Fall 11-2009

SEQUENCE COMPARISON AND STOCHASTIC MODEL BASED ON MULTI-ORDER MARKOV MODELS

Xiang Fang
University of Nebraska at Lincoln, xfang@huskers.unl.edu

SEQUENCE COMPARISON AND STOCHASTIC MODEL BASED ON MULTI-ORDER MARKOV MODELS

by

Xiang Fang

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Statistics

Under the Supervision of Professors Walter W. Stroup, Shunpu Zhang and Guoqing Lu

Lincoln, Nebraska

November 2009

SEQUENCE COMPARISON AND STOCHASTIC MODEL BASED ON MULTI-ORDER MARKOV MODELS

Xiang Fang

University of Nebraska, 2009

Advisers: Walter W. Stroup, Shunpu Zhang and Guoqing Lu

This dissertation presents two statistical methodologies developed on multi-order Markov models. First, we introduce an alignment-free sequence comparison method, which represents a sequence using a multi-order transition matrix (MTM). The MTM contains information of multi-order dependencies and provides a comprehensive representation of the heterogeneous composition within a sequence. Based on the MTM, a distance measure is developed for pair-wise comparison of sequences. The new method is compared with the traditional maximum likelihood (ML) method, the complete composition vector (CCV) method and the improved version of the complete composition vector (ICCV) method using simulated sequences. We further illustrate the application of the MTM method using two real data sets, influenza A virus hemagglutinin gene sequence and complete mitochondrial genome sequences.

We then present a stochastic model named Multi-Order Markov Model under Hidden States (MMMHS) for representing heterogeneous sequences. MMMHS is similar to the conventional Hidden Markov Model (HMM) and Double Chain Markov Model (DCMM) in terms of using hidden states to describe the non-homogeneity of a sequence, but it provides a more flexible dependency structure by changing the order of Markov dependency under different hidden states. We extend the forward-backward procedure to MMMHS and provide the complete model estimation procedure based on

Expectation-Maximization (EM) algorithm. The method is then illustrated with applications on several real data sets, and the results are compared with that of traditional methods.

ACKNOWLEGEMENTS

I would like to thank Dr. Shunpu Zhang for the guidance and suggestions he has offered since I started my graduate study in Statistics in 2001. I greatly thank Dr. Walter Stroup for his guidance and support throughout my PhD study. Especially, I would like to thank Dr. Guoqing Lu for his support and research opportunities that lead to the completion of this dissertation. Many thanks go to Dr. Erin Blankenship for proof reading the dissertation proposal, to Dr. Steve Kachman for the great classes he taught. I would also like to thank Dr. Etsuko Moriyama and everyone for serving on my PhD supervisory committee. A special thank you goes out to my family and friends, especially my parents and wife, for their love and support.

List of Figures

List of Tables

CONTENTS

1. Introduction

1.1 Background Information

1.1.1 Alignment-Free Sequence Comparison Methods

The advances in biological sequencing technologies have generated an overwhelming amount of sequence data, which created tremendous opportunities for biologists and medical researchers to address both fundamental issues (e.g., molecular evolution) and practical problems (e.g., drug design). On the other hand, the increasing volume of data requires more efficient and reliable methods for sequence comparison. It has been realized that the traditional methods based on multiple sequence alignments are not suitable for large sequence data because of the fundamental and computational limitations, such as the difficulty of searching for optimal solutions and the ambiguity of choosing an evolutionary model (Attwood, 2000; Pearson, 2000; Vinga and Almeida, 2003; Wiens and Servedio, 1998). Consequently, considerable efforts have been made to research alternatives, i.e., alignment-free, methods for sequence comparison.

Alignment-free methods proposed in recent years can be classified into two major categories- methods based on the word frequencies, and methods that represent the sequence without using the word frequencies. To be more specific, the first category involves counting words of pre-selected lengths and the second category includes the use of certain data compression algorithms. These two categories originated from distinct mathematical theories with far more techniques explored in the published reports for the first category. The main interest of discussion here is in the first category.

As one of main categories, the first category of alignment-free sequence comparison methods can further be classified into three sub-categories. In the first sub-category, a genetic sequence is represented with a frequency vector of fixed length words and the majority of research has been focused on developing similarity or dissimilarity measures based on word frequencies (see Hao and Qi, 2004;

Helden, 2004; Kantorovitz et al., 2007; Qi et al., 2004; Stuart et al., 2002a; Stuart et al., 2002b; Wu et al., 1997; Wu et al., 2001). Vinga and Almeida (2003) provided an extensive review on various quantitative measures developed in recent years. The second sub-category (Li et al., 2002, Lu et al., 2008; Wu et al., 2006; Wu et al., 2007) compares sequences using the information extracted from multi-length word frequencies, realizing that the information contained in the vector of fixed length word frequencies is limited. To certain extent, this group of methods is an extension of the first sub-category. The third sub-category (Li and Sayood, 2005; Pham and Zuegg, 2004) includes the methods that represent the sequences using the transition matrix of a Markov chain of a pre-specified order k, and then compare the kth Markov models built for the sequences to obtain the dissimilarity measures. Unlike the first two sub-categories, this sub-category of methods compares sequences based on the relationships between frequencies of k-mer words and (k+1)-mer words.

1.1.2 Hidden Markov Models (HMM)

The basic theory of Hidden Markov Model (HMM) was introduced by Baum, Eagon, Petrie and others in a series of papers (Baum et al. (1966, 1967, 1968,1970), Baum, L. E. (1972)) published in the late 1960s and early 1970s. The applications of HMM were introduced to automatic speech recognition by researchers at IBM and Carnegie Mellon University in mid 1970s. However, the widespread understanding and application of HMM to either speech processing or computational biology did not occur until the mid 1980s. The earliest application of HMM on computational biology can be seen in Lander and Green (1987), where HMM was used in the construction of genetic linkage maps. Churchill (1989) introduced the application of HMM on modeling DNA sequences and searching for the coding region in DNA sequences. In the last several decades, HMM have been extensively applied in computational biology, such as database searching and multiple sequence alignment.

Hidden Markov models (HMM) consist of a Markov chain with a finite number of hidden states and an observable random sequence. Each of the hidden states is associated with a probabilistic distribution. Under the assumption of HMM, each discrete position of the random sequence has an unknown state that determines the value at that point based on its corresponding probabilistic distribution. The hidden Markov chain would change its state from point to point according to the transition probabilities. The definitions and notation of HMM are give below:

(1) Hidden Markov chain with a finite number of states. Let $\{X_t\}_{t=1}^{\infty}$ be a 1st-order Markov chain with a finite state space $S(X) = \{1,...,M\}$ with M states. The transition probability distribution is $A = \{a_{ij}\}$ with constraints $a_{ij} \geq 0$ and $\sum_{j \in S(X)} a_{ij} = 1$, where

$$a_{ij} = P(X_{t+1} = j \mid X_t = i),\ t \geq 1,\ i, j \in S(X). \tag{1.1}$$

The initial state distribution is $\pi = \{\pi_j\}$ with constraints $\pi_j \geq 0$ and $\sum_{j \in S(X)} \pi_j = 1$, where

$$\pi_j = P(X_1 = j),\ j \in S(X). \tag{1.2}$$

(2) Observable random sequence. Let $\{Y_t\}_{t=1}^{\infty}$ be a random sequence with a finite state space $S(Y) = \{1,...,K\}$. The conditional probability distribution, which relates the hidden Markov chain and the random sequence, is $B = \{b_k^j\}$ with constraints $b_k^j \geq 0$ and $\sum_{k \in S(Y)} b_k^j = 1$, where

$$b_k^j = P(Y_t = k \mid X_t = j),\ k \in S(Y), j \in S(X) \tag{1.3}$$

The above three probability distributions completely specify an HMM. Therefore, an HMM μ can be defined as $\mu = \{A, B, \pi\}$.

For an HMM, there are three basic questions that need to be answered.

(a). How do we compute the likelihood of the observed sequence given an HMM

μ?

(b). How do we estimate A, B and π given the observed sequence?

(c). How do we choose an optimal sequence of hidden states given an HMM μ and the observed sequence?

The first question is solved using the forward-backward procedure provided by Baum and his colleagues (Baum and Egon, 1967, Baum and Sell, 1968). They also solved the second question through the Baum-Welch algorithm (Baum et. al., 1970, Baum, 1972), which is also known as the EM (Expectation-Maximization) algorithm. For the third question, the most likely sequence of hidden states is computed by the Viterbi algorithm (Viterbi, 1967, Forney, 1973).

One major difference between HMM and Markov chain (MC) is that the states are hidden in HMM, but observable in MC. Another difference between MC and HMM is about the relation among the successive positions in a sequence. In a *k*th-order Markov chain, the value of a variable at position *t* is explained by the values of the variables at positions $t-k,...,t-1$. In the HMM, the values are not directly related or conditionally independent. Therefore, the probability of an observed sequence $Y_1,...Y_T$ given the model μ and the sequence of hidden states is:

$$P(Y_1 = k_1,...,Y_T = k_T \mid X_1 = j_1,...,X_T = j_T, \mu) = \prod_{t=1}^{T} b_{j_t}(k_t)\,. \tag{1.4}$$

The conditional independence among the successive positions in an HMM limits the capability of the model since a random sequence under a hidden Markov chain could certainly have successive positions that are directly correlated. Berchtold (1999) suggested the double chain Markov Model (DCMM), of which the HMM is a special case. The DCMM includes direct dependency among the successive positions into the HMM. Berchtold (2002) extended the DCMM to a high-order version,

where high order dependencies among the hidden states and the successive positions under a certain state are considered.

1.2 Problem Statement

The compositional structure of DNA sequences is heterogeneous due to the process of natural evolution, and is often found to be composed of locally homogeneous segments that are functionally important. It is of particular interest to find a stochastic model that could well describe the structure of a sequence. Classically, homogeneous MCs have been used to model DNA sequences, but MCs only provide good descriptions of local homogeneous structure, and are not appropriate for the overall heterogeneous structure of DNA sequences. Churchill (1989) suggested HMM for DNA sequence modeling so that each of these homogeneous segments can be classified into one of the hidden states. Berchtold (1999 and 2002) proposed DCMM for non-homogeneous sequence modeling. Similar to HMM, the DCMM also classifies each homogeneous segments into a hidden state, but it assumes a Markov dependent structure among successive bases within a homogeneous segment while the HMM assumes an independent structure. In other words, the HMM assume that each individual homogeneous segment follows an independent model, i.e. 0^{th}-order Markov model, while the DCMM assumes that each individual homogeneous segment follows a Markov model with positive order. Clearly, the newly developed DCMM can handle some situations that both HMM and MC can not, but there still exist some cases that are not considered by current MC, HMM and DCMM. As part of the study, we introduce a stochastic model for non-homogeneous sequences. The new model is similar to both DCMM and HMM in terms of assigning a hidden state to each homogeneous segment, but it assumes that both the dependence structure and the independence structure are possible for a homogeneous segment. Therefore, under the new model, each individual homogeneous segment can be modeled as either an independent sequence or a Markov sequence.

Besides the stochastic model for heterogeneous DNA sequences described above, I would also introduce a statistical method for comparing these types of sequences. Despite the widely discussed heterogeneous compositional properties of DNA structure, most existing alignment-free sequence comparison methods are based on the frequencies of words of pre-fixed length, which contain very limited information about the compositional features of DNA sequences. In this study, I will provide an alignment-free method that compares two heterogeneous sequences using a measure based on multi-order Markov chains. The main advantage of the multi-order Markov chains based method is that it takes into consideration the heterogeneous structure of DNA sequences while comparing two sequences. An order selection method is also introduced to identify the most informative orders that account for the majority of the heterogeneity in DNA sequences.

1.3 Research Objectives

Throughout this study, the following research objectives will be addressed:

1. To develop an alignment-free sequence comparison method based on multi-order Markov chain models.
2. To compare the performance of the proposed sequence comparison method with other current alignment-free and alignment-based methods.
3. To apply the new method to lineage analysis of Influenza A viruses
4. To develop an HMM based stochastic model that could handle the situation in a DNA sequence, where successive positions under a hidden state could have both dependence and independence structure.
5. To apply the proposed model to analyze DNA sequences and some other sequences, and compare the performance of the proposed model with MC, HMM and DCMM.

2. Literature Review

2.1 Alignment-Free Sequence Comparison Methods

Alignment-free methods based on word frequencies can be classified into three sub-categories. There are unequal amounts of literature for each one of the three sub-categories, with most publications in the first category and fewest in the third category. Review of these publications is conducted by category.

2.1.1 Category I: Alignment-free Methods Based on Frequency Vector of Fixed Length Words

The feature of methods in this category is that a sequence is resolved into overlapping words of pre-selected length L and represented as a frequency vector of all possible L-words. Let X be a DNA sequence of length n_X, the number of times of each possible L-word with overlapping capacity observed in X can be represented by

$$C_L^X = (c_{L,1}^X, ..., c_{L,G}^X), \tag{2.1}$$

where $G = 4^L$ is the number of all possible L-words. Then the frequency vector is given as:

$$F_L^X = (f_{L,1}^X, ..., f_{L,G}^X), \tag{2.2}$$

where $f_{L,i}^X = \dfrac{c_{L,i}^X}{n_X - L + 1}$ is the relative count of the ith L-word. With DNA sequences mapped into the space defined by the frequency vector of a fixed L, the pair-wise distances among all the sequences can be computed. Research interests in this category have focused on developing a distance function $d(X,Y)$ that can correctly identify the similarity or dissimilarity measure between DNA sequences, identifying an optimal value for L, as well as understanding the statistical properties of the frequency vector. Here $d(X,Y)$ is the distance function that assigns a real number as the distance between each pair of X and Y from a given data set.

Euclidean type distance is one of the distance functions that were researched in depth in the early phase of searching for a reliable distance function. Zharkikh and Rzhetsky (1993) documented the statistical properties of the standard Euclidean distance calculated from the frequency vectors of *L*-words and the statistical relationships between the resulting distances when different values of *L* are used. Their study demonstrated that *L*-word frequencies are very useful for explaining evolutionary relationships between DNA sequences, and that frequencies of longer words tend to have a distribution that is more similar to an independent sequence than that of shorter words.

Torney *et al.* (1990) pointed out that different *L*-words may contribute differently to the standard Euclidean distance, which lead to the exploration of the weighted Euclidean distance:

$$d(X,Y) = \sum_{i=1}^{G} \pi_i (f_{L,i}^X - f_{L,i}^Y)^2 \tag{2.3}$$

In Eq. (2.3), π_i is the weight assigned to the *i*th *L*-word based on its frequency. Summing the weighted Euclidean distance over different values of *L* provides a new type of distance function, which is designated as *d2* distance and has been used as a tool for database searches (Hide *et al.*, 1994). The same report also talked about the identification of the optimal value for *L* and showed that the search results were similar to that of FASTA when *L*=8 was used for the particular case discussed. Due to its computational efficiency, *d2* distance has been widely applied to classification of EST sequences (Burke *et al.*, 1998, Burke *et al*., 1999, Miller *et al.*, 1999, Davison and Burke, 2001).

Other Euclidean type distances that have been explored are the Mahalanobis distance and the standard Euclidean distance:

$$D(X,Y) = (F_L^X - F_L^Y)^T S^{-1} (F_L^X - F_L^Y), \tag{2.4}$$

where *S* is the covariance matrix of the frequency vector under the model assumption. Eq (2.4), the Mahalanobis distance, can be reduced to the standard Euclidean distance:

$$D(X,Y) = (F_L^X - F_L^Y)^T [diag(s_{11},...,s_{GG})]^{-1} (F_L^X - F_L^Y), \quad (2.5)$$

when *S* is not invertible or too complex to invert. $s_{11},...,s_{GG}$ are the diagonal elements in *S*.

