative in regions two and three. These two regions, between them, model the same substitution rate (which is assumed to be the same among all types of errors due to machine). Thus, the only differentiating factor between regions two and three are the monoinserts – captured in region two, but not in region three – and all other pyrosequencing errors that are not interesting and which are captured in region three but not in region two.

With three regions and several parameters, and considering that reads are only about 100 nucleotides long, I expected this test to have low power, and hence I set the level of significance at 10% *a priori*. Three regions were necessary for this experiment because there are three distinct types of indels which the optimiser was required to differentiate from each other, namely, indel processes due to evolution (region 1), indel processes due to monoinserts (region 2), and indel processes due to sequencing errors caused by all other machine artifacts (region 3).

5.3.3 The Homopolymer Effect – Results

Before studying the results obtained from the homopolymer experiment, I briefly revise the meaning of the indel length parameter a and of the region switch parameter ρ.

Figure 5.6 illustrates how a very small value of a, say 0.01, would mean that the corresponding region would allow, in all probability, indels that are at most one gap in length. At the other end of the scale, a large value of a, say 0.75, would mean that indels in this region are unlikely to be of the same length, and that their exact length would be harder for the trace-back procedure to resolve accurately. In each case, however, whether an indel would occur would still be determined solely by the indel rate parameter r of that region.

Figure 5.7 shows the three switching parameters of the three-region model that I used for this experiment. These parameters determine the transition probabilities of the three-state HMM that assigns regions to sites. One point to note here is that a large value of ρ belonging to a region would mean that the three-region model is spending very little time in that region. Another important point is that ρ values are independent from each other in the sense that they do not necessarily add to one. For example, one ρ value does not increase directly at the expense of any of the other two.

Each panel in Appendix D shows three pairwise alignments. The first is constructed by the trace-back procedure under the null and the second under the alternative. The third alignment is produced by ClustalW. The first sequence in these alignments is always the pyrosequenced read while the second is the cognate reference sequence. This means that only homopolymer patterns with a single gap in the second sequence are of interest in this experiment since I am restricting my investigation to homopolymer inserts (in the first sequence) of exactly one base.

The first thing to notice in these panels is that ClustalW alignments are always the longest, while alignments under H_a are always the shortest. This shows that the three-region model is very economical in gap insertions, and considering that these sequence pairs have a high percentage identity, this behaviour was expected. The reason for the longer alignments of ClustalW could be attributed to the fact that I

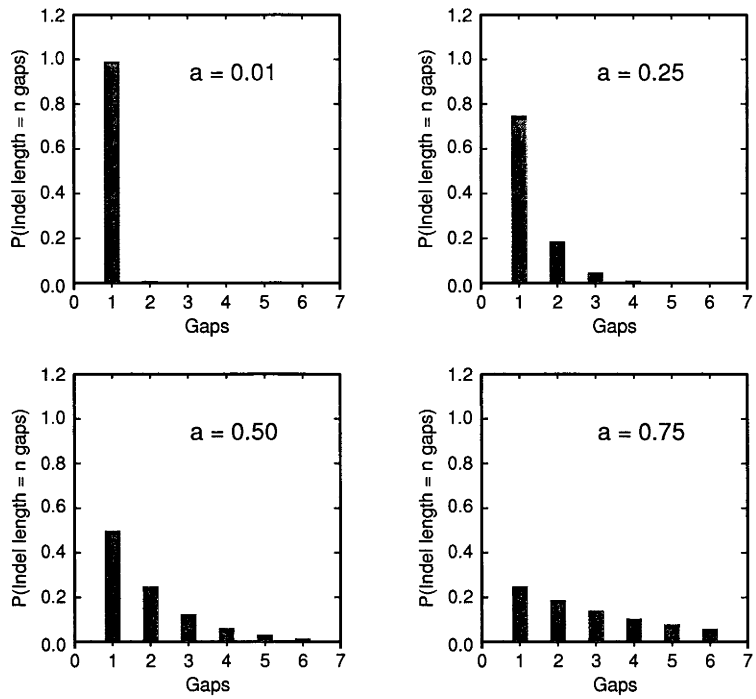


Figure 5.6: The plots show different responses to corresponding values of the parameter $\{a : 0 < a < 1\}$ using P(Indel Length $= n$ gaps) $= (1-a)a^{n-1}$. In the first panel, it is shown that a very low level of the estimator $\hat{a}$ would suggest that *given* an indel in the pairwise alignment, there is a very high probability that the length of this indel would be of just one gap. On the other hand, a very high level (the fourth panel) would suggest that it is hard for the optimiser to resolve the true length. Given that pairwise alignments in this experiment consist of close homologues, a very low level would mean that there is a high probability that no gaps are present.

arbitrarily used a gap opening penalty of 1 and a gap extension penalty of 3 together with a Jukes-Cantor substitution model. In my experiment, these settings were not critical since I needed ClustalW alignments solely for the purpose of computing percentage identities before randomly selecting pairs to construct my sample. What is important here is that the high percentage identities are compatible with the fact that alignments under H_a never expressed insertions that are longer than one gap, as was expected.

Table 5.1 gives a summary of the nine panels. Columns 2, 3, and 4 have three numbers in each row for regions 2 and 3. The first is the level under the null, which is the same in regions 2 and 3 under the assumption of no homopolymer effect. The second and third are levels under the alternative in regions 2 and 3, respectively.

All levels in column 2, with only one exception, are very low as expected. This

$$
\begin{array}{c} \\ \mathrm{R}_1 \\ \mathrm{R}_2 \\ \mathrm{R}_3 \end{array}
\begin{array}{c}
\begin{array}{ccc} \mathrm{R}_1 & \mathrm{R}_2 & \mathrm{R}_3 \end{array} \\
\begin{bmatrix}
1-\rho_1 & \frac{1}{2}\rho_1 & \frac{1}{2}\rho_1 \\
\frac{1}{2}\rho_2 & 1-\rho_2 & \frac{1}{2}\rho_2 \\
\frac{1}{2}\rho_3 & \frac{1}{2}\rho_3 & 1-\rho_3
\end{bmatrix}
\end{array}
$$

Figure 5.7: A three-region model requires a 3-state HMM with three switching parameters, namely, ρ_1, ρ_2, and ρ_3. The smaller the value of the switching parameter in a region, the higher the probability the three-region model will remain in that region. When the model exits that region, it will then enter either of the other two regions with equal probability.

is because, as mentioned earlier, we do not expect insertions to have more than one gap. Levels in column 3 are also very low as expected since sequence pairs are made from close homologues. All third levels in column 4 are high, indicating that the model did not spend too much time resolving non-homopolymer effects, while it spent most of its time in resolving homopolymer effects in region two. Note that second levels in this column are always less than or equal to first levels under the null. This indicates the small homopolymer effect detected under the alternative with three regions. Finally, number of insertions (and deletions) – shown in the last column – due to evolutionary processes are just two in each alignment, which is plausible and consistent with the fact that I kept percentage identity within a narrow range.

In all, I counted just ten insertions that were due to the homopolymer effect, namely, 7 cytosines, 2 adenines, 1 guanine, and none thymine. With a sample of 89 pairs (after discarding 11 pairs which did not yield a positive LR), and a conservative average of, say, 108 bases per read, this gives me an error rate of 1.040×10^{-3} within the class of reads that have a percentage identity between 0.75 and 0.96. This is about half as much as that stated by HUSE *et al.* (2007), which was 1.756×10^{-3} for errors attributed to homopolymer insertions across all reads. However, HUSE *et al.* (2007) based their computation on insertions of all possible lengths and not just one-residue overshoots as in my experiment, and this would be the reason for the disparity. That is, the error rate of overshoots that are longer than one would be about 0.7×10^{-3}.

Panel	a	$r.t^{\dagger}$	ρ	Homopolymers (monoinserts)[§]	Indels[‡]
1	0.00^{a} 0.00^{b} 0.24^{c}	0.04^{a} 0.04^{b} 0.00^{c}	0.19^{a} 0.17^{b} 0.99^{c}	1 (C)	2
2	0.00 0.00 0.24	0.01 0.01 0.01	0.19 0.16 0.97	0	2
3	0.00 0.00 0.01	0.02 0.03 0.02	0.41 0.18 0.51	0	2
4	0.00 0.00 0.04	0.04 0.04 0.04	0.20 0.18 0.99	2 (C, C)	2
5	0.00 0.00 0.12	0.04 0.02 0.22	0.20 0.20 0.99	2 (A, C)	2
6	0.40 0.26 0.41	0.02 0.02 0.00	0.16 0.15 0.99	2 (C, G)	2
7	0.00 0.00 0.78	0.04 0.04 0.00	0.17 0.17 0.99	1 (C)	2
8	0.00 0.00 0.06	0.04 0.03 0.21	0.20 0.20 0.99	2 (A, C)	2
9	0.00 0.00 0.14	0.01 0.01 0.01	0.19 0.16 0.97	0	2

[a] Estimators level in regions two and three, under the null: no homopolymer effect.