Wu *et al.* (1997) introduced the use of these two distances into the area of sequence comparison. In this report, they adopted a covariance structure of the word frequencies under the independent model of base composition, which was originally derived by Gentleman and Mullin (1989). Wu *et al.* (2001) further extended both the Mahalanobis distance and the standard Euclidean distance to the Markov chain models of base composition. Evaluation using a human lipoprotein lipase data set showed that both distances had better selectivity and sensitivity than the standard Euclidean distance.

Wu *et al.* (2001) also examined the application of the Kullback-Leibler discrepancy, an information theory based distance:

$$d(X,Y) = \sum_{i=1}^{G} f_{L,i}^X \log_2 (\frac{f_{L,i}^X}{f_{L,i}^Y}). \quad (2.6)$$

The report concluded that the Mahalanobis distance is the distance of best performance in term of selectivity and sensitivity, followed by the standardized Euclidean distance and then the Kullback-Leibler discrepancy. All these three distance functions outperformed the traditional Euclidean distance.

The correlation coefficient is another type of distance function that has been used to measure the similarity or dissimilarity between sequences. The distance between two sequences *X* and *Y* then is defined as the linear correlation coefficient between the two frequency vectors F_L^X and F_L^Y as detailed below:

$$d(X,Y) = \frac{G\sum_{i=1}^{G} f_{L,i}^X \cdot f_{L,i}^Y - \sum_{i=1}^{G} f_{L,i}^X \cdot \sum_{i=1}^{G} f_{L,i}^Y}{[G\sum_{i=1}^{G} (f_{L,i}^X)^2 - (\sum_{i=1}^{G} f_{L,i}^X)^2]^{\frac{1}{2}} \cdot [G\sum_{i=1}^{G} (f_{L,i}^Y)^2 - (\sum_{i=1}^{G} f_{L,i}^Y)^2]^{\frac{1}{2}}}. \quad (2.7)$$

This distance function has been applied to classify protein sequences based on dipeptide frequencies (Petrilli, 1993) and used to query large sequence databases (Petrilli and Tonukari, 1997). The results have shown that only a fraction of all the possible dipeptide frequencies were needed to provide correct classification based on the correlation coefficient. Although this type of distance function is not as popularly pursued as the Euclidean type of distance function, it provided a possible path for the future development of the distance function.

Stuart *et al.* (2002a, b) proposed a new sequence comparison method. The new method used the singular value decomposition of the *L*-words frequency matrix to represent whole genome protein sequences as high-dimensional vectors, then calculated the pair-wise distances based on the angle cosine between these vectors:

$$d(X,Y) = -\ln \frac{1 + C(X,Y)}{2}, \tag{2.8}$$

where $C(X,Y) = \dfrac{\sum_{j=1}^{\omega} \sigma_j^X \cdot \sigma_j^Y}{\left[\sum_{j=1}^{\omega} (\sigma_j^X)^2 \cdot \sum_{j=1}^{\omega} (\sigma_j^Y)^2 \right]^{\frac{1}{2}}}$. σ_j^X and σ_j^Y are the resulting vectors of singular value decomposition for F_L^X and F_L^Y. Here ω is the dimension of σ_j^X and σ_j^Y. The singular value decomposition reduced the dimension of the resulting vectors by using only the high value eigenvalues, which enhanced the computational efficiency on the one hand and reduced the white noise from the information on the other hand.

Qi, J. *et al.* (2004) proposed a composition vector (CV) method for inferring whole proteome prokaryote phylogeny. In the CV method, the observed frequency of each *L*-word is normalized using an estimated expected frequency, which is based on the observed frequencies of shorter words. The normalization was originally proposed by Brendel *et al.* (1986) and has been used with minor

modifications for phylogenetic studies of prokaryotes and viruses in Qi, J. *et al.* (2004). The normalized observed frequency vector of all possible *L*-words is the composition vector of a sequence, which is used to calculate the pair-wise distance. Qi, J. *et al.* (2004) adopted a distance function slightly different from Eq. (2.8) as detailed below:

$$d(X,Y) = \frac{1 - C(X,Y)}{2}, \tag{2.9}$$

where $C(X,Y) = \dfrac{\sum_{j=1}^{4^L} cv_{L,j}^X \cdot cv_{L,j}^Y}{\left[\sum_{j=1}^{4^L} (cv_{L,j}^X)^2 \cdot \sum_{j=1}^{4^L} (cv_{L,j}^Y)^2\right]^{\frac{1}{2}}}$. The CVs for sequences *X* and *Y* are $CV_L^X = (cv_{L,1}^X, ..., cv_{L,4^L}^X)$ and $CV_L^Y = (cv_{L,1}^Y, ..., cv_{L,4^L}^Y)$, respectively. Some elements in CV_L^X and CV_L^Y could be zero as the corresponding estimated expected frequencies are zero. In the report, the CV method was applied with *L*=5 and 6, and the results were generally consistent with the commonly accepted phylogenies.

2.1.2 Category II: Alignment-Free Methods Based on Frequency Vector of Multi-Length Words

Compared to the first category, the second category features resolving a sequence into overlapping words of multiple lengths, i.e., *L* is not a fixed integer but a set of integers. The purpose of using multiple lengths is to avoid the ambiguity of choosing an optimal *L* value and to attain more complete information from a sequence.

Only a few reports have been published in this category. One of the recent reports is Li *et al.* (2002), which introduced the Complete Information Set (CIS) to the field of sequence comparison and phylogenetic studies. Under the concept of CIS method, a sequence *X* of length n_X has a complete information set U^X, which contains all the primary information of *X* and is defined as below:

$$U^X = (F_1^X, ..., F_{n_X}^X), \tag{2.10}$$

where F_L^X for $L \in [1, n_X]$ is the frequency vector defined in Eq. (2.2). Although the CIS of sequence X is defined on every L in the range $[1, n_X]$, only a single frequency vector is chosen to calculate the pair-wise distance. An empirical formula was provided in Li *et al.* (2002) for choosing that particular vector, but the origin of the formula remained unclear in the report. For the distance function, the Kullback-Leibler discrepancy [Eq. (2.7)] was used. The resulting phylogenetic tree based on whole genome sequences was highly consistent and supported separate monophyletic cluster of species with similar phenotype. Although the CIS method does not actually compare the sequences using information of multi-length words, it indicates that the information extracted from multi-length words is comprehensive compared to fixed-length words.

One difficulty of combining multi-length word frequencies is the different scales associated with words of different lengths. Shorter words are always observed with higher frequency, which tends to have more influence on the similarity measure than longer words. Therefore, it is necessary to standardize word frequencies before combining multi-length word frequencies. Wu *et al.* (2006) proposed the Complete Composition Vector (CCV) method, which extended the CV method mentioned in Section 2.1.1 by concatenating CVs from multi-length words into a single vector. Since the CCV consists of the normalized frequencies of multi-length words, the effect of a single word on the resulting distance depends on the difference between the observed frequency from the expected frequency. Thus, a word is considered to be information rich when its observed frequency deviates significantly from the expected frequency.

The method to estimate the expected frequencies of L-words in X was originally introduced by Brendel *et al.* (1986). Given the observed frequencies of (L-1)-words and (L-2)-words, the expected frequency of L-words is estimated as:

$$\hat{f}^X(\alpha_1,...,\alpha_L) = \frac{f^X(\alpha_1,...,\alpha_{L-1}) \cdot f^X(\alpha_2,...,\alpha_L)}{f^X(\alpha_2,...,\alpha_{L-1})}, \tag{2.11}$$

for $3 \le L \le n_X$. The normalization function is given as:

$$a^X(\alpha_1 \cdots \alpha_L) = \frac{f^X(\alpha_1 \cdots \alpha_L) - \hat{f}^X(\alpha_1 \cdots \alpha_L)}{\hat{f}^X(\alpha_1 \cdots \alpha_L)}. \tag{2.12}$$

The normalized observed frequencies of multi-length words are then used to calculate the pair-wise distance. Both the Euclidean distance function and the angle cosine distance function have been found to be applicable with CCV. For the convenience of computation, Wu *et al.* (2006) set the upper limit of L to 7. The CCV method was found to provide finer evolutionary information than the CV method using a data set of 103 microbes and 6 eukaryotes.

A potential problem associated with the normalization function [Eq. (2.12)] was pointed out by Lu *et al.*, (2008). As the expected frequency of a L-word $\alpha_1 \cdots \alpha_L$ is estimated by the observed frequencies of $\alpha_1 \cdots \alpha_{L-1}$ and $\alpha_2 \cdots \alpha_L$, there is a positive correlation between the observed frequency $f^X(\alpha_1 \cdots \alpha_L)$ and the estimated expected frequency $\hat{f}^X(\alpha_1 \cdots \alpha_L)$. Therefore, the difference between $f^X(\alpha_1 \cdots \alpha_L)$ and $\hat{f}^X(\alpha_1 \cdots \alpha_L)$ tends to be smaller than the difference between $f^X(\alpha_1 \cdots \alpha_L)$ and the true expected frequency, which indicates that the information contributed by selective evolution is underestimated. Lu *et al.* (2008) also provided an improved version of CCV (ICCV), where the estimated expected frequency is replaced by the exact expected frequency and variance under the uniform and independent model of base composition. Results from a simulated data set showed that the ICCV method is more robust in resolving phylogenetic relationships of remotely related clades than the existing CCV method. The improved method was also applied on a set of 54 influenza A viral HA sequences for phylogeny inference. The resulting tree was highly consistent with the tree generated by the alignment-based maximum likelihood method.

A major disadvantage for both CCV and ICCV is the high dimension of the vector space when the upper limit of L is large, which significantly decreases the computational efficiency. Lu *et al.* (2008) suggested that increasing the upper limit of L might not improve the phylogenetic reconstruction due to the overlapping nature of words. They also introduced a numerical method for choosing a optimal upper limit for L. For CCV method, Wu *et al.* (2007) also proposed a method to limit the dimension of CCV by only selecting information rich words into the composition vector.

2.1.3 Category III: Alignment-Free Methods Based on Markov Models

The methods in the third category model the sequences as a Markov chain of a pre-specified order k, and then compare estimated transition matrices to obtain the dissimilarity measures. Therefore, the comparison is based on the relationships between frequencies of k-words and (k+1)-words.

One of the earliest reports is Gibbs *et al.* (1971), where the transition matrix was used in describing and classifying proteins by their amino acid sequences. As describing the sequences, Gibbs *et al.* (1971) used amino acid doublet frequencies. Each sequence was modeled as a 1^{st}-order Markov chain, then the resulting 1^{st}-order transition matrices were classified by the CENTPERC program. Blaisdell (1986) also represented eukaryotic DNA sequences as 1^{st}, 2^{nd} and 3^{rd}-order Markov chains separately and developed Chi-square tests to assess the homogeneity of sets of sequences based on the transition matrices.

Almagor (1983), Blaisdell (1985) and Phillips *et al.* (1987) also presented Markov analysis of biological sequences and discussed the criterion of choosing a proper Markov order. In some of these studies, it was found that the order of Markov dependencies may vary from segment to segment. Scherer *et al.* (1994) applied a 7^{th}-order Markov model to detect the segments with different patterns. The results in Fickett and Tung (1992) showed that 5^{th}-order Markov model worked best for finding genes in

protein sequences. Generally, these studies showed that the base composition of biological sequences could be highly heterogeneous, but consist of homogeneous local segments.

Li *et al.* (2005) proposed a genome signature based on a triplet Markov chain model. Instead of considering a single nucleotide as a unit, the triplet Markov chain used the nucleotide triplet as a single unit to calculate the transition probability. As the transition varies in the directions of 5` to 3` and 3` to 5`, the transition probability is calculated as the average of the transition probabilities of two directions. Let T_i be the ith triplet out of the 64 possible triplets for DNA sequences. The triplet transition probability for a 1st-order triplet Markov chain is defined as:

$$P(T_i \mid T_j) = \frac{P_{5\to 3}(T_i \mid T_j) + P_{3\to 5}(T_i \mid T_j)}{2}\ , \tag{2.13}$$

where $P_{5\to 3}(\cdot)$ and $P_{3\to 5}(\cdot)$ are transition probabilities for 5` to 3` and 3` to 5` directions. The resulting triplet transition matrix is called the signature matrix. The absolute differences between these matrices then can be utilized as the distance measures for constructing phylogenetic trees.

Pham and Zuegg (2004) proposed a probabilistic measure for sequence comparison. This method is based on the Markov modeling of DNA sequences. They suggested a probabilistic approach to compare the Markov models to obtain the similarity measures. The method starts by modeling DNA sequences using a Markov chain of a pre-selected order. Denote the fitted Markov models for sequences X and Y by λ_X and λ_Y, respectively. A probabilistic distance between X and Y is then defined as:

$$d(\lambda_X, \lambda_Y) = 1 - \exp[\frac{D(\lambda_X, \lambda_Y) + D(\lambda_Y, \lambda_X)}{2}], \tag{2.14}$$

where $D(\lambda_X, \lambda_Y)$ is the approximate Kullback-Leibler divergence (KLD).

$$D(\lambda_X, \lambda_Y) = \frac{1}{n_Y} \log \frac{p(Y \mid \lambda_X)}{p(Y \mid \lambda_Y)}, \tag{2.15}$$

where $p(Y \mid \lambda_X)$ and $p(Y \mid \lambda_Y)$ are the probabilities of observing sequence Y under models λ_X and λ_Y, respectively. Since the KLD is not symmetric, a symmetrized version of KLD is used in Eq. (2.14):

$$D(\lambda_Y, \lambda_X) = \frac{1}{n_X} \log \frac{p(X \mid \lambda_Y)}{p(X \mid \lambda_X)}. \tag{2.16}$$

The proposed probabilistic measure based on the 1st-order Markov model was tested against the data set used in Wu *et al.* (2001) and compared to Wu's methods. The results showed that the probabilistic measure achieved better selectivity than Wu's Mahalanobis and standardized Euclidean distances and had the same sensitivity.

2.2 Stochastic Models

2.2.1 Hidden Markov Models (HMM)

Baum et al. (1966) first introduced the basic theory of HMM. Later, a series of papers (Baum et al. (1967, 1968, 1970), Baum, L. E. (1972)) were published by Baum and his colleagues to continue the discussion on HMM and the likelihood maximization techniques for model estimation. The iterative algorithm for maximizing the likelihood in HMM provided by Baum et al. (1970) was further developed by Dempster et al. (1977) as the EM algorithm. The widespread application of HMM on speech recognition and computational biology occurred in the late 1980s, especially after the publication of several tutorial materials on HMM.

One of the most popular tutorial reports, Rabiner, R. L. (1989), thoroughly discussed the three fundamental questions, which are:

1) How to compute the probability of the observed sequence, given the HMM.
2) How to adjust the model parameters so that the probability of the observed sequence can be maximized, and

3) How to find an optimal hidden state sequence, given both the HMM and the observed sequence.