[b] Estimators level in region two under the alternative: homopolymer effect.

[c] Estimators level in region three under the alternative: homopolymer effect.

[†] Both t and r are expected to be small under both the null and the alternative, and hence the product of these two rates is more informative in this setting.

[§] Monoinserts i.e. Homopolymers that result in an overshoot of exactly one insertion.

[‡] Indels solely due to evolutionary processes.

Table 5.1: Summary of the nine panels in Appendix D.

5.3.4 Conclusions

Although HUSE *et al.* (2007) used the V6 hyper-variable region to construct the reference sequences, extensions in the reads were not uniform across the four bases A, C, G, and T. They found that the frequency of A/T extensions was 24% higher than expected, and that of C/G extensions was concomitantly less, in the

reads. Yet, my results show that homopolymer errors consisting of extensions with one-residue overshoot in the reads are caused mostly by cytosine nucleotides and very rarely, if ever, by thymine. Considering that A/T extensions were by far more prevalent in the HUSE *et al.* (2007) data, I had expected a bias in favour of A and T in my results on errors due to homopolymer effects with exactly one insertion, but this is not the case. With these data, and on the basis of my results, machine accuracy has clearly not been uniform across A, C, G, and T, but heavily biased in favour of T. This could be attributed to either a machine artifact or to the fact that the V6 region favoured C/G extensions against A/T extensions in the reference sequences by a ratio of 55:45, as reported by HUSE *et al.* (2007).

The authors also reported that the GS 20 provides a quality score for every position in a read. The score is a measure of confidence that the homopolymer length at that position is accurately resolved. At one end, a high score indicates that no homopolymer is present, and the position is therefore easier to resolve. At the other end, a low score indicates that a long homopolymer is present, and the position is difficult to resolve.

However, they also found three effective criteria for reducing the error rate. That is, they found that by removing reads (1) which contain at least one N, (2) whose lengths are aberrantly short or long, and (3) which do not match perfectly to the primer, the error rate decreases from 0.49% to 0.16%. This reduction would be practical since only about 10% of total reads would have to be culled.

An implication here is that filtering based on these criteria would raise the level of quality scores. Quality scores could then be used in a correlation test after monoinserts – due to homopolymer effects – had been identified as described here using three-region HMM-PHMM modelling. Since quality scoring and predicting monoinserts are two independent methods, a high correlation would confirm whether quality scores are compatible with the predictions obtained from the HMM-PHMM model.

CHAPTER 6
Discussion

I have addressed the issue of heterogeneity in evolutionary rates along the DNA taken from a broad range of randomly sampled species. Two of these rates that are central to understanding evolution are (1) the rates of substitution in the case of DNA/codon biological encodings (BEs) and of replacement in the case of protein BE, and (2) the rates of indels.

Biologists had for long been aware of heterogeneity, and several methods had been used to uncover its causes. My approach was to take into account the role of secondary structure in these evolutionary rates within the species. When working with protein and codon BEs, my idea was to devise a parameter that can sense hydrophilic amino acids, or their cognate codons, and use this parameter to augment their corresponding background probabilities. This was a novel idea based on the method used successfully in the multiple sequence aligner ClustalW, whereby the opening gap penalty and the gap extension penalty are reduced whenever patches of hydrophilic amino acids are encountered. For my purpose, however, this parameter was not useful on its own. I needed to study its behaviour in conjunction with the classical parameters, namely, the parameter that models rates of replacement (or substitution) and the indel parameter set – composed of the length and rate parameters – which models indel behaviour.

The classical parameters had been employed successfully in a PHMM setting by KNUDSEN and MIYAMOTO (2003) (KM), and therefore I only required to incorporate the hydrophilicity parameter in this device. In addition, however, I needed to allow all parameters to vary freely and independently in the different regions implicit in the data in accordance with secondary structure components. I needed, therefore, yet another novel idea that would allow me to combine the classical HMM with a pair of KM-PHMMs in order to model the two broad types of heterogeneity. To achieve this, I employed a stationary Markov chain of hidden states, with one hidden state for each region (or more precisely one for each KM-PHMM). This led

me to a two-tiered HMM-PHMM topology suitable for pairwise alignments with secondary structure regional context.

HMM-PHMM topologies had been used successfully by various workers in the field of gene finding. MEYER and DURBIN (2002) for example, used an HMM-PHMM topology to exploit the similarities between a pair of DNA sequences, together with splicing and coding information, to simultaneously predict gene structure and a pairwise alignment. HOBOLTH and JENSEN (2005) extended this concept with three homologous DNA sequences, taken from prokaryotic organisms, thus enlarging the three state PHMM to a set of 15 states. For what I had set out to achieve, namely, a better pairwise alignment, a simple HMM-PHMM configuration with a single silent state at the centre sufficed and proved to be very effective. This is because my aim was not to *predict* structures but rather to *exploit* the biological fact that secondary structure is a very important determinant of evolutionary rates.

Two important components of secondary structure in coding DNA are the hydrophilic and the conserved regions. It had always been reasonable to assume that slower rates of evolution would occur at the core, where the DNA codes for important functions and structures, while faster rates would occur on the surface. It had never been shown quantitatively, however, that the solvent regions and the much faster rates of both substitution and indel rates coexist at least spatially in all likelihood. This also implied that the two rates are also mostly co-located in the solvent regions. In non-coding DNA, however, the two fast rates were no longer, or at most weakly, co-located. In this case, the distinction between slow and fast substitution rates was sharp, reflecting upon the fact that the evolutionarily conserved secondary structure in rRNA molecules are well defined (WUYTS *et al.*, 2004). Here, however, conservation was not a strong determinant on the placement of indels, thus suggesting that the co-location of the two fast rates – substitutions and indels – is a property solely of the solvent regions.

Several serendipitous topics for investigation emerged following my successful application of the HMM-PHMM topology in this work. First was the distribution – across PIPs, and hence across pairs of unique species – of slow substitution rates in one region and the distribution of fast substitution rates in the other region. I

was not surprised to find that the two distributions are largely independent of each other. This considering that secondary structure components are highly distinct. I had also expected that the two distributions would be radically different. It was natural that I tested first for the distribution of the slow rates. They turned out to have a normal distribution, as is often the case with random variables that are much better understood. On the other hand, however, I had initially thought that the fast rates would be largely erratic, and that their distribution would be merely noise and not convey any information. Not until I realised that the occasionally and exceedingly high substitution rates were mere artifacts of the optimiser did I start to notice the log-normality of the fast rates. There is reason here to believe that although the two rates – slow and fast – appear to be remote from each other, yet they turn out to be closely related. My conjecture at this point would be that both rates are contributing to survival, but in a different and in what appears to be a complementary way. Second is the ratio between the synonymous and the non-synonymous rates of substitutions. The parameter modelling this ratio has a distribution which is poorly understood. Here I have proposed a way how to deal with this parameter separately from all the other parameters in the model, using a two-step estimation procedure that had not been tried before in the literature. In attempting to detect positive selection in my data set, it was unfortunate that I was let down by the GY94 model. This model, being purely mechanistic and modelling only single base substitutions in each codon, and not being neighbour context aware, could not deal effectively with the fast rate regions. Nevertheless, I have shown that this parameter plays a significantly different role in the two regions, and with a richer substitution model, the two-step estimation can prove to be very useful. Third, it is clear that codon usage is different in the two regions. At the same time, the chemical agents that determine which codons are predominantly located in the fast regions are mutually exclusive from the chemical agents that determine which codons are most accommodating to indels in this region. There is a strong indication that a systematic interplay among chemical agents that control codon behaviour exist in the fast rate regions and is yet to be understood.

My HMM-PHMM topology can go beyond the comparative prediction of

purely evolutionary processes. I found that the topology is versatile and can also be used for detecting pyrosequencing errors efficiently through sampling. The experimental setting for this purpose turned out to be more elaborate than I had expected. First it required me to increase the number of regions from two to three. This meant that now I needed a Markov chain with three states to switch in between three KM-PHMMs. Second, to differentiate between evolutionary indels and machine induced indels, I was also required to raise the order of the emission probabilities Markov chains to two. The increase in the number of parameters in the model, together with the fact that the difference between the two types of indels is very subtle, meant that the model will have low power in this setting. Nevertheless, a good estimate of the rate of sparse errors caused by the homopolymer effect inherent in the technology could still be obtained.