For the first question, the likelihood of the observed sequence given the HMM μ must be computed,

$$L = P(Y_1,...,Y_T \mid \mu). \tag{2.17}$$

The likelihood can be computed by the forward procedure (Baum and Egon, 1967, Baum and Sell, 1968) as follows. To illustrate the forward-backward algorithm and its extension in other models included in the study , we adopt the system of notations from Berchtold (2002). Let the forward variable

$$\alpha_t(j) = P(Y_1,...,Y_t, X_t = j) \tag{2.18}$$

be the joint probability of the partial observed sequence $Y_1,...,Y_t$ and the *t*th hidden state X_t given the HMM μ. For the simplicity of notation, it will not be indicated that the probability is conditional on the model μ. For $t = 1$, Eq. (2.18) becomes

$$\alpha_1(j) = P(Y_1, X_1 = j) = P(X_1 = j)P(Y_1 \mid X_1 = j) = \pi_j b^j_{y_1}, \tag{2.19}$$

for $j \in S(X)$. For $t = 2$, it is

$$\begin{aligned}
\alpha_2(j) &= P(Y_1, Y_2, X_2 = j) \\
&= \sum_{i \in S(X)} P(Y_1, Y_2, X_1 = i, X_2 = j) \\
&= \sum_{i \in S(X)} P(Y_1, Y_2 \mid X_1 = i, X_2 = j)P(X_1 = i, X_2 = j) \\
&= \sum_{i \in S(X)} P(Y_1 \mid X_1 = i)P(Y_2 \mid X_2 = j)P(X_2 = j \mid X_1 = i)P(X_1 = i) \\
&= \sum_{i \in S(X)} P(Y_1, X_1 = i)P(Y_2 \mid X_2 = j)P(X_2 = j \mid X_1 = i) \\
&= b^j_{y_2} \sum_{i \in S(X)} \alpha_1(i)a_{ij}
\end{aligned} \tag{2.20}$$

By induction, it can be shown

$$\alpha_{t+1}(j) = b^{j}_{y_{t+1}} \sum_{i \in S(X)} \alpha_t(i) a_{ij} \,, \tag{2.21}$$

for $1 \le t \le T-1$. Based on the forward procedure, the likelihood of the complete observed sequence is

$$L = \sum_{j \in S(X)} \alpha_T(j) \,. \tag{2.22}$$

In a similar way, the backward procedure can be defined as follows. Let the backward variable

$$\beta_t(j) = P(Y_{t+1}, ..., Y_T \mid X_t = j) \tag{2.23}$$

be the joint probability of the partial observed sequence $Y_t, ..., Y_T$ given the tth hidden state X_t and the HMM μ. For $t = T$, Eq. (2.23) is defined as

$$\beta_T(j) = 1, \tag{2.24}$$

for $j \in S(X)$. For $t = T-1$, it becomes

$$\begin{aligned}
\beta_{T-1}(j) &= P(Y_T \mid X_{T-1} = j) \\
&= P(Y_T, X_{T-1} = j) / P(X_{T-1} = j) \\
&= \sum_{i \in S(X)} P(Y_T, X_{T-1} = j, X_T = i) / P(X_{T-1} = j) \\
&= \sum_{i \in S(X)} P(Y_T \mid X_{T-1} = j, X_T = i) P(X_{T-1} = j, X_T = i) / P(X_{T-1} = j) \\
&= \sum_{i \in S(X)} P(Y_T \mid X_{T-1} = j, X_T = i) P(X_T = i \mid X_{T-1} = j) \\
&= \sum_{i \in S(X)} P(Y_T \mid X_T = i) P(X_T = i \mid X_{T-1} = j) \\
&= \sum_{i \in S(X)} b^{i}_{y_T} a_{ji}
\end{aligned} \tag{2.25}$$

For $t = T-2$, it becomes

$$\begin{aligned}
\beta_{T-2}(j) &= P(Y_{T-1}, Y_T \mid X_{T-2} = j) \\
&= P(Y_{T-1}, Y_T, X_{T-2} = j) / P(X_{T-2} = j)
\end{aligned}$$

$$= \sum_{i \in S(X)} P(Y_{T-1}, Y_T, X_{T-2} = j, X_{T-1} = i) / P(X_{T-2} = j)$$

$$= \sum_{i \in S(X)} P(Y_{T-1} \mid Y_T, X_{T-2} = j, X_{T-1} = i) P(Y_T = k_T, X_{T-2} = j, X_{T-1} = i) / P(X_{T-2} = j)$$

$$= \sum_{i \in S(X)} P(Y_{T-1} \mid X_{T-1} = i) P(X_{T-1} = i \mid X_{T-2} = j) P(Y_T \mid X_{T-2} = j, X_{T-1} = i)$$

Note: It can be proved that $P(Y_T \mid X_{T-2} = j, X_{T-1} = i) = P(Y_T \mid X_{T-1} = i)$ using (2.25).

$$= \sum_{i \in S(X)} b^i_{y_{T-1}} a_{ji} \beta_{T-1}(i) \tag{2.26}$$

By induction, it can be shown

$$\beta_t(j) = \sum_{i \in S(X)} b^j_{y_{t+1}} a_{ji} \beta_{t+1}(i), \tag{2.27}$$

for $1 \le t \le T-1$. Based on both forward and backward procedures, the likelihood of the complete observed sequence can be written as:

$$L = \sum_{j \in S(X)} \alpha_t(j) \beta_t(j), \tag{2.28}$$

for $t = 1, ..., T$. Eq. (2.24) is equivalent to $t = T$.

For the second question, the estimation of the three sets of probabilities A, B and π in HMM μ can be done using the EM algorithm, which is also know as the Baum-Welch algorithm (Baum and Egon, 1967, Baum et al., 1970, Baum, 1972). To describe the algorithm, the following variable is defined:

$$\gamma_t(j) = P(X_t = j \mid Y_1, ..., Y_T), \tag{2.29}$$

which is the probability of j being the tth hidden state given the observed sequence and μ. Eq. (2.29) can also be written in terms of the forward and backward variables as below:

$$\gamma_t(j) = \frac{P(Y_1, ..., Y_T, X_t = j)}{P(Y_1, ..., Y_T)} = \frac{\alpha_t(j) \beta_t(j)}{\sum_{j \in S(X)} \alpha_t(j) \beta_t(j)}, \tag{2.30}$$

with $\sum_{j \in S(X)} \gamma_t(j) = 1$. Then the joint probability of j being the tth hidden state and i being the (t+1)th hidden state given the observed sequence and μ is defined as:

$$\varepsilon_t(j,i) = P(X_t = j, X_{t+1} = i \mid Y_1, ..., Y_T). \tag{2.31}$$

In term of the forward and backward variables, Eq. (2.31) can be expressed as:

$$\varepsilon_t(j,i) = \frac{P(Y_1, ..., Y_T, X_t = j, X_{t+1} = i)}{P(Y_1, ..., Y_T)}$$

$$= \frac{1}{P(Y_1, ..., Y_T)} P(Y_1, ..., Y_t, X_t = j)\ P(X_{t+1} = i \mid Y_1, ..., Y_t, X_t = j) P(Y_{t+1} \mid Y_1, ..., Y_t, X_t = j, X_{t+1} = i)$$

$$P(Y_{t+2}, ..., Y_T \mid Y_1, ..., Y_{t+1}, X_t = j, X_{t+1} = i)$$

$$= \frac{1}{P(Y_1, ..., Y_T)} P(Y_1, ..., Y_t, X_t = j)\ P(X_{t+1} = i \mid X_t = j) P(Y_{t+1} \mid X_{t+1} = i) P(Y_{t+2}, ..., Y_T \mid X_{t+1} = i)$$

$$= \frac{\alpha_t(j) a_{ji} b^i_{y_{t+1}} \beta_{t+1}(i)}{\sum_{i \in S(X)} \sum_{j \in S(X)} \alpha_t(j) a_{ji} b^i_{y_{t+1}} \beta_{t+1}(i)}. \tag{2.32}$$

It can be seen that the connection between $\gamma_t(j)$ and $\varepsilon_t(j,i)$ is

$$\gamma_t(j) = \sum_{i \in S(X)} \varepsilon_t(j,i). \tag{2.33}$$

Using the variables defined above, the reestimation formulas for A, B and π are

$$\hat{\pi}_j = \gamma_1(j) = P(X_1 = j \mid Y_1, ..., Y_T), \tag{2.34}$$

$$\hat{a}_{ji} = \frac{\sum_{t=1}^{T-1} \varepsilon_t(j,i)}{\sum_{t=1}^{T-1} \gamma_t(j)} = \frac{\sum_{t=1}^{T-1} P(X_t = j, X_{t+1} = i \mid Y_1, ..., Y_T)}{\sum_{t=1}^{T-1} P(X_t = j \mid Y_1, ..., Y_T)}, \tag{2.35}$$

and

$$\hat{b}_k^j = \frac{\sum_{t=1:\, Y_t=k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} = \frac{\sum_{t=1:\, Y_t=k}^{T} P(X_t = j \mid Y_1,...,Y_T)}{\sum_{t=1}^{T} P(X_t = j \mid Y_1,...,Y_T)}. \tag{2.36}$$

Note that the above formulas are reestimation formulas due to the iterative nature of the Baum-Welch algorithm. Therefore, these formulas will be applied repeatedly until the likelihood of the observed sequence reaches a maximum.

With the model estimated, the third question, which is to search for an optimal sequence of hidden state $X_1,...X_T$ that maximizes the probability,

$$P(X_1,...X_T \mid Y_1,...,Y_T), \tag{2.37}$$

can be solved. A technique that has been proposed for solving this problem is the Viterbi algorithm (Viterbi, 1967, Forney, 1973). The Viterbi algorithm was first proposed by Viterbi (1967) as an error-correction scheme for noisy digital communication links. It is now widely applied in decoding the convolutional codes used in digital communication, such as CDMA, GSM and 802.11 wireless LANs. It is also commonly used in speech recognition to find the most likely sequence of text given the acoustic signal.

To illustrate the algorithm, the following variable is defined

$$\delta_t(j) = \underset{X_1,...,X_{t-1} \in S(X)}{Max} P(X_1,...,X_t = j, Y_1,...,Y_t) \tag{2.38}$$

for $2 \le t \le T$, and

$$\delta_1(j) = P(X_1 = j, Y_1) = P(X_1 = j)P(Y_1 \mid X_1 = j) = \pi_j b_{y_1}^j. \tag{2.39}$$

By induction, it can be shown

$$\delta_{t+1}(j) = \underset{X_1,...,X_t \in S(X)}{Max} P(X_1,...,X_{t+1} = j, Y_1,...,Y_{t+1})$$

$$= \underset{X_1,...,X_t \in S(X)}{Max} P(X_1,...,X_t, Y_1,...,Y_t)P(X_{t+1} = j \mid X_1,...,X_t, Y_1,...,Y_t)$$

$$P(Y_{t+1} \mid X_1,...,X_{t+1} = j, Y_1,...,Y_t)$$

$$= \underset{X_1,...,X_t \in S(X)}{Max} P(X_1,...,X_t, Y_1,...,Y_t) P(X_{t+1} = j \mid X_t) P(Y_{t+1} \mid X_{t+1} = j)$$

$$= \underset{i \in S(X)}{Max} \underset{X_1,...,X_{t-1} \in S(X)}{Max} P(X_1,...,X_t = i, Y_1,...,Y_t) P(X_{t+1} = j \mid X_t = i) P(Y_{t+1} \mid X_{t+1} = j)$$

$$= \underset{i \in S(X)}{Max} \left(\delta_t(i) a_{ij} \right) b^j_{y_{t+1}}, \tag{2.40}$$

for $1 \le t \le T-1$. As it is necessary to keep track of the j that maximizes the Eq (2.40), we define the variable

$$\psi_{t+1}(j) = \arg\max_{i \in S(X)} \left(\delta_t(i) a_{ij} \right) \tag{2.41}$$

for $1 \le t \le T-1$, and $\psi_1(j) = 0$. Therefore, the hidden state for the last position of the sequence is estimated as

$$\hat{X}_T = \arg\max_j \left[\delta_T(j) \right]. \tag{2.42}$$

The rest of the optimal sequence of the hidden states can be estimated by path-backtracking as

$$\hat{X}_t = \psi_{t+1}(X_{t+1}), \tag{2.43}$$

for $1 \le t \le T-1$. Note that Eq (2.40) is maximized over all the previous hidden states. A slightly different approach would be to maximize Eq (2.40) over a fixed number of previous hidden states, which could reduce the computational complexity and is reasonable for most of the applications (Viterbi, 1967, Forney, 1973).

Before the widespread understanding and application of HMM, studies of the DNA compositional properties depended on some basic methods in its early stage of development, such as calculating the base composition and analyzing the base frequency of nearest neighbor. Elton (1974) showed that models with homogeneous structure, such as Markov chain models with stationary probabilities, could not provide enough description of the variation of base composition in DNA

sequences. Staden (1984) suggested a method to determine the heterogeneity by scanning a sequence with a fixed-length window and computing the statistics of local base composition. Some other methods also involved dividing the sequence into segments, and then testing the local composition using Chi-square statistics.

As one of the first applications in computational biology, Churchill (1989) proposed using HMM for modeling the heterogeneous structure of DNA sequences. The model assumes that the structure of DNA sequences consists of segments that have homogeneous base composition, but the compositional structure may vary from segment to segment. Under the HMM, each segment is represented by a hidden state and each hidden state represents an underlying homogeneous composition with a Markov chain model. A smoothing algorithm that can be used to reconstruct the sequence of hidden state was also introduced in this report. Some further applications of HMM on analysis of genome structure can be seen in Churchill (1992). Similarly, Muri (1998) used HMM to model bacterial genomes and identified the homogeneous regions in DNA sequences. In this report, the author applied a Markov Chain Monte Carlo (MCMC) method based on a stochastic EM algorithm. Compared to the Baum-Welch algorithm, the MCMC alternative is less sensitive to a poor choice of the starting point and is faster for each iteration, but requires more iterations to reach convergence.

Besides statistical modeling, the application of HMM has also been seen in gene finding (Krogh *et al*., 1994, Burge and Karlin, 1997, Kulp *et al.* (1996)). Generally, the gene finding application of HMM is based on its capability of dividing a sequence into homogeneous segments. The gene is identified by comparing the feature of each of the homogeneous segments to a prototype of interest. Kulp *et al.* (1996) introduce a generalized HMM for recognition of genes, which extended the standard HMM by considering a semi-Markov process for the hidden chain. Under the semi-Markov process, the transition probabilities of hidden states are not constant throughout the sequence but change along the

sequence following a probability distribution. Based on the hidden semi-Markov model, Burge and Karlin (1997) developed the gene finding software GENSCAN, which adopted a geometric distribution for the duration of the transition probabilities. Similar application of HMM can also be found on searching horizontal gene transfers (Bize *et al.* 1999), where HMM is used to find potentially transferred genes by identifying genes with similar structure between species.

Applications of HMM can also been seen in linkage analysis, sequence alignment, and so on. Generally, HMM provides a flexible mechanism that features a complex sequence built of short and simple segments, within which the sequence follows the same distribution, and between which the sequence has different distributions. Besides, these segments are not independent of each other. Their appearance follows a transition probability with a Markovian structure. A generalized case of HMM is Markov models in random environments (Cogburn, 1984), where the appearance of hidden states could be independent of each other or follow a decision rule.