The effectiveness of the topology could perhaps be increased by increasing the order of the Markov chain within the state transition matrices. This approach presented me with the computational difficulty in that transition probabilities would now have much finer gradations, and this tended to cause underflow errors. A more serious problem is computational time. Each pairwise alignment was taking, on average, approximately ten to fifteen minutes to complete on a Cray XD1 Supercomputer. With large samples of, say, 100-200 PIPs, experiments are therefore very costly to carry out to completion. Implementing better coding techniques may help to alleviate these problems. Shorter computational times would allow me to produce replications of my experiments and thus confirm with higher certainty that the results that I have obtained in this work are repeatable.

Another shortcoming in my work, due to long computational times, is the omission (for expeditious reasons) of confidence intervals of my estimators. All my inferences have been based on point estimators without regard to statistical reliability. To make matters worse, when I first started this work in late 2004, there was no theory that could show whether estimators computed with two sequences using a pair HMM were consistent. It had been known that estimators computed with just one sequence using a classical HMM are consistent (ARRIBAS-GIL *et al.*, 2006), but with two sequences I was working with uncertainty. It was very relieving

that after more than two years since I had started, I discovered that consistency of maximum likelihood estimators holds also for two sequences given that their observed length is sufficiently informative, even if the evolutionary distance between them is not known (i.e. ARRIBAS-GIL *et al.*, 2006, p. 657).

Until now, the construction of phylogenetic trees had been based on the alignment of the corresponding biologically encoded sequences. The fact that this alignment is invariably given prior to the construction implies that there is a major flaw in this approach to phylogeny based studies. The two important evolutionary problems, namely, alignment and phylogeny, are profoundly interdependent, and their respective maximisation by the maximum likelihood method need to be formulated as one problem. Furthermore, the probability of the alignment itself had also been assumed to be the sum over different alignments that represent the set of evolutionary events, namely, *mutations*, *insertions*, and *deletions*. With my HMM-PHMM topology I have added a new element to this set, namely, *secondary structure*. On the one hand I am very disappointed that LÖYTYNOJA and GOLDMAN (2008) did not acknowledge my announcing this new element in the pairwise alignment method at the Brisbane International Congress just over two years ago, but on the other hand I am also very pleased that this concept is already proving to be useful and may become the norm over the coming years within the community of researchers working in this field.

APPENDIX A
Chapter One

A.1 Silent Chains

I discuss here a method of how to determine all possible silent chains between states a and b in an HMM that has D silent states. I am not aware of a mathematical expression that can directly identify these chains, and until the time of writing I have not been able to derive such an expression myself. I have chosen, at this stage, to take the following approach.

Consider, for example, an HMM with $D = 4$. The set of all possible silent chains between states a and b can be enumerated from 1 to 16 as follows:

1 (a, b)
2 (a, 1, b)
3 (a, 1, 2, b)
4 (a, 1, 2, 3, b)
5 (a, 1, 2, 3, 4, b)
6 (a, 1, 2, 4, b)
7 (a, 1, 3, b)
8 (a, 1, 3, 4, b)
9 (a, 1, 4, b)
10 (a, 2, b)
11 (a, 2, 3, b)
12 (a, 2, 3, 4, b)
13 (a, 2, 4, b)
14 (a, 3, b)
15 (a, 3, 4, b)
16 (a, 4, b)

It turns out to be relatively simple to construct the following corresponding matrix, which I call the *silent mapping matrix.*

1	2	3	4	5	4	3	4	3	2	3	4	3	2	3	2
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	8	7	6	5	4	3	2	1	4	3	2	1	2	1	1
0	0	4	3	2	1	2	1	1	0	2	1	1	0	1	0
0	0	0	2	1	1	0	1	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

To start with, the number of columns in this matrix is equal to 2^D, and the number of rows is equal to $D + 2$. The first row has what I call *peak integers*. I define a *peak integer* as an integer whose both abutting integers are smaller, and I denote it by $\wp_u$ where $u = 1, 2, \ldots, D$. Hence, for $D = 4$, I have *peak integers* $\wp_1 = 5, \wp_2 = 4, \wp_3 = 4$, and $\wp_4 = 3$, and they correspond to silent chains numbered earlier as $5, 8, 12$, and 15 respectively. The challenge now is to locate each *peak integer* and determine its value.

To do this, I first have to construct rows 2 to $D + 2$. Row 2 is simply filled with integers starting from 2^D all the way down to 1 decrementing by one from left to right. Row 3 is filled with integers starting from 2^{D-1} in the second column all the way down to 1 from left to right, and then start again with 2^{D-2}, and keep repeating until I place 2^0 in the last cell. I repeat this process in row 3, although now I start with 2^{D-2} in the third column, and I keep repeating with 2^{D-3}, until it only remains to place 2^0 in column $D + 1$ of the last row.

Across the entire matrix, I next identify all the ones that are immediately preceded by a zero. Each of these ones point at the column of each *peak integer*. For example, $\wp_1$ is in the same column of the one preceded by a zero in the last row. $\wp_2$ and $\wp_3$ are in the columns of the two ones preceded by a zero in the second last row, and so on. The value of each *peak integer* is equal to the number of integers in its column. For example, $\wp_1$ has five integers in its column, and hence $\wp_1 = 5$.

Once the *peak integers* in the first row have been determined, filling the cells in between with decrementing integers is trivial. Denote each of these integers by c_j, $j = 1, 2, \ldots, 2^D$, where j is the column number. Then, the number of silent states in silent chain j is equal to $c_j - 1$.

What is left to be done is to determine the index of each silent state in each

silent chain. Define the sets $\{2^q, 2^q - 1, \ldots, 1\}$ for $q = D-1, D-2, \ldots, 0$, and number these sets $1, 2, \ldots, D$ respectively. From now on I consider only the last D rows of the *silent mapping matrix.*

All elements in column one are zero, and hence, the first silent chain is empty. That is, flow is directly from state a to state b. In the second column, I only have integer 8 which belongs to set number 1. Hence, the second silent chain has only one silent state with index $w = 1$. In the third column I have two integers, namely, 7 and 4. The first belongs to set number 1 and the second belongs to set number 2. Hence, the third silent chain has two silent states with indexes $w = 1$ and $w = 2$. Continuing in this manner, I find that in column 2^D I only have the integer $2^q, q = 0$, which belongs to set number D. This means that the last silent chain has only one silent state with index $w = D$.

This procedure for constructing the *silent mapping matrix* and deducing silent chains may seem elaborate. I have found, however, that once the mosaic of this matrix reveals itself, it becomes a straightforward task to implement this procedure in computer code in order to construct all the possible silent chains between any two given states for a given HMM with D silent states. The availability of these chains makes it possible to compute Ω_{ab} and then generalise the forward, backward, and Baum-Welch algorithms which I have formulated independently as shown below.

I should at this point mention that similar generalised algorithms may have been implemented by the authors of the HMMER (EDDY, 2003) computer program. I may contact these authors for possible discussion on this issue at some stage.

A.2 General Formulas

A.2.1 Notation for the General Forward, Backward and Baum-Welch Algorithms

(i) $P(y^n|HMM)$: the probability of observing sequence y of length n given the HMM,

(ii) $s_t^{(i)}$: the emitting state with index i, $i = 1, 2, \ldots, r$, of the HMM at position t, $t = 1, 2, \ldots, n$,

(iii) $d_t^{(w)}$: the silent state with index w, $w = 1, 2, \ldots, D$, of the HMM at position t,

(iv) $y_t^{(k)}$: the observable item with index k, $k = 1, 2, \ldots, K$, emitted at position t,

(v) $f_t^{(i)}$ ($b_t^{(i)}$): the forward (backward) score contributed by state $s^{(i)}$ up to position t,

(vi) T_{ab}: the transition probability from state a to state b,

(vii) E_{jk}: the emission probability of the observable item $y^{(k)}$ by the current state $s^{(j)}$,

(viii) $\mathfrak{B}$ ($\mathfrak{E}$): the begin (end) state,

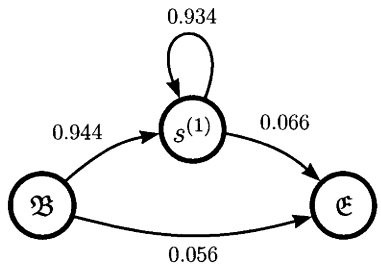
(ix) Ω_{ab}: the total probability of transitions from state a to state b through all the possible silent chains between state a and state b.

$$
\begin{array}{c}
 \\
\mathfrak{B} \\
d^{(1)} \\
d^{(2)} \\
s^{(1)}
\end{array}
\begin{array}{c}
\begin{array}{cccc} d^{(1)} & d^{(2)} & s^{(1)} & \mathfrak{E} \end{array} \\
\left[\begin{array}{cccc}
0.8 & 0.0 & 0.2 & 0.0 \\
0.0 & 0.7 & 0.3 & 0.0 \\
0.0 & 0.0 & 0.9 & 0.1 \\
0.8 & 0.1 & 0.1 & 0.0
\end{array}\right]
\end{array}
\begin{array}{c}
\begin{array}{cc} \mathbb{A} & \mathbb{B} \end{array} \\
\left[\begin{array}{cc}
0.0 & 0.0 \\
0.0 & 0.0 \\
0.0 & 0.0 \\
0.2 & 0.8
\end{array}\right]
\end{array}
$$

Figure A.1: Example 3.10 in ISAEV (2004)

A transition matrix T with silent states $d^{(w)}, w = 1, 2, \ldots, D$, such as the one shown in Figure A.1 with $D = 2$, can be reduced to a transition matrix denoted by T^*, whereby the silent chains are eliminated. It is trivial to apply the method described in Section A.1 for obtaining silent chains, and then sum silent chain probabilities $\forall\, a, b$ to construct the reduction from T to T^*. For example, the matrix T^* and the associated HMM shown in Figure A.2 is obtained after reducing the matrix T in Figure A.1. The purpose of this reduction is to simplify the general Baum-Welch.

In what follows, source emitting and source silent states will be indexed by i and w respectively, and similarly, sink emitting and sink silent states will be indexed by j and z respectively.



$$
\begin{array}{c} \\ \mathfrak{B} \\ s^{(1)} \end{array}
\begin{array}{c}
\begin{array}{cc} s^{(1)} & \mathfrak{E} \end{array} \\
\left[\begin{array}{cc} 0.944 & 0.056 \\ 0.934 & 0.066 \end{array}\right]
\end{array}
\left[\begin{array}{cc} \mathbb{A} & \mathbb{B} \\ 0.0 & 0.0 \\ 0.2 & 0.8 \end{array}\right]
$$

Figure A.2: Reduction of transition matrix T with silent states shown in Figure A.1 to a matrix T^* without silent states.

A.2.2 The General Forward Algorithm

$$f_1^{(j)} = \left(T_{\mathfrak{B}j} + \sum_{w=1}^{D} \Omega_{\mathfrak{B}w} T_{wj} \right) E_{jk},$$

$$f_t^{(z)} = \sum_{i=1}^{r} f_t^{(i)} \Omega_{iz},$$

where

$$t = 1, 2, \ldots, n-1.$$

$$f_t^{(j)} = \left(\sum_{i=1}^{r} f_{t-1}^{(i)} T_{ij} + \sum_{w=1}^{D} f_{t-1}^{(w)} T_{wj} \right) E_{jk},$$

where

$$t = 2, 3, \ldots, n.$$

$$P(y^n | HMM) = \sum_{i=1}^{r} f_n^{(i)} \Omega_{i\mathfrak{E}},$$

where

$$j = 1, 2, \ldots, r,$$

$$z = 1, 2, \ldots, D,$$

$$k \in \{1, 2, \ldots, K\}.$$

A.2.3 The General Backward Algorithm

$$b_n^{(i)} = T_{i\mathfrak{E}} + \sum_{z=1}^{D} T_{iz}\Omega_{z\mathfrak{E}},$$

$$b_t^{(w)} = \sum_{j=1}^{r} \Omega_{wj} E_{jk} b_t^{(j)},$$

where

$$t = n, n-1, \ldots, 2.$$

$$b_t^{(i)} = \sum_{j=1}^{r} T_{ij} E_{jk} b_{t+1}^{(j)} + \sum_{z=1}^{D} T_{iz} b_{t+1}^{(z)},$$

where

$$t = n-1, n-2, \ldots, 1.$$

$$P(y^n|HMM) = \sum_{j=1}^{r} \Omega_{\mathfrak{B}j} b_1^{(j)},$$

where

$$i = 1, 2, \ldots, r,$$

$$w = 1, 2, \ldots, D,$$

$$k \in \{1, 2, \ldots, K\}.$$

A.2.4 The General Baum-Welch Algorithm

For brevity, formulas are for the transition matrix only. One sequence of training data is assumed for simplicity. The normalisation factor is omitted.

Emitting states only cases

$$\overline{T_{\mathfrak{B}j}} = T^*_{\mathfrak{B}j} E_{jk} b_1^{(j)},$$

$$\overline{T_{ij}} = \sum_{t=2}^{n-1} f_t^{(i)} T^*_{ij} E_{jk} b_{t+1}^{(j)},$$

$$\overline{T_{i\mathfrak{E}}} = f_n^{(i)} T^*_{i\mathfrak{E}}.$$

Silent states only cases

$$\overline{T_{wz}} = f_t^{(w)} T_{wz} b_t^{(z)}.$$

Mixed cases

$$\overline{T_{\mathfrak{B}z}} = T_{\mathfrak{B}z} b_1^{(z)},$$

$$\overline{T_{wj}} = f_t^{(w)} T_{wj} b_t^{(j)},$$

$$\overline{T_{iz}} = f_t^{(i)} T_{iz} b_t^{(z)},$$

$$\overline{T_{w\mathfrak{E}}} = f_n^{(w)} T_{w\mathfrak{E}}.$$

In each case

$$t = 1, 2, \ldots, n-1,$$

$$i = 1, 2, \ldots, r,$$

$$w = 1, 2, \ldots, D,$$

$$k \in \{1, 2, \ldots, K\}.$$

APPENDIX B
Chapter Two

B.1 Taylor Series Expansions

I am grateful to David Eberly for his instructive ideas which enabled me to construct the following derivation.

Consider the following Taylor series expansions

$$f(x+a) = f(x) + af'(x) + \frac{a^2}{2!}f''(x) + \frac{a^3}{3!}f'''(x) + \epsilon(4), \tag{B.1}$$

$$f(x-a) = f(x) - af'(x) + \frac{a^2}{2!}f''(x) - \frac{a^3}{3!}f'''(x) + \epsilon(4), \tag{B.2}$$

where $\epsilon(n)$ is the error term of order n. Subtract B.2 from B.1

$$f(x+a) - f(x-a) = 2af'(x) + \frac{2a^3}{3!}f'''(x) + \epsilon(5),$$

$$\frac{f(x+a) - f(x-a)}{a} = 2f'(x) + \frac{2a^2}{3!}f'''(x) + \epsilon(4). \tag{B.3}$$

Consider further the following expansions

$$f(x+2a) = f(x) + 2af'(x) + 2a^2f''(x) + \frac{8a^3}{3!}f'''(x) + \epsilon(4), \tag{B.4}$$

$$f(x-2a) = f(x) - 2af'(x) + 2a^2f''(x) - \frac{8a^3}{3!}f'''(x) + \epsilon(4). \tag{B.5}$$

Subtract B.5 from B.4

$$f(x+2a) - f(x-2a) = 4af'(x) + \frac{16a^3}{3!}f'''(x) + \epsilon(5),$$

$$\frac{f(x+2a) - f(x-2a)}{a} = 4f'(x) + \frac{16a^2}{3!}f'''(x) + \epsilon(4). \tag{B.6}$$

It now remains to subtract B.6 from B.3 to obtain

$$f'(x) = \frac{8\left[f(x+a) - f(x-a)\right] - \left[f(x+2a) - f(x-2a)\right]}{12a}.$$