2.2.2 Double Chain Markov Model (HMM) and its High-Order Extensions

As introduced in Section 1.1.2, an HMM is characterized by two sequences: the hidden state sequence $X = (X_1, ..., X_T)$ and the observed sequence $Y = (Y_1, ..., Y_T)$. On the classical HMM, the hidden states $X_i's$ are generated according to a 1st-order Markov model with transition probability $A = \{a_{ij}\}$. Under the hidden state X_i, the observations are generated independently with the probability $B = \{b_k^j\}$, where

$$b_k^j = P(Y_t = k \mid X_t = j),\ k \in S(Y), j \in S(X) \quad . \tag{2.44}$$

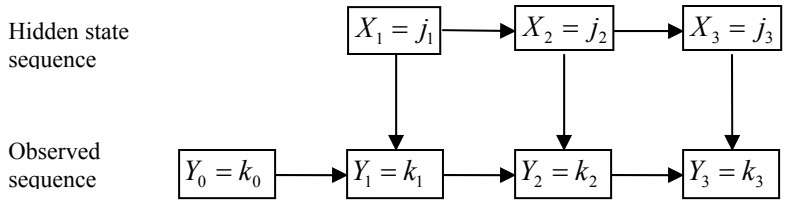
Thus, the classical HMM can be denoted as $M(1) - M(0)$, which indicates that the Markov orders of hidden sequence and observed sequence are 1 and 0, respectively. Muri (1998) and Bize (1999)

mentioned a more general form of HMM $M(1)-M(r)$, which assumes an rth-order Markov dependency between observations conditional to the hidden state with the probability $B=\{b_k^j\}$, where

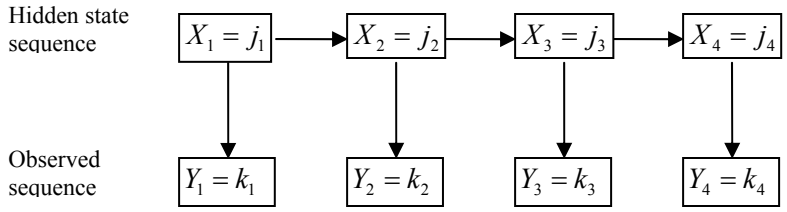
$$b^j_{k_{t-r},\ldots k_t} = P(Y_t = k_t \mid Y_{t-r} = k_{t-r},\ldots,Y_{t-1} = k_{t-1}, X_t = j), k \in S(Y), j \in S(X). \tag{2.45}$$

Wellekens (1987) showed a similar model with $r=1$ considering the case of continuous observations with applications on speech recognition. Paliwal (1993) also applied a similar HMM in the discrete case on speech recognition and provided the forward procedure for the new model. However, the three fundamental questions for this extended HMM were not systematically and comprehensively answered until Berchtold proposed the concept of DCMM in 1999.

DCMM is equivalent to model $M(1)-M(1)$ that assumes both hidden and observed sequences are 1st-order Markov chains. Therefore, as demonstrated in Fig. 2&3 in Berchtold (1999), the major difference between DCMM and the classical HMM is that DCMM assumes a Markov dependency of 1st-order between successive observations conditional to the hidden state (Fig. 2.1(a)), while HMM assumes the conditional independence between successive observations (Fig. 2.1(b)).



(a) DCMM



(b) HMM

Figure 2.1. Demonstration of (a) DCMM and (b) HMM.

To answer the three fundamental questions for DCMM, Berchtold (1999) provided a complete derivation of the adjusted forward-backward procedure, Baum-Welch and Viterbi algorithm based on the different assumption of DCMM. A high-order extension of DCMM was provided by Berchtold in a later report (Berchtold, 2002), where the DCMM was generalized as $M(l)-M(f)$ with both l and f non-negative integers. The high-order version of DCMM assumes that the hidden states are generated according to an *l*th-order Markov model and the observed sequence follows a Markov dependency of *f*th-order between successive observations conditional to the hidden state.

Since each observation depends on its previous f observations, the observed sequence needs f observations to initialize. For convenience purposes, the observed sequence is denoted as $Y_{-f+1},...,Y_T$ with $Y_{-f+1},...,Y_0$ for the initiation. To illustrate the high-order extension of DCMM, we borrow the notations and definitions pertaining to Berchtold (2002):

a) The sequence of hidden states is still denoted as $X_1,...,X_T$

b) The state space of hidden states: $S(X)=\{1,...,M\}$.

c) The state space of possible observation: $S(Y)=\{1,...,K\}$

d) The probability distribution of the first l hidden states given the previous states: $\pi=\{\pi_1=P(X_1),\pi_2=P(X_2\mid X_1),...,\pi_{l|1,...,l-1}=P(X_l\mid X_1,...,X_{l-1})\}$.

e) The *l*th-order transition probability between hidden states:

$A=\{a_{j_l,...,j_0}=P(X_t=j_0\mid X_{t-l}=j_l,...,X_{t-1}=j_1)\}$, $j_l,...,j_0\in S(X)$.

f) The *f*th-order transition probability between successive observations given a hidden state:

$B=\{b^j_{i_f,...,i_0}=P(Y_t=i_0\mid Y_{t-f}=i_f,...,Y_{t-1}=i_1,X_t=j)\}$, $j\in S(X)$, $i_f,...,i_0\in S(Y)$.

The DCMM is a generalized version of HMM. Therefore, the fundamental questions of HMM also apply for DCMM and they can be answered in a similar way. The forward-backward procedure and

Baum-Welch algorithm for HMM need to be adjusted according to the high-order Markov dependency existing in the DCMM. To answer the first question, the likelihood of the observed sequence given the DCMM $\mu = \{\pi, A, B\}$,

$$L = P(Y_{-f+1}, ..., Y_T \mid \mu) \tag{2.46}$$

must be computed.

The forward variable is

$$\alpha_t(j_{l-1}, ..., j_0) = P(Y_{-f+1}, ..., Y_t, X_{t-l+1} = j_{l-1}, ..., X_t = j_0). \tag{2.47}$$

For $t = 1$, Eq. (2.47) becomes

$$\alpha_1(j_0) = P(Y_{-f+1}, ..., Y_1, X_1 = j_0)$$

$$= b^{j_0}_{y_{-f+1}, ..., y_1} \pi_1(j_0) \tag{2.48}$$

Since $Y_{-f+1}, ..., Y_0$ are independent of $X_1 = j_0$ and $Y_{-f+1}, ..., Y_0$ are assumed to be known, $P(Y_{-f+1}, ..., Y_0, X_1 = j_0)$ can be written as $P(X_1 = j_0)$.

For $t = 2$, it is

$$\alpha_2(j_1, j_0) = P(Y_{-f+1}, ..., Y_2, X_1 = j_1, X_2 = j_0)$$

$$= b^{j_0}_{y_{-f+2}, ..., y_2} \pi_{2|1}(j_1, j_0) \alpha_1(j_1). \tag{2.49}$$

For $t = 3, ..., l$,

$$\alpha_t(j_{t-1}, ..., j_0) = P(Y_{-f+1}, ..., Y_t, X_1 = j_{t-1}, ..., X_t = j_0)$$

$$= b^{j_0}_{y_{-f+t}, ..., y_t} \pi_{t|1,...,t-1}(j_{t-1,...,} j_0) \alpha_{t-1}(j_{t-1}, ..., j_1) \tag{2.50}$$

For $t = l+1, ..., T$,

$$\alpha_t(j_{l-1}, ..., j_0) = P(Y_{-f+1}, ..., Y_t, X_{t-l+1} = j_{l-1}, ..., X_t = j_0)$$

$$= b^{j_0}_{y_{-f+t},...,y_t} \sum_{j_l \in S(X)} a_{j_l,...,j_0} \alpha_{t-1}(j_l,...,j_1). \tag{2.51}$$

Based on the forward procedure, the likelihood of the complete observed sequence is

$$L = \sum_{j_{l-1},...,j_0 \in S(X)} \alpha_T(j_{l-1},...,j_0). \tag{2.52}$$

The backward variable is defined as

$$\beta_t(j_{l-1},...,j_0) = P(Y_{t+1},...,Y_T \mid Y_{t-f+1},...,Y_t, X_{t-l+1} = j_{l-1},...,X_t = j_0). \tag{2.53}$$

For $t = T$, Eq. (2.53) becomes

$$\beta_T(j_{l-1},...,j_0) = 1, \tag{2.54}$$

for $j_{l-1},...,j_0 \in S(X)$. For $t = T-1,...,l$, it becomes

$$\begin{aligned}\beta_t(j_{l-1},...,j_0) &= P(Y_{t+1},...,Y_T \mid Y_{t-f+1},...,Y_t, X_{t-l+1} = j_{l-1},...,X_t = j_0) \\ &= \sum_{j \in S(X)} a_{j_{l-1},...j_0,j} b^j_{y_{t-f+1},...,y_{t+1}} \beta_{t+1}(j_{l-2},...,j_0,j).\end{aligned} \tag{2.55}$$

For $t = 1,...,l-1$, it is

$$\begin{aligned}\beta_t(j_{t-1},...,j_0) &= P(Y_{t+1},...,Y_T \mid Y_{t-f+1},...,Y_t, X_1 = j_{t-1},...,X_t = j_0) \\ &= \sum_{j \in S(X)} \pi_{t+1|1,...,t}(j_{t-1},...j_0,j) b^j_{y_{t-f+1},...,y_{t+1}} \beta_{t+1}(j_{t-1},...,j_0,j).\end{aligned} \tag{2.56}$$

Based on both forward and backward procedures, the likelihood of the complete observed sequence can be written as:

$$L = \sum_{j_{l-1},...,j_0 \in S(X)} \alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0) \tag{2.57}$$

for $t = l+1,...,T$, and

$$L = \sum_{j_{t-1},...,j_0 \in S(X)} \alpha_t(j_{t-1},...,j_0)\beta_t(j_{t-1},...,j_0) \tag{2.58}$$

for $t = 1,...,l$. It can been shown that Eq. (2.52) is equivalent to Eq. (2.56) with $t = T$.

The second question is to estimate the DCMM $\mu = \{\pi, A, B\}$ given the observed sequence. The Baum-Welch algorithm was used to answer this question for HMM. For the DCMM, Berchtold (2002) provided a similar algorithm. To describe the algorithm, the joint probability of successive hidden states conditional to the observed sequence is defined as

$$\gamma_t(j_{l-1},...,j_0) = P(X_{t-l+1} = j_{l-1},...,X_t = j_0 \mid Y_{-f+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0)}{\sum_{j_{l-1},...,j_0 \in S(X)} \alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0)}$$

$$= \frac{\alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0)}{L} \tag{2.59}$$

for $t = l+1,...,T$, and

$$\gamma_t(j_{t-1},...,j_0) = P(X_1 = j_{t-1},...,X_t = j_0 \mid Y_{-f+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{t-1},...,j_0)\beta_t(j_{t-1},...,j_0)}{\sum_{j_{t-1},...,j_0 \in S(X)} \alpha_t(j_{t-1},...,j_0)\beta_t(j_{t-1},...,j_0)}$$

$$= \frac{\alpha_t(j_{t-1},...,j_0)\beta_t(j_{t-1},...,j_0)}{L} \tag{2.60}$$

for $t = 1,...,l$. Similarly, it is defined

$$\varepsilon_t(j_{l-1},...,j_0,j) = P(X_{t-l+1} = j_{l-1},...,X_t = j_0, X_{t+1} = j \mid Y_{-f+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{l-1},...,j_0)a_{j_{l-1},...,j_0,j}b^j_{y_{t-f+1},...,y_{t+1}}\beta_{t+1}(j_{l-2},...,j_0,j)}{L} \tag{2.61}$$

for $t = l,...,T-1$, and

$$\varepsilon_t(j_{t-1},...,j_0,j) = P(X_1 = j_{t-1},...,X_t = j_0, X_{t+1} = j \mid Y_{-f+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{t-1},...,j_0)\pi_{t+1|1,...,t}(j_{t-1},...,j_0,j)b^j_{y_{t-f+1},...,y_{t+1}}\beta_{t+1}(j_{t-1},...,j_0,j)}{L} \tag{2.62}$$

for $t = 1,...,l-1$. The relations between γ and ε are

$$\gamma_t(j_{l-1},...,j_0) = \sum_{j \in S(X)} \varepsilon_t(j_{l-1},...,j_0,j) \tag{2.63}$$

for $t = l,...,T-1$, and

$$\gamma_t(j_{t-1},...,j_0) = \sum_{j \in S(X)} \varepsilon_t(j_{t-1},...,j_0,j) \tag{2.64}$$

for $t = 1,...,l-1$. Using the variables defined above, the reestimation formulas for π are

$$\hat{\pi}_1(j_0) = P(X_1 = j_0 \mid Y_{-f+1},...,Y_T) \tag{2.65}$$

for $t = 1$, and

$$\hat{\pi}_{t|1,...,t-1}(j_{t-1},...,j_0) = \frac{\gamma_t(j_{t-1},...,j_0)}{\gamma_{t-1}(j_{t-1},...,j_1)} \tag{2.66}$$

for $t = 2,...,l$. The reestimation formulas for the transition probabilities between hidden states and between successive observations are

$$\hat{a}_{j_{l-1},...,j_0,j} = \frac{\sum_{t=l}^{T-1} \varepsilon_t(j_{l-1},...,j_0,j)}{\sum_{t=l}^{T-1} \gamma_t(j_{l-1},...,j_0)} \tag{2.67}$$

and

$$\hat{b}^{j_0}_{i_f,...,i_0} = \frac{\sum_{t=1:\, Y_{t-f}=i_f,...,Y_t=i_0}^{T} \sum_{j_{l-1} \in S(X)} ... \sum_{j_1 \in S(X)} \gamma_t(j_{l-1},...,j_0)}{\sum_{t=1:\, Y_{t-f}=i_f,...,Y_{t-1}=i_1}^{T} \sum_{j_{l-1} \in S(X)} ... \sum_{j_1 \in S(X)} \gamma_t(j_{l-1},...,j_0)}. \tag{2.68}$$

The estimation process needs to start with a set of starting values for π, A and B, which would be used to compute the forward-backward valuables as well as variables γ and ε. Then based on Eq. (2.64) – Eq. (2.67), a new set of values for π, A and B can be calculated. The model is estimated by repeating the process until the likelihood of the observed sequence converges.

The DCMM adopted a slightly different Viterbi algorithm to find the optimal sequence of hidden states. To illustrate the algorithm, the following variables are defined

$$\delta_1(j_0) = P(Y_{-f+1},...,Y_1, X_1 = j_0) = \pi_1(j_0) b^{j_0}_{y_{-f+1},...,y_1} \tag{2.69}$$

for $t = 1$,

$$\begin{aligned}\delta_2(j_1, j_0) &= P(Y_{-f+1},...,Y_1, X_1 = j_1, X_2 = j_0) \\ &= \delta_1(j_1)\pi_{2|1}(j_1, j_0) b^{j_0}_{y_{-f+2},...,y_2}\end{aligned} \tag{2.70}$$

for $t = 2$,

$$\begin{aligned}\delta_t(j_{t-1},..., j_0) &= P(Y_{-f+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0) \\ &= \delta_{t-1}(j_{t-1},..., j_1)\pi_{t|1,...,t-1}(j_{t-1},..., j_0) b^{j_0}_{y_{t-f},...,y_t}\end{aligned} \tag{2.71}$$

for $t = 3,..., l$, and

$$\begin{aligned}\delta_t(j_l,..., j_0) &= P(Y_{-f+1},...,Y_t, X_{t-l} = j_l,..., X_t = j_0) \\ &= \delta_{t-1}(j_l,..., j_1) a_{j_l,...,j_0} b^{j_0}_{y_{t-f},...,y_t}\end{aligned} \tag{2.72}$$

for $t = l+1,..., T$.

For each δ variable, a vector is formed to maintain all possible values for later backtracking. For $\delta_1(j_0)$ and $\delta_2(j_1, j_0)$, the vectors are

$$\delta_1 = [\delta_1(1),..., \delta_1(M)] \tag{2.73}$$

and

$$\delta_2 = [\delta_2(1,1), \delta_2(1,2),..., \delta_2(1,M),..., \delta_2(M,1), \delta_2(M,2),..., \delta_2(M,M)]. \tag{2.74}$$

Similarly, the vector can be defined for $t = 3,..., T$. Note that the dimension changes for different *t* values. Therefore, the hidden state for the last position of the sequence is estimated as

$$\hat{X}_T = floor\left(\frac{\arg\max(\delta_T) - 1}{M^l}\right) + 1 . \tag{2.75}$$

The rest of the optimal sequence of hidden state can be estimated by path backtracking as

$$\hat{X}_t = floor\left(\frac{\arg\max(\delta_t a_{j_{l-1},\ldots,j_0,\hat{X}_{t+1}}) - 1}{M^{l-1}}\right) + 1 \tag{2.76}$$

for $t = T-1,\ldots,l$, and

$$\hat{X}_t = floor\left(\frac{\arg\max(\delta_t \pi_{t+1|1,\ldots,t}(j_{t-1},\ldots,j_0,\hat{X}_{t+1})) - 1}{M^{t-1}}\right) + 1 \tag{2.77}$$

for $t = l-1,\ldots,1$.