B.2 The Protein Replacement Model

aa	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Arg	0.6750	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Asn	0.5896	1.1891	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Asp	0.4625	0.6055	3.5734	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Cys	1.0654	0.3144	0.5899	0.2470	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Gln	1.1118	2.9678	2.2998	1.6861	0.2452	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Glu	1.0463	1.2018	1.2778	4.4000	0.0911	4.1597	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Gly	1.5880	0.5238	1.3749	0.7350	0.3171	0.5968	0.4638	-	-	-	-	-	-	-	-	-	-	-	-	-
His	0.5808	1.4571	2.2830	0.8393	0.4115	1.8122	0.8778	0.4763	-	-	-	-	-	-	-	-	-	-	-	-
Ile	0.4646	0.3596	0.4261	0.2668	0.4175	0.3153	0.3042	0.1802	0.2852	-	-	-	-	-	-	-	-	-	-	-
Leu	0.8044	0.5207	0.4101	0.2691	0.4508	0.6258	0.3208	0.2599	0.3640	4.1625	-	-	-	-	-	-	-	-	-	-
Lys	0.8320	4.9565	2.0376	1.1142	0.2742	3.5213	2.4160	0.5810	0.9859	0.3748	0.4980	-	-	-	-	-	-	-	-	-
Met	1.5467	0.8135	0.7378	0.3419	0.6186	2.0674	0.5318	0.4653	0.3809	3.6581	5.0023	0.6611	-	-	-	-	-	-	-	-
Phe	0.5462	0.3034	0.4252	0.2190	0.6692	0.4060	0.2242	0.3540	0.5762	1.4953	2.3926	0.2695	2.3069	-	-	-	-	-	-	-
Pro	1.2416	0.6558	0.7115	0.7756	0.1987	0.8501	0.7946	0.5883	0.4561	0.3662	0.4301	1.0361	0.3375	0.4811	-	-	-	-	-	-
Ser	3.4523	0.9101	2.5726	1.4409	0.9987	1.3483	1.2055	1.4021	0.8000	0.5306	0.4025	1.2346	0.9454	0.6132	1.2177	-	-	-	-	-
Thr	1.7514	0.8952	1.8232	0.9942	0.8473	1.3206	0.9496	0.5422	0.8304	1.1141	0.7798	1.2907	1.5515	0.7189	0.7809	4.4490	-	-	-	-
Trp	0.3501	0.6188	0.4224	0.3625	0.4457	0.7204	0.2613	0.3787	0.7244	0.5163	0.7948	0.4334	0.7684	3.2952	0.4999	0.4963	0.3837	-	-	-
Tyr	0.5732	0.6286	0.7200	0.4362	0.5563	0.7290	0.5072	0.2847	2.2110	0.5706	0.8110	0.6649	0.9325	5.8947	0.4337	0.5938	0.5235	2.9962	-	-
Val	2.0631	0.3887	0.4744	0.2757	0.9989	0.6344	0.5276	0.3147	0.3058	8.0028	2.1131	0.5262	1.7374	0.9838	0.5513	0.5075	1.8997	0.4296	0.7168	-

Table B.1: The PMB Matrix R

B.3 Pascarella and Argus Methods

In their analyses, PASCARELLA and ARGOS (1992) adopted the following scheme to detect the preferred environment, where arrows point at residues that flank an indel (in this case a deletion)

```
   ↓↓     ↓↓
XXZZXXXZZXXXXX
YYZ----ZYYYYY
  ↑     ↑
```

Define $\imath$ = percentage residue identity interval $\imath \in \{1-5, \ldots, 95-100\}$,

$n^{(\imath)}$ = number of indels within $\imath$,

$\ell_i^{(\imath)}$ = length of the ith indel within $\imath$,

$K^{(\imath)}$ = number of pairs within $\imath$,

$X_k^{(\imath)}$ = length of sequence X in pair k within $\imath$,

$r^{(\imath)}$ = number of indels per aligned site in interval $\imath$,

m_j = number of occurrences of flanking amino acid j,

p_j = preference for amino acid j.

The following are the essential Pascarella and Argus statistics:

$$\bar{\ell}^{(\imath)} = \frac{1}{n^{(\imath)}} \sum_{i=1}^{n^{(\imath)}} \ell_i^{(\imath)}, \tag{B.7}$$

$$r^{(\imath)} = \frac{\sum_{k=1}^{K^{(\imath)}} n_k^{(\imath)}}{\sum_{k=1}^{K^{(\imath)}} \min(X_k^{(\imath)}, Y_k^{(\imath)})}, \tag{B.8}$$

$$N = \sum_{j=1}^{20} m_j, \tag{B.9}$$

$$p_j = \frac{m_j}{N}, \tag{B.10}$$

$$\sigma_{p_j} = \left[\frac{p_j(1-p_j)}{N}\right]^{\frac{1}{2}}. \tag{B.11}$$

APPENDIX C
Chapter Four

C.1 Codon Preference

Codon	Preference (p)	SD(p)	Amino Acid	Hydrophilic
AAG	0.05714	0.01962	Lysine	Yes
GCA	0.05000	0.01842	Alanine	No
AAA	0.04286	0.01712	Lysine	Yes
ACA	0.04286	0.01712	Threonine	No
GAG	0.04286	0.01712	Glutamic acid	Yes
TAT	0.03571	0.01568	Tyrosine	No
ACT	0.02857	0.01408	Threonine	No
CTG	0.02857	0.01408	Leucine	No
GAT	0.02857	0.01408	Aspartic acid	Yes
TGG	0.02857	0.01408	Tryptophan	No
GGC	0.02857	0.01408	Glycine	Yes
GAC	0.02857	0.01408	Aspartic acid	Yes
GCC	0.02857	0.01408	Alanine	No
ATC	0.02143	0.01224	Isoleucine	No
ATA	0.02143	0.01224	Isoleucine	No
AGC	0.02143	0.01224	Serine	Yes
CAT	0.02143	0.01224	Histidine	No
AAT	0.02143	0.01224	Asparagine	Yes
GGT	0.02143	0.01224	Glycine	Yes
GGG	0.02143	0.01224	Glycine	Yes
GTG	0.02143	0.01224	Valine	No
GCT	0.02143	0.01224	Alanine	No
CTT	0.01429	0.01003	Leucine	No
CCT	0.01429	0.01003	Proline	Yes
AGA	0.01429	0.01003	Arginine	Yes
CAC	0.01429	0.01003	Histidine	No
ACG	0.01429	0.01003	Threonine	No
AGT	0.01429	0.01003	Serine	Yes
CCC	0.01429	0.01003	Proline	Yes
CAG	0.01429	0.01003	Glutamine	Yes
CGC	0.01429	0.01003	Arginine	Yes
TAC	0.01429	0.01003	Tyrosine	No
TCG	0.01429	0.01003	Serine	Yes
TTA	0.01429	0.01003	Leucine	No
GTA	0.01429	0.01003	Valine	No
TTG	0.01429	0.01003	Leucine	No
ATG	0.00714	0.00712	Methionine	No
AAC	0.00714	0.00712	Asparagine	Yes
ATT	0.00714	0.00712	Isoleucine	No
CTA	0.00714	0.00712	Leucine	No
CTC	0.00714	0.00712	Leucine	No
CCG	0.00714	0.00712	Proline	Yes
CAA	0.00714	0.00712	Glutamine	Yes
TGT	0.00714	0.00712	Cysteine	No
TTT	0.00714	0.00712	Phenylalanine	No
GGA	0.00714	0.00712	Glycine	Yes
CGT	0.00714	0.00712	Arginine	Yes
GAA	0.00714	0.00712	Glutamic acid	Yes
TCA	0.00714	0.00712	Serine	Yes
GTC	0.00714	0.00712	Valine	No
GCG	0.00714	0.00712	Alanine	No
TTC	0.00714	0.00712	Phenylalanine	No
GTT	0.00714	0.00712	Valine	No
TCC	0.00714	0.00712	Serine	Yes
TCT	0.00714	0.00712	Serine	Yes
AGG	0.00000	0.00000	Arginine	Yes
CCA	0.00000	0.00000	Proline	Yes
CGA	0.00000	0.00000	Arginine	Yes
CGG	0.00000	0.00000	Arginine	Yes
TGC	0.00000	0.00000	Cysteine	No
ACC	0.00000	0.00000	Threonine	No

Table C.1: This table shows the preference index of each codon in fast rate regions. None of the gaps in slow rate regions had flanking codons that met the criteria illustrated in Section B.3. The table is sorted by the preference index p in the second column.