To examine the DCMM's capability of modeling non-homogeneous sequences, Berchtold (1999 & 2002) applied the model on several experimental data sets including a sequence of wind speeds, DNA sequence, a song of the Wood Pewee and behavior of young monkeys. The results showed not only that DCMM successfully represented non-homogeneous sequences, but it outperformed both the Markov models and HMMs.

Both HMM and DCMM have limitations on modeling non-homogeneous sequences. The HMM assumes conditional independence and ignores the possible dependence between successive observations, and the DCMM is the complete opposite of HMM. Therefore, it is desirable to have a model that takes into account the main features of both the HMM and DCMM.

3. Sequence Comparison Using Multi-Order Markov Chains

3.1 Markov Chain Model for DNA Sequences

It is natural to model DNA sequences with Markov chain models since dependencies are expected between nucleotides within DNA sequences. A Markov chain with stationary transition probabilities would provide a good description of a DNA sequence when the dependencies are homogeneous throughout the sequence. However, it is often observed that different patterns of dependencies and base composition exist in different segments of a DNA sequence. Churchill (1989) pointed out that the composition of naturally occurring DNA sequences is often strikingly heterogeneous. Therefore, the information extracted from a DNA sequence by a single order Markov model is not as representative as expected in most cases. In this study, we propose to represent a sequence using multi-order transition matrix (MTM) based on multi-order Markov chains. The MTM contains information of multi-order dependencies and gives a more comprehensive representation of the heterogeneous composition within a DNA sequence. Using the proposed MTM, a similarity measure can be developed for pair-wise comparison of sequences.

A *k*th-order homogeneous Markov chain $\{X_N\}_{N=0}^{\infty}$ with the state space $\mathbb{Z} = \{A, C, G, T\}$ can be summarized in a transition matrix as:

$$C_k = [p_{j_0,\ldots,j_k}], \tag{3.1}$$

where $j_0,\ldots, j_k \in \mathbb{Z}$ and

$$p_{j_0,\ldots,j_k} = P(X_t = j_k \mid X_{t-k} = j_0,\ldots, X_{t-1} = j_{k-1}). \tag{3.2}$$

It suggests that the value taken by X_t is decided by the values taken by $X_{t-k},\ldots, X_{t-1}$. Using the *k*th-order Markov chain as a model, the likelihood function of a DNA sequence $y_0 \ldots y_n$ is given as:

$$L(y_0 \ldots y_n \mid C_k) = \pi(y_0 \ldots y_{k-1}) \prod_{j_0,\ldots,j_k \in S} (p_{j_0,\ldots,j_k})^{n_{j_0 \ldots j_k}}, \tag{3.3}$$

where $\pi(y_0 ... y_{k-1})$ is the probability of observing $y_0 ... y_{k-1}$ for the first *k* positions in the sequence and $n_{j_0 \cdots j_k}$ is the observed frequency of the word $j_0 \ldots j_k$. By maximizing the likelihood, it can be shown that the maximum likelihood estimate (MLE) of $p_{j_0,\ldots,j_k}$ is:

$$\hat{p}_{j_0,\ldots,j_k} = \frac{n_{j_0 \cdots j_{k-1} j_k}}{n_{j_0 \cdots j_{k-1}+}}, \tag{3.4}$$

where

$$n_{j_0 \cdots j_{k-1}+} = \sum_{j_k \in S} n_{j_0 \cdots j_{k-1} j_k} \ . \tag{3.5}$$

The MLE of C_k for this DNA sequence is given as:

$$\hat{C}_k = [\hat{p}_{j_0,\ldots,j_k}]. \tag{3.6}$$

3.2 Multi-Order Transition Matrix (MTM)

Specifically, $\hat{C}_k$ only contains information about *k*th-order dependencies in a DNA sequence. Therefore, a combination of $\hat{C}_k$ with multiple orders would provide a more comprehensive description of dependencies existing in the DNA sequence. For a data set containing *m* sequences, we define the MTM for sequence *i* in the data set as $M(i)_K = \left\{ {}_c\hat{C}_k^i \right\}_{k \in K}$, where $i = 1, ..., m$, *K* is the set of values of *k* of interest and the pre-subscript *c* stands for concatenating $\hat{C}_k^i$'*s* as a column of matrices. For instance, $M(i)_K$ with $K = \{0,1\}$ is given as:

$$M(i)_K = \begin{bmatrix} \hat{C}_0^i \\ \hat{C}_1^i \end{bmatrix} = \begin{bmatrix} \hat{p}_A^i & \hat{p}_C^i & \hat{p}_G^i & \hat{p}_T^i \\ \hat{p}_{AA}^i & \hat{p}_{AC}^i & \hat{p}_{AG}^i & \hat{p}_{AT}^i \\ \hat{p}_{CA}^i & \hat{p}_{CC}^i & \hat{p}_{CG}^i & \hat{p}_{CT}^i \\ \hat{p}_{GA}^i & \hat{p}_{GC}^i & \hat{p}_{GG}^i & \hat{p}_{GT}^i \\ \hat{p}_{TA}^i & \hat{p}_{TC}^i & \hat{p}_{TG}^i & \hat{p}_{TT}^i \end{bmatrix}, \tag{3.7}$$

with $\hat{C}_0^i = \begin{bmatrix} \hat{p}_A^i & \hat{p}_C^i & \hat{p}_G^i & \hat{p}_T^i \end{bmatrix}$ and

$$\hat{C}_1^i = \begin{bmatrix} \hat{p}_{AA}^i & \hat{p}_{AC}^i & \hat{p}_{AG}^i & \hat{p}_{AT}^i \\ \hat{p}_{CA}^i & \hat{p}_{CC}^i & \hat{p}_{CG}^i & \hat{p}_{CT}^i \\ \hat{p}_{GA}^i & \hat{p}_{GC}^i & \hat{p}_{GG}^i & \hat{p}_{GT}^i \\ \hat{p}_{TA}^i & \hat{p}_{TC}^i & \hat{p}_{TG}^i & \hat{p}_{TT}^i \end{bmatrix}. \tag{3.8}$$

3.3 Order Selection

Wu *et al.* (2007) proposed a scoring scheme to independently evaluate information content associated with each word by assigning a score. A higher score indicates that more information is contained in the word. We apply the same scheme to assign a score to each transition probability within the MTM. Then by examining the average scores by orders, we can decide which group of orders should be considered more informative than others and included in set *K*.

Let $G = \{0,1,...,l\}$, where *l* is the largest order to be considered for the data set. We use the following steps to select a subset *K* of *G*. First of all, we calculate the sequence *i*'s MTM $M(i)_G$ for $i = 1,...,m$. Then we concatenate all *m* sequences into a single sequence *S* and calculate the MTM $M(S)_G$ for sequence *S*. The score for a transition probability $p_{j_0,...,j_k}$ is defined as:

$$s(p_{j_0,...,j_k}) = \left| \sum_{i=1}^{m} \hat{p}^i_{j_0,...,j_k} \ln\left(\frac{\hat{p}^i_{j_0,...,j_k}}{\hat{p}^S_{j_0,...,j_k}}\right) \right|, \tag{3.9}$$

which is the Kullback-Leibler distance between $\hat{p}^i_{j_0,...,j_k}$'*s* and $\hat{p}^S_{j_0,...,j_k}$. A large value of $s(p_{j_0,...,j_k})$ indicates that $\hat{p}^i_{j_0,...,j_k}$'*s* are diversified among all the sequences. Thus, this particular transition probability is more informative than others. Since each transition probability is associated with a particular order, we average the non-zero $s(p_{j_0,...,j_k})$'s by their associated orders. The set *K* should

include the orders that have the highest average scores. Applications of the order selection are shown in the Results and Discussions section.

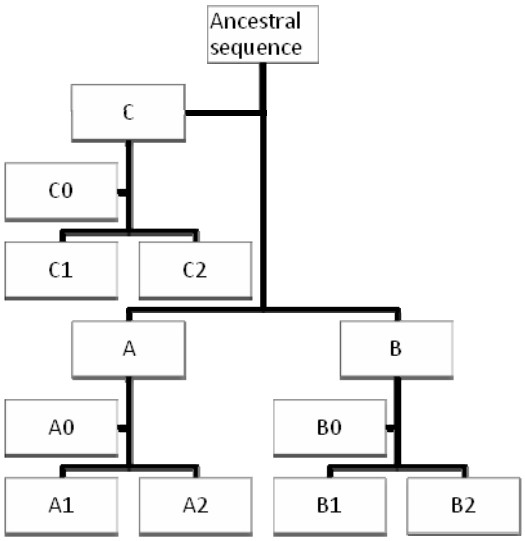


Figure 3.1 The pre-defined relationship used to generate simulated sequences.

3.4 Distance Measure

Let $M(S_1)_K = [\alpha_{ij}]$ and $M(S_2)_K = [\beta_{ij}]$ be the MTM of two DNA sequences S_1 and S_2, respectively. To calculate the distance between S_1 and S_2, denoted as $D(S_1, S_2)$, we adopt the distance measure as detailed below:

$$D(S_1, S_2) = \frac{1 - C(S_1, S_2)}{2}, \tag{3.10}$$

where $C(S_1, S_2) = \dfrac{\sum_{i,j} \alpha_{ij} \times \beta_{ij}}{(\sum_{i,j} \alpha_{ij}^2 \times \sum_{i,j} \beta_{ij}^2)^{1/2}}$.

The row dimension of the MTM can be very large as the number of order of interest in K increases. However, it is not necessary to use all the elements in the MTM to calculate the pairwise distance. Each element in the MTM represents a transition probability associated with a word. In our application, we only compared the transition probabilities of words that are shared by both sequences.

The majority of words that are shared by DNA sequences are relatively short. Observing a particular long word w in sequence S_1 but not in sequence S_2 is generally a result of random process instead of a selection process, because the probability that this word occurs in sequence S_2 is considerably small. For example, if w is a word of length 15, then the chance of finding w in a DNA sequence with 2000 nucleotides is about 1 out of 500,000. Therefore, including this transition probability of w while comparing sequences S_1 and S_2 could increase the random noise contained in the distance measure.

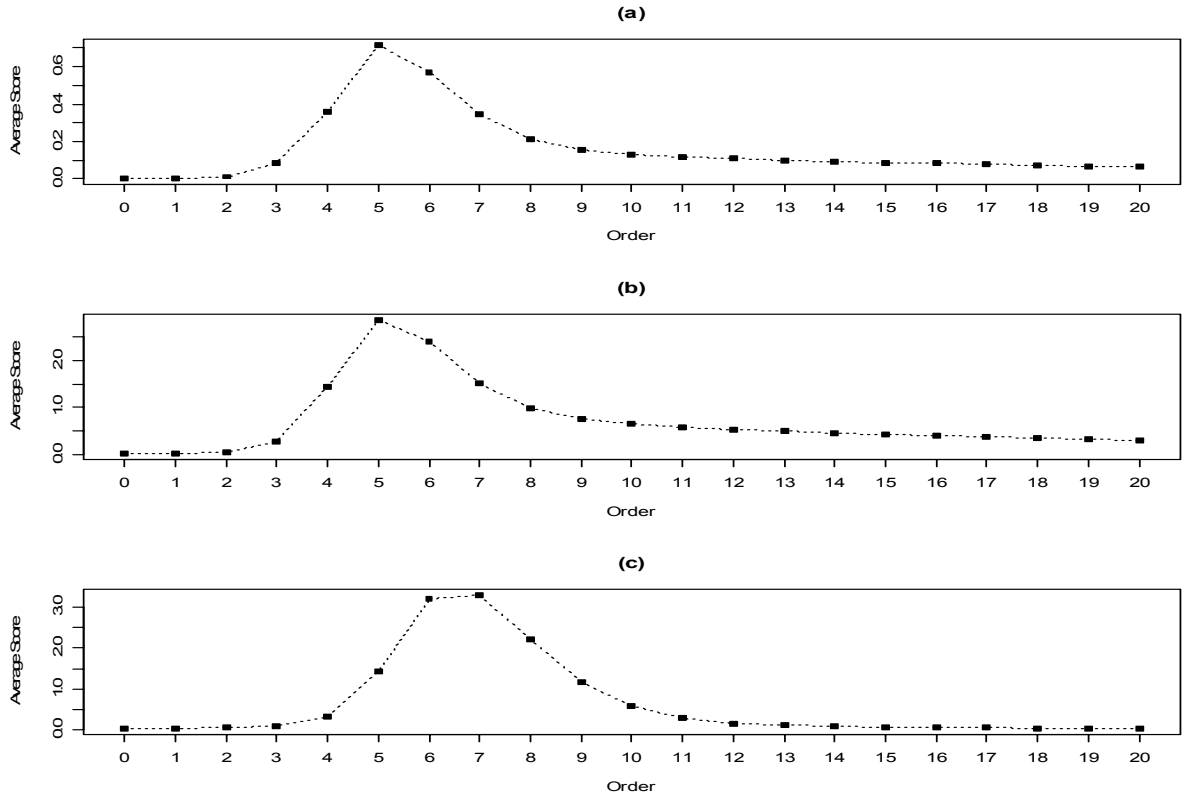


Figure 3.2. Average scores of a transition probability as a function of the order of the MTM experimented using (a) simulated dataset, (b) influenza A viral hemagglutinin gene sequences, and (c) the compete mitochondrial genome sequences of eutherian and non-eutherian organisms.

3.5 Data Sets

To compare the performance of the MTM with the ML and the CCV methods, we adopted a similar approach as in Otu and Sayood (2003) to simulate sequences. In brief, we started with an ancestral sequence randomly picked from our influenza virus database. We then evolved it into three

main sequences A, B, and C using different types of mutations (insertion, deletion, substitution, inversion, transposition and translocation) at a rate of 5-10%. A and B were evolved in a similar way to preserve resemblance to each other. Each of the three main sequences has three offspring sequences 0, 1 and 2. Sequence 1 and 2 were similarly generated using all six types of mutations at the rate of 5-20%, and sequence 0 was generated using only three types of mutations (insertion, deletion, substitution) at the rate of 5-20%. The higher mutation rates were applied in the last step to increase the divergence of within-group sequences. Fig. 3.1 shows the pre-defined evolutionary relationship for the above simulation. The nine derived sequences were used for analysis. A total of 1000 data sets were generated for the estimation of phylogeny support.

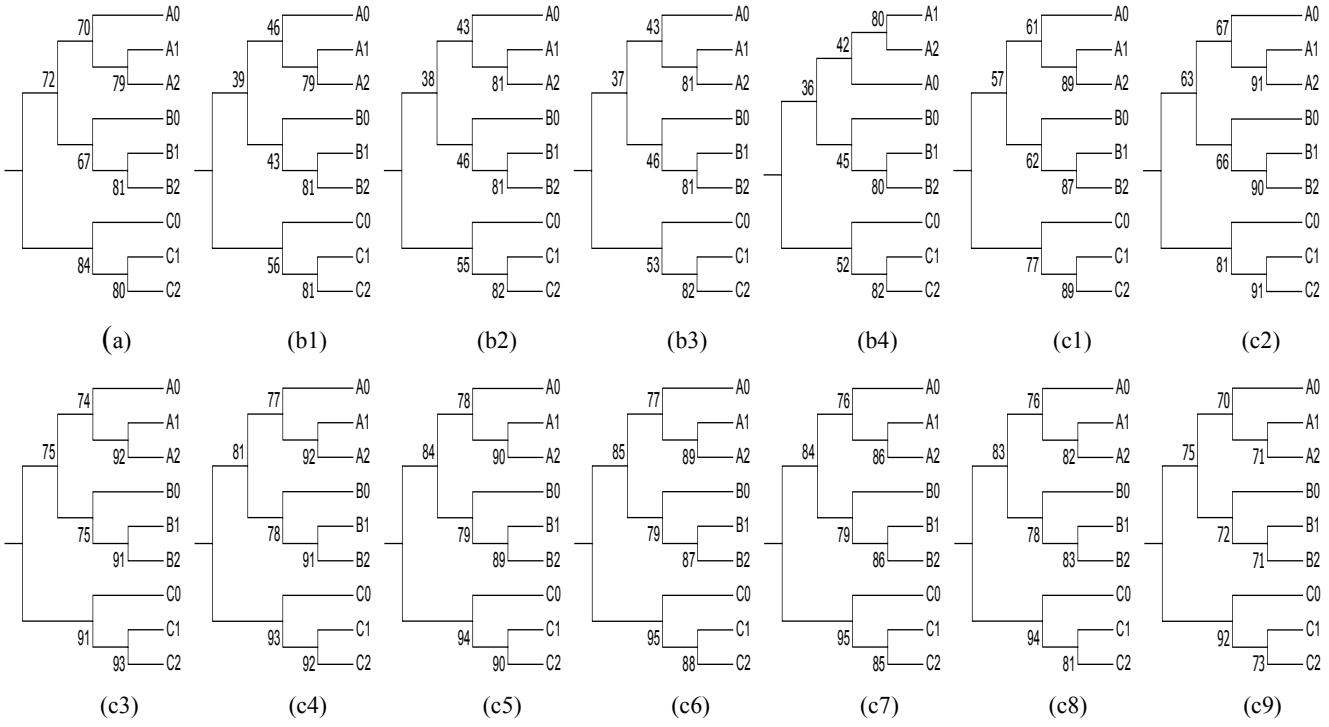


Figure 3.3 Consensus trees obtained from the 1000 simulated sequences using (a) Maximum likelihood, (b1)-(b3) CCV method with selected strings of size 500, 2500 and 5000, (b4) CCV method, (c1)-(c8) MTM method with K including the selected 1, 3, 5, 6, 7, 8, 9, and 11 orders with highest average score, and (c9) MTM method with $K = \{0,1,...,20\}$.

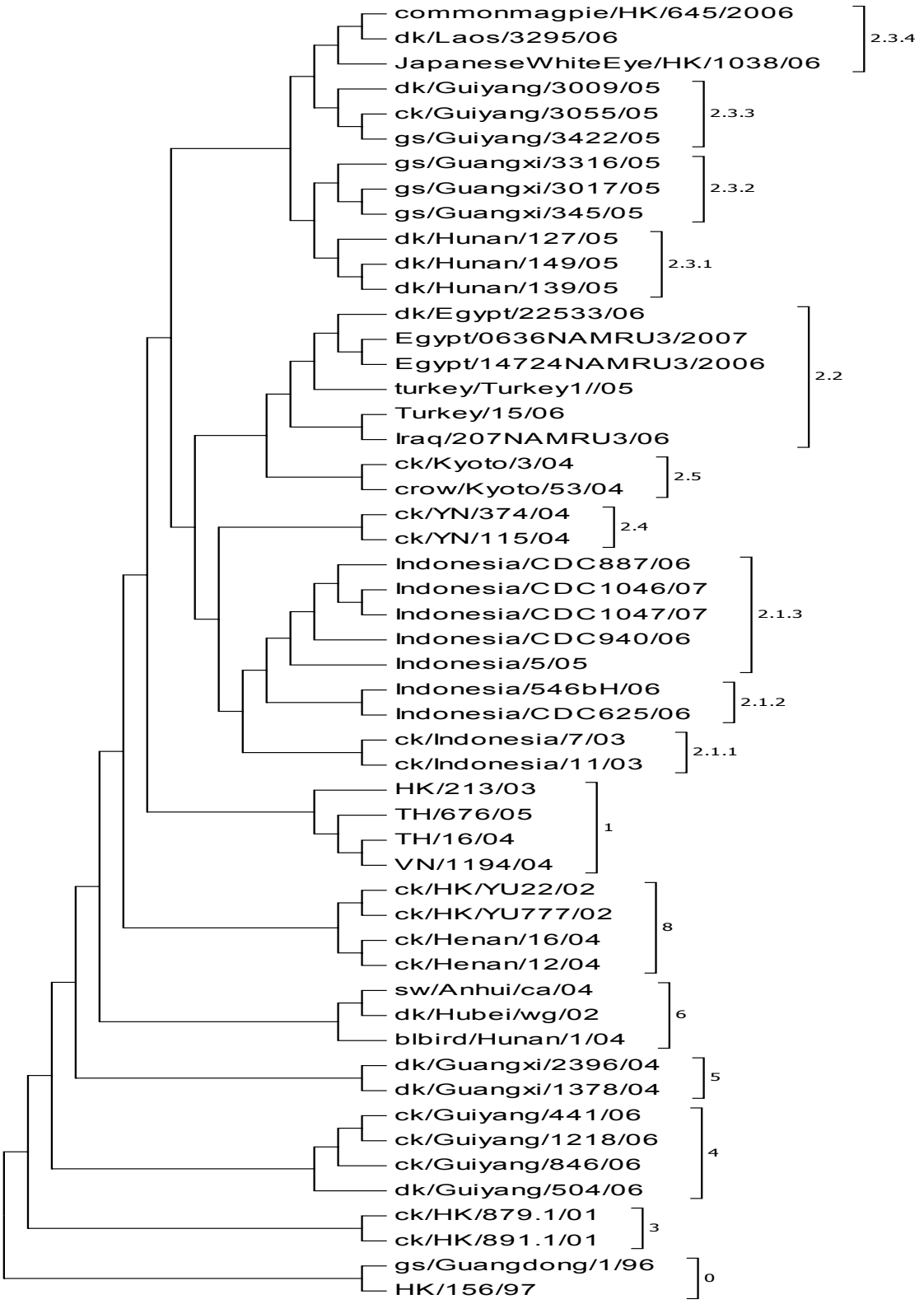


Figure 3.4 Lineage analysis of influenza A viral hemagglutinin gene sequences using the MTM method. Numbers labeled for each group are adopted from the International H5N1 Evolution Working Group as the clade nomenclature system (WHO/OIE/FAO H5N1 Evolution Working Group, 2008).

In addition, we used MTM method to analyze two real experimental data sets and compared the resulting topologies with the published ones (WHO/OIE/FAO H5N1 Evolution Working Group, 2008, Cao *et al.*, 1997; Cao *et al.*, 1998; Li *et al.*, 2001; Otu and Sayood, 2003; Reyes *et al.*, 2000). The first data set from the influenza A viral HA gene sequences consists of 52 sequences, with the length around 1600bp. The second data set consists of 29 mitochondrial genomes of eutherians and noneutherians (the later used as outgroup) (Appendix I Table A.1 and A.2).

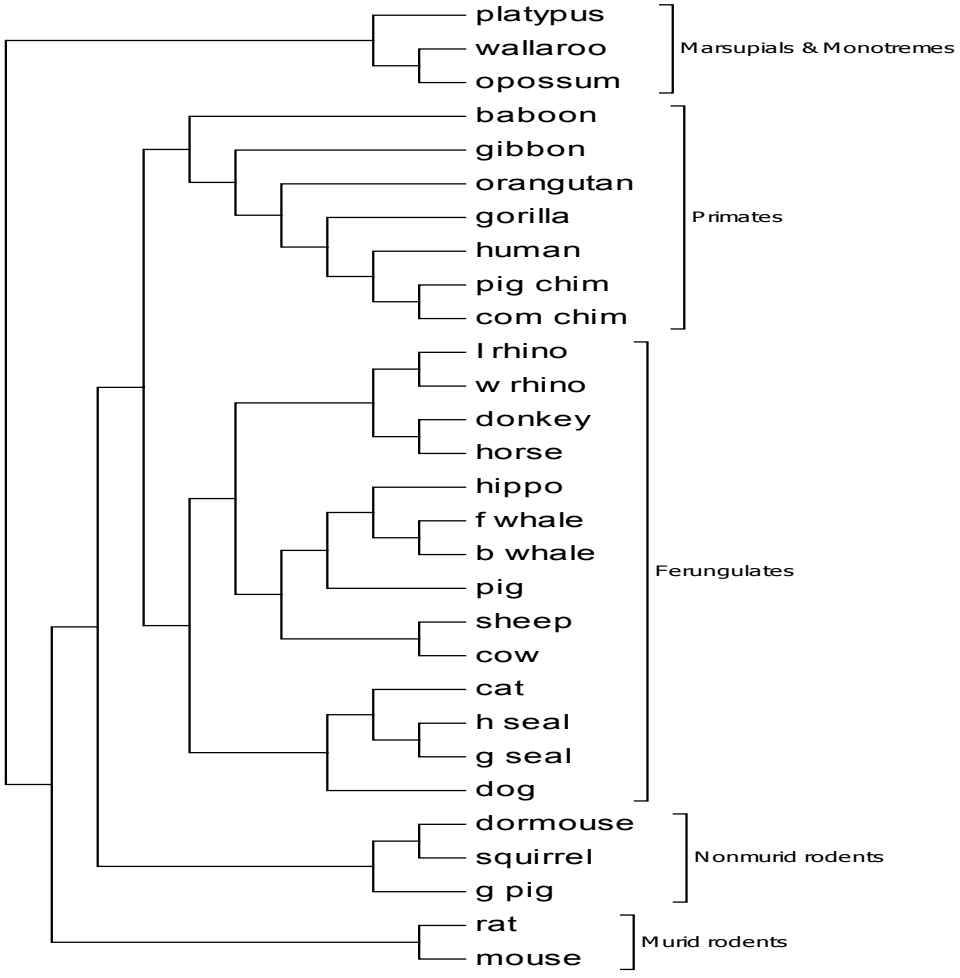


Figure 3.5 The phylogenetic trees built from the complete mitochondrial genome sequences using the MTM method with selected orders.

3.6 Analysis of the Simulated Data Sets

Before analyzing the simulated sequences, we examined the average scores by orders to determine the set of orders that should be included in set *K*. Fig. 3.2 (a) shows the average scores by orders for one simulated dataset. Clearly, order 4 to 7 are the orders that have the highest scores. To investigate the performance of the MTM method, we included different set of top scored orders in *K* as we analyzed the 1000 simulated data sets.

Fig. 3.3 shows that all the methods are able to recover the true topology of the simulated sequences. However, the supporting values on the clades of the consensus trees show that the performance fluctuates among the different methods. Fig. 3.3 (c1)-(c8) show the consensus trees of the MTM method with selected top 1, 3, 5, 6, 7, 8, 9 and 11 orders (i.e., K={5}, {4, 5, 6}, {4,…,8}, {4,…9}, {4,…10}, {4,…,11}, {4,…,12}, {4,…,14}, respectively). It is clear that the supporting values increase as *K* includes more orders, and then reach a limit and start to decline as more large orders are included in set *K*. This suggests that as the order increases, higher order Markov chain models do not provide a better fit of the sequences than lower order Markov models after a certain point. We also observe that the supporting values on sub-clades (A1, A2), (B1, B2) and (C1, C2) reach the limit faster than others. Fig. 3.3 (c9) is the consensus tree of the MTM method with all 21 orders, which further confirms the declination of the supporting values as more orders are included in set *K*.

Comparing the resulting trees of the MTM methods with the consensus tree of the ML method [Fig. 3.3 (a)], we can see that the MTM methods show overall better performance than the ML method when *K* contains 5, 6, 7, 8, 9 and 11 top orders. The rest of trees of the MTM methods show mixed results of higher and lower supporting values than the resulting tree of the ML method. Results are also shown for the CCV method and the CCV method with selected strings of size 500, 2500 and 5000 [Fig. 3.3 (b1)-(b4)]. The consensus trees give relatively low supporting values on the three main clades and

the relationship between clades A and B, compared to the trees from both the ML and MTM methods. We also notice that the supporting values of the consensus tree from the CCV method are not significantly different with those from the CCV method with selected strings in this case.

3.7 Lineage Analysis of Influenza A Virus

We examined the average score of each order from 0 to 20 for the influenza A viral HA sequences [Fig. 3.2 (b)]. In the orders from 4 to 7, the average scores are higher than the other orders. The average score peaks at the order 5.

According to the analysis result of the simulated dataset as described earlier, this experimental dataset was analyzed by selecting top 6, 7, 8, 9 and 10 orders (i.e., K={4,…,9}, {4,…,10}, {4,…,11}, {4,…,12} and {4,…,13}) respectively in the MTM method. No matter which of the above orders were applied, the resulting trees were nearly identical. The consensus tree is shown in Fig. 3.4. The clade numbers shown at the end of each group follow the clade nomenclature system originally designated by the International H5N1 Evolution Working Group. The consensus tree generated from the MTM method is consistent with those determined by the ML, maximum parsimony (MP) and Neighbor-joining algorithms (WHO/OIE/FAO H5N1 Evolution Working Group, 2008).

3.8 Phylogeny of Eutherian Orders

The phylogeny of eutherian orders remains unsolved (Novacek, 1992; Cao *et al.*, 1998), because conflicting topologies have been obtained using different protein sequences or using different methods. Several questions remain the focus of discussion: (1) the outgroup status of rodents, i.e. (rodents (ferungulates, primates)) or (ferungulates (primates, rodents)), (2) the issue of rodent monophyly versus rodent paraphyly/polyphyly, and (3) whether guinea pigs are rodents.

Similarly, the average score of each order was used to decide the number of orders that need to be included in the analysis of the mitochondrial genome data set (Appendix I Table 2). Fig. 3.2 (c) shows a slight different pattern from the first two data sets, where higher average scores are reached in the orders from 5 to 9. We thus used the top 6, 7, 8, 9 and 10 orders (i.e., K={5,…,10}, {4,…,10} and {4,…,11}, {4,…,12} and {4,…,13}), respectively in the MTM method. The corresponding five trees were found to be consistent with each other. The resulting consensus tree (Fig. 3.5) agrees with the overall structure of published trees (Cao *et al.*, 1997; Cao *et al.*, 1998; Li *et al.*, 2001; Otu and Sayood, 2003; Reyes *et al.*, 2000). With noneutherian mammals as outgroup, the tree suggests that murid rodents are the early branch of the tree (Reyes *et al*. 2000). It also shows that ferungulates are more closely related to primates, which support the (rodents (ferungulates, primates)) grouping (Cao *et al.*, 1997; Cao *et al.*, 1998; Li *et al.*, 2001; Otu and Sayood, 2003).

As for the phylogenetic position of rodents, our tree agrees with rodent paraphyly (Otu and Sayood, 2003; Reyes *et al.*, 2000). Guinea pig was classified as nonmurid rodents in our tree, which is contradicted with the observation by D'Erchia *et al.* (1996). However, the position of guinea pig remains unresolved because the results from various studies are contradictory.