C.2 Codon Usage in Regions 1 and 2

Codon	Slow	Fast	Slow+Fast	Slow÷Fast	Amino Acid	Hydrophilic
—	0.00008	0.01962	0.01970	0.00416	gap	NA
AGC	0.00459	0.00695	0.01154	0.65962	Serine	Yes
AGT	0.00328	0.00481	0.00809	0.68136	Serine	Yes
CGT	0.00413	0.00560	0.00973	0.73761	Arginine	Yes
TCA	0.00385	0.00517	0.00903	0.74448	Serine	Yes
AGA	0.00428	0.00514	0.00942	0.83175	Arginine	Yes
TCT	0.00588	0.00669	0.01257	0.87805	Serine	Yes
TCG	0.00330	0.00375	0.00705	0.87826	Serine	Yes
CGC	0.00690	0.00723	0.01413	0.95485	Arginine	Yes
CGG	0.00233	0.00243	0.00477	0.95973	Arginine	Yes
AGG	0.00299	0.00274	0.00573	1.08929	Arginine	Yes
CGA	0.00194	0.00170	0.00364	1.14423	Arginine	Yes
CTT	0.00761	0.00650	0.01410	1.17085	Leucine	No
CAA	0.00901	0.00746	0.01647	1.20788	Glutamine	Yes
CAG	0.00894	0.00733	0.01627	1.22049	Glutamine	Yes
ACA	0.00690	0.00539	0.01229	1.28182	Threonine	No
TTG	0.00792	0.00612	0.01404	1.29333	Leucine	No
TTA	0.01009	0.00744	0.01753	1.35526	Leucine	No
ACG	0.00537	0.00395	0.00932	1.35950	Threonine	No
TCC	0.00570	0.00406	0.00976	1.40161	Serine	Yes
AAA	0.01906	0.01351	0.03257	1.41063	Lysine	Yes
ACT	0.00720	0.00506	0.01226	1.42258	Threonine	No
CTA	0.00312	0.00209	0.00521	1.49219	Leucine	No
GCG	0.00868	0.00576	0.01444	1.50708	Alanine	No
CTC	0.00916	0.00584	0.01500	1.56704	Leucine	No
GTA	0.00640	0.00405	0.01044	1.58065	Valine	No
ATA	0.00800	0.00501	0.01301	1.59609	Isoleucine	No
AAT	0.01358	0.00850	0.02208	1.59693	Asparagine	Yes
ACC	0.01074	0.00664	0.01738	1.61671	Threonine	No
GCA	0.01066	0.00638	0.01704	1.67008	Alanine	No
TGT	0.00431	0.00258	0.00689	1.67089	Cysteine	No
AAG	0.01660	0.00976	0.02636	1.70067	Lysine	Yes
CCA	0.00640	0.00353	0.00992	1.81481	Proline	Yes
CAC	0.00860	0.00473	0.01333	1.81724	Histidine	No
CAT	0.00788	0.00423	0.01211	1.86486	Histidine	No
CCG	0.00690	0.00366	0.01056	1.88839	Proline	Yes
CCT	0.00754	0.00398	0.01152	1.89344	Proline	Yes
AAC	0.01340	0.00702	0.02042	1.90930	Asparagine	Yes
TGC	0.00496	0.00258	0.00754	1.92405	Cysteine	No
GCC	0.01785	0.00924	0.02709	1.93286	Alanine	No
CCC	0.00633	0.00326	0.00960	1.94000	Proline	Yes
CTG	0.01521	0.00777	0.02298	1.95798	Leucine	No
GCT	0.01384	0.00707	0.02091	1.95843	Alanine	No
GAG	0.01875	0.00942	0.02817	1.99133	Glutamic acid	Yes
GTT	0.01142	0.00571	0.01714	2.00000	Valine	No
ATT	0.01689	0.00836	0.02525	2.02148	Isoleucine	No
GAA	0.02588	0.01224	0.03812	2.11467	Glutamic acid	Yes
GTG	0.01505	0.00710	0.02215	2.11954	Valine	No
TAT	0.01262	0.00573	0.01834	2.20228	Tyrosine	No
GAT	0.02277	0.00961	0.03238	2.36842	Aspartic acid	Yes
ATC	0.01608	0.00671	0.02278	2.39659	Isoleucine	No
GGA	0.01237	0.00504	0.01741	2.45307	Glycine	Yes
ATG	0.01498	0.00592	0.02091	2.52893	Methionine	No
TTT	0.01472	0.00561	0.02033	2.62209	Phenylalanine	No
TGG	0.00917	0.00346	0.01263	2.65094	Tryptophan	No
TAC	0.01364	0.00514	0.01878	2.65397	Tyrosine	No
TTC	0.01410	0.00501	0.01911	2.81433	Phenylalanine	No
GAC	0.02102	0.00739	0.02841	2.84327	Aspartic acid	Yes
GTC	0.01226	0.00424	0.01650	2.88846	Valine	No
GGG	0.00796	0.00273	0.01069	2.92216	Glycine	Yes
GGT	0.01466	0.00468	0.01934	3.12892	Glycine	Yes
GGC	0.02153	0.00622	0.02774	3.46194	Glycine	Yes

Table C.2: This table shows the frequency of codons and of gaps in regions one and two. The table is sorted by the ratio slow÷fast in the fifth column.

APPENDIX D
Chapter Five

The following nine panels show results of pairwise alignments that had significance at the 10% level in the homopolymer experiment. This experiment consisted of 89 pyrosequenced reads. Each read was aligned with a cognate reference sequence using a three-region model as described in Chapter 5.

Each panel shows three pairwise alignments: the first was obtained under the null, the second under the alternative, and the third is the ClustalW alignment. The third row of the first and second alignments shows the predicted region number at each site. The third row of the third alignment shows the position number of each site. The table under these alignments shows estimator levels obtained under the null (first alignment) and under the alternative (second alignment). Only the second alignment was used for counting monoinserts.

Panel One

```
Alignments 46 (p-value = 0.0520)

CAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
00000000000000000000000111100000111111111111111100000001111111111111100011111221111111000000000000000000

CAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
00000000000000000000000111000000111111111111111100000001111111211111202111110011111110000000000000000000

CAACGCGAAGAACCTTACCTGGGTTTGACATC--CTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTG-ACAGGTGCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACAT-GCCGGCCGTGGAAACACGGCTTTCCAG-CTTG-CTGGAC-GTGTACACAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
```

Hyp	Est	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00000	0.00369	0.00369
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.10542	0.00000	0.00000
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	0.99706	9.94432	9.94432
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08452	0.18783	0.18783
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00000	0.00445	0.00445
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.20241	0.00000	0.23817
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	6.83567	9.93803	0.02069
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.10669	0.17262	0.98971

Panel Two

Alignments 47 (p-value = 0.0864)

```
CAACGCGAAGAACCTTACCCGGGCTCAAATGCTGGACGACAGTCCCTGA-AAGGGGATCTCCTTCGGG-CGTCCAGCAAGGTGCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAGATGAAATCCCCTGAAAAGGGGCTTTCCTTCGGGACATCTGTAGAGGTGNTGCATGGCTGTCG
00000000000000000001111122111111111111111111111000000000011111000000011111111111100000000000000000

CAACGCGAAGAACCTTACCCGGGCTCAAATGCTGGACGACAGTCCCTGA-AAGGGGATCTCCTTCGGG-CGTCCAGCAAGGTGCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAGATGAAATCCCCTGAAAAGGGGCTTTCCTTCGGGACATCTGTAGAGGTGNTGCATGGCTGTCG
00000000000000000001111000111111111111111111111000000000011111100000001111111111110000000000000000

CAACGCGAAGAACCTTACCCGGGCTCAAAT-GCTGGACGACAGTCCCTGAAA-GGGGATCTCCTTCGGG-CGTCCAGCA-AGGTGCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAG-ATGAAATCCCCTGAAAAGGGGCTTTCCTTCGGGACATCT-GTAGAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00089	0.00134	0.00134
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00019	0.00040	0.00040
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.84885	9.61214	9.61214
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08270	0.19471	0.19471
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00081	0.00147	0.00147
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00001	0.00049	0.24455
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.98295	9.51716	9.94296
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.09783	0.15918	0.96872

Panel Three

Alignments 60 (p-value = 0.0438)

```
CAACGCGAAGAACCTTACCTGGGCTCAAATGCAGAGTGACAGTCCCTGA-AAGGGGATTTTC--TTCGG-ACAGTCTGCAAGGTGATGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTAAATGTATGATGACCGCTTCTGAAAAG--GAGTTTCCCTTCGGGGCATTATACAAGGTGNTGCATGGCTGTCG
00000000000000000000000000011000022222200022212200000000000220000000000011122111100000000000000000000

CAACGCGAAGAACCTTACCTGGGCTCAAATGCAGAGTGACAGTCCCTGAAAAGGGGATTTT-CTTCGG-ACAGTCTGCAAGGTGATGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTAAATGTATGATGACCGCTTCTGAAAAGGAGTTTCCCTTCGGGGCATTATACAAGGTGNTGCATGGCTGTCG
00000000000000000000000000011110211111122211112220001111111111110000111111111110000000000000000000

CAACGCGAAGAACCTTACCTGGGCTCAAATGCA-GAGTGACAGTCCCTGAAAGGGGATTTTC--TTCGGA-CAGTCTGCAAGGTGATGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTAAATGTATGA-TGACCGCTTCTGAAAAGG-AGTTTCCCTTCGGGGCATTATACAAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00186	14.37794	14.37794
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.37635	0.00086	0.00086
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.97557	0.00145	0.00145
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08094	0.41164	0.41164
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00000	0.00312	0.00312
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.73446	0.00066	0.01016
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	2.60959	9.91673	5.37811
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08521	0.17917	0.51335