As shown in Fig. 3.2, there exist a set of orders that are more informative than other orders in each data set examined. For the simulated sequence and the influenza A viral HA gene sequences, the orders with higher average scores are from 4 to 8. For the 29 mitochondrial genomes, the orders with higher average scores are from 5 to 10. This finding demonstrates the heterogeneous composition of the sequences, which is the motivation for the development of the MTM method. As for how many numbers should be included in computing the MTM, we recommend the most informative orders should be included. However, we discourage including too many orders in the selected set of orders (i.e., K). As shown in Fig. 3.3, when K reaches a turning point, the resulting consensus trees deteriorate as K expands.

3.9 Conclusion

In this study, we proposed a new alignment-free method for sequence comparison. Unlike most existing methods, the new method takes into consideration the compositional structure of DNA sequences, and represents the DNA sequences in a more comprehensive manner so that less information is lost when comparing sequences. An order selection method was introduced to determine the most informative orders included in calculating the MTM. The results have shown that the MTM method can be successfully applied to phylogeny inference using either the whole genomes or the genomic segments. Overall, our method shows a unique perspective on molecular sequence comparison. A natural extension of the research in the future would be to investigate the behavior of different distance metrics with the MTM, including the correlation-based and information theory-based metrics.

4. Multi-Order Markov Model under Hidden States (MMMHS)

The MTM method mentioned above utilizes the Markov models of different orders to represent a heterogeneous DNA sequence, and then combines the information from the different Markov models for a complete representation of the sequence. To a certain extent, this method points out a direction for a stochastic model for heterogeneous sequences. The model, which I named Multi-Order Markov model under Hidden States (MMMHS), assumes that different segments in a heterogeneous sequence follow different order Markov models that are associated with each other by an unobservable Markov chain. In terms of the hidden states or unobservable Markov chain, MMMHS is similar to the existing HMM and DCMM.

HMM consists of an unobservable Markov chain with a finite number of states and an observable random sequence. Each of the hidden states is associated with a probabilistic distribution. Under the assumption of HMM, each discrete position of the random sequence has an unknown state that determines the value at that point based on its corresponding probabilistic distribution. When modeling heterogeneous sequence using HMM, each locally homogeneous segment is represented as a short random sequence and classified into one of the finite hidden states. It is important to notice that the value taken at each position of the sequence is only decided by the hidden state associated with that position and has nothing to do with the values taken by the previous positions. In other words, the value taken at each position is independent of the values taken by the previous positions given the hidden state associated with that position.

However, within a real DNA sequence, dependencies among nucleotides are always expected. Thus, conditional independence in HMM is not a reasonable assumption while modeling DNA sequences. To resolve the limitation of HMM, Berchtold (1999 and 2002) introduced the Double Chain Markov model (DCMM), a generalization of HMM, which assumes that the observed sequence follows

a Markov model with a pre-selected order *f* instead of an independent model. Therefore, under DCMM, the value taken at each position depends on both the hidden state associated and values taken by the previous *f* positions.

Introduction of DCMM only solves part of the problem for DNA sequence modeling, because it is hard to imagine that each nucleotide in a DNA sequence is consistently related to the previous *f* nucleotides. A more reasonable thought would be that the value taken at a position in a sequence could be associated with any number of previous values or none of those values. Based on this idea, we suggest the MMMHS, which assumes a different dependence structure of the consecutive positions in a sequence under different hidden states. The MMMHS is an extension of the classical HMM since it still has the part of unobservable Markov chain. The characteristic separates MMMHS from HMM and DCMM is that the observable sequence is assumed be a combination of Multi-Order Markov chains. The model is formally defined in next section. For the convenience of comparing DCMM with MMMHS, we continue to use the notations and definitions in Berchtold (2002) for the illustration of DCMM.

4.1 Model

To describe the model, let us start with a random variable *Y* that takes value in the state space $S(Y) = \{1,...,K\}$ and a sequence of observations of *Y*. The model assumes that the hidden state *X* follows an *l*th Markov model with the state space $S(X) = \{0,1,...,M\}$. Each element in $S(X)$ is a possible value that hidden state *X* can take. It also decides the dependence structure in the observed *Y* sequence. For example, at the *t*th position of the observed sequence, if the hidden state $X_t = m$ ($m \in S(X)$), the value of Y_t is decided by both its hidden state X_t and the values taken by $Y_{t-m},...,Y_{t-1}$.

The sequence of observation of Y is defined as $Y_{-M+1}, Y_{-M+2}, ..., Y_0, Y_1, ..., Y_T$. Since Y depends on its past, the first M observations of the sequence $Y_{-M+1}, Y_{-M+2}, ..., Y_0$ are reserved to initiate the sequence for convenience purposes. The likelihood of the sequence is calculated based on the rest of the sequence $Y_1, ..., Y_T$. Fig. 4.1 shows the sequence of observations of Y and the corresponding hidden states at each of the positions. To completely describe the model, we need to define the following transition

Hidden States X :	X_1 , X_2 , ... , X_{T-1} , X_T
Observations of Y :	$Y_{-M+1}, ..., Y_0$, Y_1 , Y_2 , ... , Y_{T-1} , Y_T

Figure 4.1 Initiation of sequence $Y_1, ..., Y_T$ and corresponding hidden states sequence $X_1, ..., X_T$.

probabilities:

1). Probability distribution of the first l hidden states in the hidden state sequence given the previous hidden states,

$$\pi = \left\{ \pi_1 = P(X_1), \pi_{2|1} = P(X_2 \mid X_1), ..., \pi_{l|1,...,l-1} = P(X_l \mid X_1, ..., X_{l-1}) \right\}. \tag{4.11}$$

2). Transition matrix of order l between the hidden states,

$$A = \left\{ a_{j_l, ..., j_0} = P(X_t = j_0 \mid X_{t-l} = j_l, ..., X_{t-1} = j_1) \right\}, \tag{4.12}$$

where $j_l, ..., j_0 \in S(X)$.

3). Transition matrices between the successive observations of Y given a hidden state X,

$$B = \{ B^{j_0} \}, \tag{4.13}$$

with

$$B^{j_0} = \left\{ b^{j_0}_{i_{j_0}, ..., i_0} = P(Y_t = i_0 \mid Y_{t-j_0} = i_{j_0}, ..., Y_{t-1} = i_1, X_t = j_0) \right\}, \tag{4.14}$$

for $j_0 \geq 1$ and

$$B^{j_0} = \left\{ b^{j_0}_{i_0} = P(Y_t = i_0 \mid X_t = j_0) \right\}, \tag{4.15}$$

for $j_0 = 0$, where $j_0 \in S(X)$ and $i_{j_0}, ..., i_0 \in S(Y)$.

With above transition probabilities defined, a MMMHS μ is written as $\mu = \{\pi, A, B\}$. Similar to the HMM and DCMM, MMMHS has the three fundamental questions:

1) How to compute the probability of the sequence of observations of *Y*, given $\mu = \{\pi, A, B\}$?
2) How to adjust the model parameters so that the probability of the observed sequence can be maximized?
3) How to find an optimal hidden state sequence, given both $\mu = \{\pi, A, B\}$ and the observed sequence?

The next three sections will propose the algorithms to solve these questions.

4.2 Likelihood of the Observed Sequence

The likelihood of the observed sequence given the model $\mu = \{\pi, A, B\}$ can be written as

$$L = P(Y_{-M+1}, ..., Y_T \mid \mu). \tag{4.16}$$

The ideas behind the forward-backward procedures used for HMM and DCMM can be extended here for the MMMHS. Derivation of the following equations in the procedures is shown in Appendix II.

Define the forward variable as

$$\alpha_t(j_{l-1}, ..., j_0) = P(Y_{-M+1}, ..., Y_t, X_{t-l+1} = j_{l-1}, ..., X_t = j_0). \tag{4.17}$$

For $t = 1$, Eq. (4.17) is

$$\alpha_1(j_0) = P(Y_{-M+1}, ..., Y_1, X_1 = j_0)$$

$$= b^{j_0}_{y_{-j_0+1}, ..., y_1} \pi_1(j_0). \tag{4.18}$$

Since $Y_{-M+1}, ..., Y_0$ is defined as observed, $P(Y_{-M+1}, ..., Y_0, X_1 = j_0)$ can be written as $P(X_1 = j_0)$.

For $t = 2$, Eq. (4.17) is

$$\alpha_2(j_1, j_0) = P(Y_{-M+1}, ..., Y_2, X_1 = j_1, X_2 = j_0)$$

$$= b^{j_0}_{y_{-j_0+2}, ..., y_2} \pi_{2|1}(j_1, j_0) \alpha_1(j_1) . \tag{4.19}$$

For $t = 3, ..., l$,

$$\alpha_t(j_{t-1}, ..., j_0) = P(Y_{-M+1}, ..., Y_t, X_1 = j_{t-1}, ..., X_t = j_0)$$

$$= b^{j_0}_{y_{t-j_0}, ..., y_t} \pi_{t|1,...,t-1}(j_{t-1,...,} j_0) \alpha_{t-1}(j_{t-1}, ..., j_1) . \tag{4.20}$$

For $t = l+1, ..., T$,

$$\alpha_t(j_{l-1}, ..., j_0) = P(Y_{-M+1}, ..., Y_t, X_{t-l+1} = j_{l-1}, ..., X_t = j_0)$$

$$= b^{j_0}_{y_{t-j_0}, ..., y_t} \sum_{j_l \in S(X)} a_{j_l, ..., j_0} \alpha_{t-1}(j_l, ..., j_1) . \tag{4.21}$$

Based on the forward procedure, the likelihood of the complete observed sequence is given as

$$L = \sum_{j_{l-1}, ..., j_0 \in S(X)} \alpha_T(j_{l-1}, ..., j_0) . \tag{4.22}$$

The backward variable is defined as

$$\beta_t(j_{l-1}, ..., j_0) = P(Y_{t+1}, ..., Y_T \mid Y_{t-M+1}, ..., Y_t, X_{t-l+1} = j_{l-1}, ..., X_t = j_0) . \tag{4.23}$$

For $t = T$, Eq. (4.23) becomes

$$\beta_T(j_{l-1}, ..., j_0) = 1 , \tag{4.24}$$

for $j_{l-1}, ..., j_0 \in S(X)$. For $t = T-1, ..., l$, it is

$$\beta_t(j_{l-1}, ..., j_0) = P(Y_{t+1}, ..., Y_T \mid Y_{t-M+1}, ..., Y_t, X_{t-l+1} = j_{l-1}, ..., X_t = j_0)$$

$$= \sum_{j \in S(X)} a_{j_{l-1}, ... j_0, j} b^j_{y_{t-j+1}, ..., y_{t+1}} \beta_{t+1}(j_{l-2}, ..., j_0, j) . \tag{4.25}$$

For $t = 1, ..., l-1$, we have

$$\beta_t(j_{t-1}, ..., j_0) = P(Y_{t+1}, ..., Y_T \mid Y_{t-M+1}, ..., Y_t, X_1 = j_{t-1}, ..., X_t = j_0)$$

$$= \sum_{j \in S(X)} \pi_{t+1|1,...,t}(j_{t-1},...j_0,j) b^j_{y_{t-j+1},...,y_{t+1}} \beta_{t+1}(j_{t-1},...,j_0,j) . \quad (4.26)$$

It can be shown that

$$\alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0) = \begin{cases} P(Y_{-M+1},...,Y_T, X_{t-l+1} = j_{l-1},...,X_t = j_0), \text{for } t = l+1,...,T \\ P(Y_{-M+1},...,Y_T, X_1 = j_{t-1},...,X_t = j_0), \text{for } t = 1,...,l \end{cases} . \quad (4.27)$$

Based on both forward and backward procedures, the likelihood of the complete observed sequence can be written as:

$$L = \sum_{j_{l-1},...,j_0 \in S(X)} \alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0) \quad (4.28)$$

for $t = l+1,...,T$, and

$$L = \sum_{j_{t-1},...,j_0 \in S(X)} \alpha_t(j_{t-1},...,j_0)\beta_t(j_{t-1},...,j_0) \quad (4.29)$$

for $t = 1,...,l$. We can see that Eq. (4.22) is equivalent to Eq. (4.28) with $t = T$.

4.3 Estimation of Model Parameters π, *A* and *B*

Estimation of π, *A* and *B* can be done using an EM algorithm. To show the EM algorithm for MMMHS, we define the following joint probabilities of $l+1$ successive hidden states. For $t = l,...,T-1$,

$$\varepsilon_t(j_{l-1},...,j_0,j) = P(X_{t-l+1} = j_{l-1},...,X_t = j_0, X_{t+1} = j \mid Y_{-M+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{l-1},...,j_0) a_{j_{l-1},...,j_0,j} b^j_{y_{t-j+1},...,y_{t+1}} \beta_{t+1}(j_{l-2},...,j_0,j)}{L(Y_{-M+1},...,Y_T)} . \quad (4.30)$$

For $t = 1,...,l-1$,

$$\varepsilon_t(j_{t-1},...,j_0,j) = P(X_1 = j_{t-1},...,X_t = j_0, X_{t+1} = j \mid Y_{-M+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{t-1},...,j_0)\pi_{t+1|1,...,t}(j_{t-1},...,j_0,j) b^j_{y_{t-j+1},...,y_{t+1}} \beta_{t+1}(j_{t-1},...,j_0,j)}{L(Y_{-M+1},...,Y_T)} . \quad (4.31)$$

Define the joint probabilities of *l* successive hidden states as

$$\gamma_t(j_{l-1},...,j_0) = P(X_{t-l+1} = j_{l-1},...,X_t = j_0 \mid Y_{-M+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{l-1},...,j_0)\beta_t(j_{l-1},...,j_0)}{L(Y_{-M+1},...,Y_T)} \tag{4.32}$$

for $t = l+1,...,T$, and

$$\gamma_t(j_{t-1},...,j_0) = P(X_1 = j_{t-1},...,X_t = j_0 \mid Y_{-M+1},...,Y_T)$$

$$= \frac{\alpha_t(j_{t-1},...,j_0)\beta_t(j_{t-1},...,j_0)}{L(Y_{-M+1},...,Y_T)} \tag{4.33}$$

for $t = 1,...,l$. It can be shown that the relations between γ and ε are

$$\gamma_t(j_{l-1},...,j_0) = \sum_{j \in S(X)} \varepsilon_t(j_{l-1},...,j_0,j) \tag{4.34}$$

for $t = l,...,T-1$, and

$$\gamma_t(j_{t-1},...,j_0) = \sum_{j \in S(X)} \varepsilon_t(j_{t-1},...,j_0,j) \tag{4.35}$$

for $t = 1,...,l$. The reestimation formulas for π, A and B can be derived by maximizing Baum's auxiliary function

$$Q(\mu,\bar{\mu}) = \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)\log P_{\bar{\mu}}(Y_{-M+1},...,Y_T,X_1,...,X_T) \tag{4.36}$$

over $\bar{\mu}$, where μ is the current model and $\bar{\mu}$ is the reestimated model. The complete derivation of the reestimation formulas is given in Appendix III. Using the variables defined above, the reestimation formulas for π are

$$\hat{\pi}_1(j_0) = P(X_1 = j_0 \mid Y_{-M+1},...,Y_T)$$

$$= \gamma_1(j_0) \tag{4.37}$$

for $t = 1$, and

$$\hat{\pi}_{t|1,...,t-1}(j_{t-1},...,j_0) = P(X_t = j_0 \mid Y_{-M+1},...,Y_T,X_1 = j_{t-1},...,X_{t-1} = j_1)$$

$$= \frac{\gamma_t(j_{t-1},...,j_0)}{\gamma_{t-1}(j_{t-1},...,j_1)} \tag{4.38}$$

for $t = 2,...,l$. The reestimation formulas for A and B are

$$\hat{a}_{j_{l-1},...,j_0,j} = \frac{\sum_{t=l}^{T-1} \varepsilon_t(j_{l-1},...,j_0,j)}{\sum_{t=l}^{T-1} \gamma_t(j_{l-1},...,j_0)} \tag{4.39}$$

and

$$\hat{b}^{j_0}_{i_{j_0},...,i_0} = \frac{\sum_{t=1:\, Y_{t-j_0}=i_{j_0},...,Y_t=i_0}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1 \in S(X)} \gamma_t(j_{l-1},...,j_0)}{\sum_{t=1:\, Y_{t-j_0}=i_{j_0},...,Y_{t-1}=i_1}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1 \in S(X)} \gamma_t(j_{l-1},...,j_0)}. \tag{4.40}$$

The estimation process starts with a set of starting values for π, A and B. Based on Eq. (4.37) – Eq. (4.40), a new set of values for π, A and B can be calculated. The final estimates of parameters are found by repeating the process until the likelihood of the observed sequence converges.