Panel Four

Alignments 68 (p-value = 0.0682)

```
CAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGATGCCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGC-ATGGCTGTCG
00000000000000000000000111100000111111111111111100000001111111111111000111110011111110000000000000000000

CAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGATGCCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGC-ATGGCTGTCG
00000000000000000000000111000000111111111111111100000001111111101111100011111001111111000000000000000000

CAACGCGAAGAACCTTACCTGGGTTTGACATC--CTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTG-ACAGGTGATGCCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACAT-GCCGGCCGTGGAAACACGGCTTTCCAG-CTTG-CTGGAC-GTGTACACAGGTGNTGC-ATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00052	0.00423	0.00423
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00378	0.00001	0.00001
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.98117	9.99302	9.99302
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.10275	0.20392	0.20392
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00076	0.00412	0.00412
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00001	0.00005	0.03863
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.92112	9.95712	9.87374
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.11122	0.17970	0.98975

Panel Five

Alignments 69 (p-value = 0.0480)

```
CAACGCGAAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGTTGCATGGCTGTCG
CAACGCGA-AGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
00000000000000000000000011110000011111111111111100000001111111111111000111110211111110000000000000000000

CAACGCGAAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGTTGCATGGCTGTCG
CAACGCGAA-GAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
00000000020000000000000111000000211111111111111000000011111110111110002111100111111100000000000000000000

CAACGCGAAAGAACCTTACCTGGGTTTGACATC--CTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTG-ACAGGTGTTGCATGGCTGTCG
CAACGCGAA-GAACCTTACCTGGGCTTGACATGTACAT-GCCGGCCGTGGAAACACGGCTTTCCAG-CTTG-CTGGAC-GTGTACACAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00087	0.00378	0.00378
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00058	0.00003	0.00003
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.94202	9.88836	9.88836
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08960	0.19874	0.19874
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00001	0.02245	0.02245
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.60435	0.00094	0.12263
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	0.90311	0.97378	9.96981
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.12911	0.20509	0.98978

Panel Six

Alignments 71 (p-value = 0.0693)

```
CAACGCGAAGAACCTTACCGGGGCTTGACATTCCCCTGAAGTCCCCGAGAAATCGGGGATCTCCCTTCGGGGACAGGGGAACAGGTGATGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAGATGAAATCCCCTGAAAA--GGGGCTTTC-CTTCG-GGACATCTGTAGAGGTGNTGCATGGCTGTCG
00000000000000000001111000000111111111201111111111111111111111110000000000001111111100000000000000000

CAACGCGAAGAACCTTACCGGGGCTTGACATTCCCCTGAAGTCCCCGAGAAATCGGGGATCTCCCTTCGGGGACAGGGGAACAGGTGATGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAGATGAAATCCCC-TGAAAA-GGGGCTTTC-CTTCG-GGACATCTGTAGAGGTGNTGCATGGCTGTCG
00000000000000000001111000000111111111001111111111111111111111110000000000001111111100000000000000000

CAACGCGAAGAACCTTACCGGGGCTTGACATTCCCC---TGAAGTCCCC-GAGAAATCGGGGATCTCCCTTCGGGGACAGGGGAACAGGTGATGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGA-A--CCGCAGATGAAATCCCCTGA-AAA--GGGGCTTTCC-TTCGGG-ACATCTGTAGAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00161	0.00177	0.00177
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00018	0.38727	0.38727
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.99143	9.97385	9.97385
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08015	0.16391	0.16391
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00152	0.00242	0.00242
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00035	0.26041	0.40765
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.94516	9.96302	0.08563
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.09735	0.14854	0.98846

Panel Seven

Alignments 79 (p-value = 0.0495)

```
CAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGGTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
00000000000000000000000111100000111111111111111000000011111111111100011111221111111000000000000000000

CAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGGTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
00000000000000000000000111000000111111111111111000000011111111211111202111110011111110000000000000000000

CAACGCGAAGAACCTTACCTGGGTTTGACATC--CTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTG-ACAGGTGGTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGACATGTACAT-GCCGGCCGTGGAAACACGGCTTTCCAG-CTTG-CTGGAC-GTGTACACAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00001	0.00388	0.00388
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.35452	0.00037	0.00037
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	0.16948	9.99292	9.99292
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08183	0.18493	0.18493
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00000	0.00451	0.00451
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.63219	0.00031	0.78393
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	5.57370	9.84978	0.01569
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.10670	0.17474	0.98977

Panel Eight

Alignments 82 (p-value = 0.0509)

```
CAAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGTTGCATGGCTGTCG
CA-ACGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
0000000000000000000000001111000001111111111111110000000111111111111100011111021111111000000000000000000

CAAACGCGAAGAACCTTACCTGGGTTTGACAT-CCTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTGACAGGTGTTGCATGGCTGTCG
CAA-CGCGAAGAACCTTACCTGGGCTTGACATGTACATGCCGGCCGTGGAAACACGGCTTTC-CAGCTTG-CTGGACGTGTACACAGGTGNTGCATGGCTGTCG
0002000000000000000000001110000002111111111111111000000011111111011111000211110011111110000000000000000000

CAAACGCGAAGAACCTTACCTGGGTTTGACATC--CTTTGACACCCCTGGAAACAGGGTTTTCCCGACTTGTCGGGACAGAGTG-ACAGGTGTTGCATGGCTGTCG
CAA-CGCGAAGAACCTTACCTGGGCTTGACATGTACAT-GCCGGCCGTGGAAACACGGCTTTCCAG-CTTG-CTGGAC-GTGTACACAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00066	0.00408	0.00408
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00028	0.00001	0.00001
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.81561	9.97371	9.97371
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.09486	0.19987	0.19987
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00000	0.02136	0.02136
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.24813	0.00000	0.05980
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	7.33091	1.22906	9.96107
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.12778	0.20234	0.98990

Panel Nine

Alignments 84 (p-value = 0.0925)

```
CAACGCGAAGAACCTTACCCGGGCTCAAATGCTGGACGACAGTCCCTGA-AAGGGGATCTCCTTCGGG-CGTCCAGCAAGGTNCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAGATGAAATCCCCTGAAAAGGGGCTTTCCTTCGGGACATCTGTAGAGGTGNTGCATGGCTGTCG
0000000000000000000111112211111111111111111111100000000001111100000001111111111110000000000000000

CAACGCGAAGAACCTTACCCGGGCTCAAATGCTGGACGACAGTCCCTGA-AAGGGGATCTCCTTCGGG-CGTCCAGCAAGGTNCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAGATGAAATCCCCTGAAAAGGGGCTTTCCTTCGGGACATCTGTAGAGGTGNTGCATGGCTGTCG
0000000000000000000111100011111111111111111111110000000001111110000000111111111111000000000000000000

CAACGCGAAGAACCTTACCCGGGCTCAAAT-GCTGGACGACAGTCCCTGAAA-GGGGATCTCCTTCGGG-CGTCCAGCA-AGGTNCTGCATGGCTGTCG
CAACGCGAAGAACCTTACCTGGGCTTGAACCGCAG-ATGAAATCCCCTGAAAAGGGGCTTTCCTTCGGGACATCT-GTAGAGGTGNTGCATGGCTGTCG
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

Hyp	Param	R1	R2	R3
Null	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00092	0.00126	0.00126
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00037	0.00089	0.00089
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.97271	9.96385	9.96385
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.08403	0.19355	0.19355
Alt	$\hat{t}_1, \hat{t}_2, \hat{t}_3$	0.00081	0.00158	0.00158
	$\hat{a}_1, \hat{a}_2, \hat{a}_3$	0.00124	0.00012	0.13754
	$\hat{r}_1, \hat{r}_2, \hat{r}_3$	9.82411	8.84830	9.26460
	$\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3$	0.09821	0.16224	0.97448

BIBLIOGRAPHY

ARRIBAS-GIL, A., E. GASSIAT, and C. MATIAS, 2006 Parameter Estimation in Pair-Hidden Markov Models. Scandinavian Journal of Statistics **33**: 651–71.

BALDI, P., Y. CHAUVIN, T. HUNKAPILLER, and M. A. MCCLURE, 1994 Hidden Markov models of biological primary sequence information. Proceedings of the National Academy of Sciences USA **91**: 1059–63.

BAUM, L. E., T. PETRIE, G. SOULES, and N. WEISS, 1970 A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. The Annals of Mathematical Statistics **41**: 164–71.

CHUONG, B. D., and S. BATZOGLOU, 2008 What is the expectation maximization algorithm? Nature Biotechnology **26**: 897–9.

CHURCHILL, G. A., 1989 Stochastic models for heterogeneous DNA sequences. Bulletin of Mathematical Biology **51**: 79–94.

DAYHOFF, M. O., R. V. ECK, M. A. CHANG, and M. R. SOCHARD, 1978 *Atlas of protein sequences and structure*. Silver Spring, Maryland: National Biomedical Research Foundation.

DEMPSTER, A. P., N. M. LAIRD, and D. B. RUBIN, 1977 Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society Series B (Methodological) **39**: 1–38.

DEVORE, J. L., 1990 *Probability and statistics for engineering and the sciences*. Brooks Cole Publishing Company.

DURBIN, R., S. R. EDDY, A. KROGH, and G. MITCHISON, 1998 *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

EDDY, S. R., 2003 HMMER User's Guide Biological sequence analysis using profile hidden Markov models.

EDGAR, R. C., editor, 2004 *MUSCLE: Multiple sequence alignment with improved accuracy and speed*, number 0-7695-2194-0/04. IEEE Computer Society, IEEE Xplore, 2001 L. Street N.W. Suite 700, Washington, DC 20036.

FELSENSTEIN, J., and G. A. CHURCHILL, 1996 A Hidden Markov Model approach to variation among sites in rate of evolution. Molecular Biology and Evolution **13**: 93–104.

GOFFE, W. L., G. D. FERRIER, and J. ROGERS, 1994 Global Optimization of Statistical Functions with Simulated Annealing. Journal of Econometrics **60**: 65–100.

GOLDMAN, N., J. L. THORNE, and D. T. JONES, 1996 Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. Journal of Molecular Biology **263**: 196–208.

GOLDMAN, N., and Z. YANG, 1994 A Codon-based Model of Nucleotide Substitution for Protein-coding DNA Sequences. Molecular Biology and Evolution **11**: 725–36.

GONNET, G. H., and S. A. BENNER, 1996 Probabilistic ancestral sequences and multiple alignments. In *Proceedings of the 5th SWAT conference, Reykjavik, Iceland*. 380–391.

GREENE, W. H., 1997 *Econometric Analysis*. Prentice Hall.

HAMILTON, J., 1994 *Time Series Analysis*. Princeton University Press, Princeton, New Jersey.

HASEGAWA, M., H. KISHINO, and T. YANO, 1985 Dating of the Human-Ape Splitting by a Molecular Clock of Mitochondrial DNA. Journal of Molecular Evolution **22**: 160–74.

HENIKOFF, S., and J. G. HENIKOFF, 1992 Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences USA **89**: 10915–9.

HOBOLTH, A., and J. L. JENSEN, 2005 Applications of Hidden Markov Models for Characterisation of Homologous DNA Sequences with a Common Gene. Journal of Computational Biology **12**: 186–203.

HUSE, S. M., J. A. HUBER, H. G. MORRISON, M. L. SOGIN, and D. M. WELCH, 2007 Accuracy and quality of massively parallel DNA pyrosequencing. Genome Biology **8**: R143.1–9.

ISAEV, A., 2004 *Introduction to Mathematical Methods in Bioinformatics*. Springer.

JAMSHIDIAN, M., and R. I. JENNRICH, 1997 Acceleration of the EM Algorithm by Using Quasi-Newton Methods. Journal of the Royal Statistical Society Series B (Methodological) **59**: 569–87.

KITAGAWA, G., 1987 Non-Gaussian State-Space Modeling of Nonstationary Time Series. Journal of the American Statistical Association **82**: 1032–41.

KNUDSEN, B., and M. M. MIYAMOTO, 2003 Sequence alignments and pair hidden Markov models using evolutionary history. Journal of Molecular Biology **333**: 453–60.

KOSIOL, C., I. HOLMES, and N. GOLDMAN, 2007 An Empirical Codon Model for Protein Sequence Evolution. Molecular Biology and Evolution **24**: 1464–79.

KROGH, A., M. BROWN, I. S. MIAN, K. SJOLANDER, and D. HAUSSLER, 1994 Hidden Markov Models in Computational Biology Applications to Protein Modeling. Journal of Molecular Biology **235**: 1501–31.

LAAN, N. C., D. F. PACE, and H. SHATKAY, 2006 Initial model selection for the Baum-Welch algorithm as applied to HMMs of DNA sequences. In *CSCBC First Canadian Student Conference on Biomedical Computing*.

LÖYTYNOJA, A., and N. GOLDMAN, 2008 A model of evolution and structure for multiple sequence alignment. Philosophical Transactions of the Royal Society B **363**: 3913–9.

MARGOLIASH, E., and E. L. SMITH, 1965 Structural and Functional Aspects of Cytochrome c in Relation to Evolution. in V. Bryson and H.J. Bogel eds. Evolving genes and proteins. Academic Press, New York.

MARGULIES, M., M. EGHOLM, W. E. ALTMAN, and ET. AL., 2005 Genome sequencing in microfabricated high-density picolitre reactors. Nature **437**: 376–80.

METROPOLIS, N., A. W. ROSENBLUTH, M. N. ROSENBLUTH, and A. H. TELLER, 1953 Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics **21**: 1087–92.

MEYER, I. M., and R. DURBIN, 2002 Comparative ab initio prediction of gene structures using pair HMMs. Bioinformatics **18**: 1309–18.

MEYER, M., U. STENZEL, and M. HOFREITER, 2008 Parallel tagged sequencing on the 454 platform. Nature Protocols **3**: 267–78.

MURPHY, K. M., and R. H. TOPEL, 1985 Estimation and Inference in Two-Step Econometric Models. Journal of Business and Economic Statistics **3**: 370–9.

NEI, M., 2005 Selectionism and Neutralism in Molecular Evolution. Molecular Biology and Evolution **22**: 2318–42.

NEI, M., and T. GOJOBORI, 1986 Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. Molecular Biology and Evolution **3**: 418–26.

NYRÉN, P., 1987 Enzymatic method for continuous monitoring of DNA polymerase activity. Analytical Biochemistry **167**: 235–8.

NYRÉN, P., B. PETTERSSON, and M. UHLÉN, 1993 Solid Phase DNA Minisequencing by an Enzymatic Luminometric Inorganic Pyrophosphate Detection Assay. Analytical Biochemistry **208**: 171–5.

PASCARELLA, S., and P. ARGOS, 1992 Analysis of insertions/deletions in protein structures. Journal of Molecular Biology **224**: 461–71.

PRESS, W. H., S. A. TEUKOLSKY, W. T. VETTERLING, and B. P. FLANNERY, 1992 *NUMERICAL RECIPES in C The Art of Scientific Computing*. Cambridge University Press.

RABINER, L. R., 1989 A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE **77**: 257–86.

RODRIGUEZ, F., J. L. OLIVER, A. MARIN, and J. R. MEDINA, 1990 The general stochastic model of nucleotide substitution. Journal of Theoretical Biology **142**: 485–501.

RONAGHI, M., S. KARAMOHAMED, B. PETTERSSON, M. UHLÉN, and P. NYRÉN, 1996 Real-Time DNA Sequencing Using Detection of Pyrophosphate Release. Analytical Biochemistry **242**: 84–9.

SAMMUT, R., P. MAXWELL, and G. A. HUTTLEY, 2006 Alignment of biological sequence pairs using two PHMMs. In *11th International Congress of Human Genetics, Brisbane, Australia*. ICMS, http://www.ichg2006.com/abstract/308.htm.

SCHUSTER, S. C., 2008 Next-generation sequencing transforms today's biology. Nature Methods **5**: 16–8.

THOMPSON, J. D., D. G. HIGGINS, and T. J. GIBSON, 1994 CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Research **22**: 4673–80.

THOMPSON, J. D., P. KOEHL, R. RIPP, and O. POCH, 2005 BAliBASE 3.0: Latest Developments of the Multiple Sequence Alignment Benchmark. PROTEINS: Structure, Function, and Bioinformatics **61**: 127–36.

THOMPSON, J. D., F. PLEWNIAK, and O. POCH, 1999a A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Research **27**: 2682–90.

THOMPSON, J. D., F. PLEWNIAK, and O. POCH, 1999b BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics **15**: 87–8.

THORNE, J. L., H. KISHINO, and J. FELSENSTEIN, 1992 Inching toward Reality: An Improved Likelihood Model of Sequence Evolution. Journal of Molecular Evolution **34**: 3–16.

VEERASSAMY, S., A. SMITH, and E. R. M. TILLIER, 2003 A transition probability model for amino acid substitutions from blocks. Journal of Computational Biology **10**: 997–1010.

WUYTS, J., G. PERRIERE, and Y. VAN DE PEER, 2004 The European ribosomal RNA database. Nucleic Acids Research **32**: 101–3.

YANG, Z., 1998 Likelihood ratio tests for detecting positive selection and application to primate lysozyme evolution. Molecular Biology and Evolution **15**: 568–73.

YANG, Z., and R. NIELSEN, 2000 Estimating Synonymous and Nonsynonymous Substitution Rates Under Realistic Evolutionary Models. Molecular Biology and Evolution **17**: 32–43.

ZUCKERKANDL, E., and L. PAULING, 1965 Evolutionary Divergence and Convergence in Proteins. in V. Bryson and H.J. Bogel eds. Evolving genes and proteins.