4.4 Find the Optimal Hidden State Sequence

With the model parameters estimated using above reestimation formulas, we now need to find the optimal sequence of hidden states that maximizes the following probability:

$$P(X_1,...,X_T \mid Y_{-M+1},...,Y_T). \tag{4.41}$$

Using the Viterbi algorithm, this problem can be solved. Define $\delta_1 = \{{}_r\delta_1(j_0)\}_{j_0 \in S(X)}$ as a size $M+1$ vector with element

$$\begin{aligned} \delta_1(j_0) &= P(Y_{-M+1},...,Y_1, X_1 = j_0) \\ &= \pi_1(j_0) b^{j_0}_{y_{-j0+1},...,y_1}, \end{aligned} \tag{4.42}$$

for $t=1$. For $t=2$, $\delta_2=\{{}_r\delta_2(j_0)\}_{j_0\in S(X)}$ is a size $(M+1)^2$ vector with element

$\delta_2(j_0)=\{{}_r\delta_2(j_{1,}j_0)\}_{j_1\in S(X)}$, where

$$\delta_2(j_{1,}j_0)=P(Y_{-M+1},...,Y_2,X_1=j_1,X_2=j_0)$$

$$=\delta_1(j_1)\pi_{2|1}(j_1,j_0)b^{j_0}_{y_{-j_0+2},...,y_2}. \tag{4.43}$$

For $t=3,...,l$, $\delta_t=\{{}_r\delta_t(j_0)\}_{j_0\in S(X)}$ are vectors of size $(M+1)^3$ to $(M+1)^l$ with element

$\delta_t(j_0)=\{{}_r\delta_t(j_{t-1},...,j_{1,}j_0)\}_{j_{t-1},...,j_1\in S(X)}$, where

$$\delta_t(j_{t-1},...,j_0)=P(Y_{-M+1},...,Y_t,X_1=j_{t-1},...,X_{t-1}=j_1,X_t=j_0)$$

$$=\delta_{t-1}(j_{t-1},...,j_1)\pi_{t|1,...,t-1}(j_{t-1},...,j_0)b^{j_0}_{y_{t-j_0},...,y_t}. \tag{4.44}$$

For $t=l+1,...,T$, $\delta_t=\{{}_r\delta_t(j_0)\}_{j_0\in S(X)}$ are vectors of size $(M+1)^{l+1}$ with element

$\delta_t(j_0)=\{{}_r\delta_t(j_l,...,j_{1,}j_0)\}_{j_{l-1},...,j_1\in S(X)}$ and

$$\delta_t(j_l,...,j_0)=P(Y_{-M+1},...,Y_t,X_{t-l}=j_l,...,X_{t-1}=j_1,X_t=j_0)$$

$$=\delta_{t-1}(j_l,...,j_1)a_{j_l,...,j_1,j_0}b^{j_0}_{y_{t-j_0},...,y_t}. \tag{4.45}$$

In any function δ_t, the variable *j* always starts with 0 and end with M. For example,

$$\delta_2=\{{}_r\delta_2(j_0)\}_{j_0\in S(X)}=\{\delta_2(0),...,\delta_2(M)\} \tag{4.46}$$

and

$$\delta_2(j_0)=\{{}_r\delta_2(j_1,j_0)\}_{j_1\in S(X)}=\{\delta_2(0,j_0),...,\delta_2(M,j_0)\}. \tag{4.47}$$

Therefore, the hidden state at the last position of the sequence is estimated as

$$\hat{X}_T=floor\left(\frac{\arg\max(\delta_T)-1}{(M+1)^l}\right)+1, \tag{4.48}$$

where $\arg\max(\delta_T)$ returns the position of the largest value in δ_T. The rest of the optimal sequence of hidden state can be estimated by path backtracking as

$$\hat{X}_t = floor\left(\frac{\arg\max\left[\delta_{t+1}(\hat{X}_{t+1})\right]-1}{(M+1)^{l-1}}\right)+1 \tag{4.47}$$

for $t = T-1,...,l$, and

$$\hat{X}_t = floor\left(\frac{\arg\max\left[\delta_{t+1}(\hat{X}_{t+1})\right]-1}{(M+1)^{t-1}}\right)+1 \tag{4.48}$$

for $t = l-1,...,1$.

4.5 Application of MMMHS

With the MMMHS and the solutions for its three fundamental questions completely specified, we use the model to analyze three data sets from different fields and compare the results with that of MC, HMM and DCMM using the Bayesian Information Criterion (BIC). BIC has been widely used for selection of maximum likelihood-based models. It is defined as

$$BIC = -2\log(L) + p\log(T), \tag{4.49}$$

where L is the maximized likelihood for the estimated model, p is the number of free parameters to be estimated, and T is the number of data points in the likelihood. Based on the discussion on adjusting degree of freedom of model while empty estimated cells occur in Bishop *et al.* (1975), p does not account for the parameters estimated to be zero.

4.5.1 Analysis of Mouse αA-crystallin Gene

Nucleotides within DNA sequences are expected to depend on each other. Therefore, it is common practice to model a DNA sequence using Markov models. In this section, we examine a mouse

Model	Number of Parameters	Log-Likelihood	BIC
Independence	1	-901.86	1810.9
MC 1	2	-889.25	1792.8
MC 2	4	-884.73	1798.1
MC 3	8	-878.20	1813.8
MC 4	16	-869.36	1853.5
MC 5	32	-858.50	1946.5
MTD 2	3	-884.89	1791.3
MTD 3	4	-884.17	1797.0
MTD 4	5	-880.84	1797.5
MTD 5	6	-880.73	1804.5
HMM 2 (1)	4	-888.68	1806.0
HMM 2 (M2)	5	-883.51	1802.9
HMM 2 (2)	6	-883.39	1809.8
HMM 2 (M3)	6	-881.82	1806.7
HMM 2 (3)	10	-867.50	1806.7
HMM 2 (4)	17	-862.09	1846.1
HMM 3 (1)	9	-881.68	1827.9
HMM 3 (M2)	10	-881.38	1834.5
HMM 3 (2)	17	-870.76	1863.4
HMM 4 (1)	12	-875.35	1836.8
DCMM 2 (1;1)	**6**	**-873.71**	**1790.5**
DCMM 2 (2;1)	8	-872.71	1802.8
DCMM 2 (1;M2)	8	-873.57	1804.5
DCMM 2 (1;2)	10	-863.32	1798.4
DCMM 2 (M2;M2)	9	-867.98	1800.5
DCMM 2 (2;M2)	10	-860.01	1791.7
DCMM 2 (2;2)	12	-859.31	1804.7
DCMM 2 (1;M3)	10	-863.53	1798.8
DCMM 2 (2;M3)	12	-859.18	1804.4
DCMM 2 (M3;M3)	12	-866.17	1818.4
DCMM 3 (1;1)	10	-866.29	1804.3
DCMM 3 (2;1)	16	-850.60	1815.9
DCMM 3 (1;2)	17	-856.82	1835.6
DCMM 3 (2;2)	21	-848.71	1848.0
MMMHS 2 (1)	**4**	**-878.94**	**1786.6**
MMMHS 2 (2)	**5**	**-877.41**	**1790.7**
MMMHS 3 (1)	11	-861.21	1801.3
MMMHS 3 (2)	25	-856.38	1892.0

Table 4.1 Different types of Markov model for the mouse αA-crystallin gene.
MC *a*: Markov chain of order *a*.
MTD *a*: mixture transition distribution model of order *a*.
HMM *b* (*a*): *b* hidden states HMM of order *a*.
DCMM *b* (*a*; *c*): *b* hidden states DCMM with hidden order *a* and visible order *c*.
MMMHS *b* (*a*): *b* hidden states MMMHS with hidden order *a*.
M: MTD approximation.

αA-crystallin gene, which was analyzed in several papers using Markov models. Raftery and Tavare (1994) presented the results of modeling this sequence with different order Markov chains and Mixture Transition Distribution models (MTD). Berchtold (2002) showed that the results can be improved by using the DCMM.

The mouse αA-crystallin gene contains 1307 nucleotides, which are provided in the Table 7 of Raftery and Tavare (1994). To be comparable with the results given in Raftery and Tavare (1994) and Berchtold (2002), the first 5 nucleotides were dropped from the calculation of the log-likelihood and each nucleotide is coded as either purine or pyrimidine. Table 4.1 shows the results of fitting different types of Markov model to the gene sequence (Results of MC, HMM, MTD and HMM are originally from Berchtold (2002)). We can see from the table that models with more complex dependence structure tend to provide higher log-likelihood, but the large number of parameters leads to higher BIC. Based on the BIC values, the best model is the MMMHS 2 (1) with only 4 independent parameters. The estimated parameters are

$$\pi = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$A = \begin{pmatrix} 0.1569 & 1 \\ 0.8431 & 0 \end{pmatrix},$$

$$B^0 = \begin{pmatrix} 0.5408 \\ 0.4592 \end{pmatrix}, \qquad B^1 = \begin{pmatrix} 0.6190 & 0.1824 \\ 0.3810 & 0.8176 \end{pmatrix}.$$

The next two models with lowest BIC values are DCMM 2 (1; 1) and MMMHS 2 (2). In MC category, the model with lowest BIC is MC 1. In MTD category, MTD 2 results in the best BIC. Generally, we can see that, for the sequence analyzed in this application, the best model in each category is the one with the simplest dependence structure.

4.5.2 Analysis of Wind Speed Time Series

The data set analyzed in this section is a time series of daily average wind speeds for 1961--1978 collected at the meteorological station at Roche's Point in the Republic of Ireland. The dependence structure of data was first analyzed in Haslett and Raftery (1989). Berchtold (1999) also tried to explain the short term autocorrelation in the time series using MC, HMM and DCMM. There are a total of 6574 observations in the dataset, which are recorded in knots. The first 4 observations are dropped from calculation of the likelihood so that the resulting log-likelihood and BIC are comparable for each model. Since the data are continuous, each observation of wind speed is classified into one of the three categories: low (< 5 knots), normal (5-20 knots) and high (> 20 knots).

Model	Number of Parameters	Log-Likelihood	BIC
Independence	2	-3805.1	7627.9
MC 1	6	-3508.2	7069.1
MC 2	14	-3491.2	7105.4
MC 3	30	-3469.5	7202.8
MC 4	60	-3434.7	7396.7
MTD 2	5	-3499.7	7043.3
MTD 3	6	-3494.6	7042.0
MTD 4	7	-3490.1	7041.8
HMM 2 (2)	5	-3577.8	7199.5
HMM 2 (3)	9	-3476.1	7031.3
DCMM 2 (1;1)	12	-3448.2	7001.9
DCMM 3 (1;1)	15	-3445.9	7023.6
MMMHS 2 (1)	**9**	**-2767.0**	**5613.0**
MMMHS 2 (2)	**10**	**-2774.6**	**5637.2**
MMMHS 3 (1)	**22**	**-2738.3**	**5670.1**
MMMHS 3 (2)	28	-2737.0	5720.2

Table 4.2 Different types of Markov model for the wind speed time series.

Results of fitting different models to the wind speed data are shown in Table 4.2 (Results of MC, HMM, MTD, HMM and DCMM are originally from Berchtold (1999)). We can see that models in MMMHS category show significant improvement on both log-likelihood and BIC over the models in

other categories. The best model is the 2-state MMMHS with hidden order 1. The estimated parameters are

$$\pi = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$A = \begin{pmatrix} 0.9754 & 0.0073 \\ 0.0246 & 0.9927 \end{pmatrix},$$

$$B^0 = \begin{pmatrix} 0.0629 \\ 0.9338 \\ 0.0033 \end{pmatrix}, \qquad B^1 = \begin{pmatrix} 0.2906 & 0.0441 & 0 \\ 0.6912 & 0.8870 & 0.7076 \\ 0.0182 & 0.0689 & 0.2924 \end{pmatrix}.$$

By definition, the hidden states of the 2-state MMMHS are 0 and 1. Based on the estimated hidden state transition matrix A, the hidden states are more likely to stay constant from day to day. The transition probabilities of 0 to 0 and 1 to 1 are 0.9754 and 0.9927, respectively. Given the hidden state is 0, which means that the wind speed is independent of that of previous day, the estimated transition matrix B^0 shows that the probability of observing the normal wind speed is 0.9338. A hidden state of 1 indicates that wind speed is correlated to that of the previous day. The resulting transition matrix B^1 shows that the probability of going from normal speed to normal speed is 0.887. The probability of going from low speed to high speed, or from high speed to low speed are very small, 0.0182 and 0, respectively.

The model with lowest BIC in other categories is DCMM 2 (1; 1), 2-state DCMM with hidden order 1 and visible order 1. Berchtold (1999) provided the estimated parameters for the model as below:

$$\pi = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$A = \begin{pmatrix} 0.9875 & 0.0148 \\ 0.0125 & 0.9852 \end{pmatrix},$$

$$B^0 = \begin{pmatrix} 0.3550 & 0.0805 & 0.0228 \\ 0.6450 & 0.8874 & 0.7721 \\ 0 & 0.0321 & 0.2051 \end{pmatrix}, \quad B^1 = \begin{pmatrix} 0.1973 & 0.0361 & 0 \\ 0.7846 & 0.8137 & 0.6826 \\ 0.0181 & 0.1502 & 0.3174 \end{pmatrix}.$$

We can see that the 1st-order transition matrices for the wind speed and hidden states from both models are consistent with each other, except that DCMM 2 (1; 1) assumes 1st-order dependence under both hidden states.

4.5.3 Analysis of the Song of Wood Pewee

The wood pewee is a small tyrant flycatcher from North America and its song contains three different phases 1, 2 and 3. A song of the wood pewee was originally introduced by Craig (1943). The song of length 1327 was analyzed in Chatfield and Lemon (1970), Bishop *et al.* (1975), Raftery and Tavare (1994) and Berchtold (2001 & 2002) using different types of Markov models. The complete data can be found in Table 12 of Raftery and Tavare (1994).

It is interesting to find that the song sequence is dominated by several patterns (Table 4.3). Note that the song has 1327 phases, so there are total 1324 patterns of length 4. We can see that the 4 patterns of length 4 in Table 4.3 have a total frequency of 1057. For the patterns of different lengths, we see the similar domination in the sequence. To be comparable with the modeling results in Berchtold (2002), 1323 data points were used to compute the likelihood. Results are reported in Table 4.4.

The model with the lowest BIC value in Table 4.4 (Results of MC, HMM, and HMM are originally from Berchtold (2002)) is DCMM 2 (2; 2), 2-state DCMM with hidden order 2 and visible order 2. The same model also resulted in one of the highest log-likelihoods. MMMHS with 2 hidden states shows poor performance on both log-likelihood and BIC since the model assumes 1st-order dependency

Pattern	Frequency
1 2	348
1 3	276
2 1	346
3 1	278
1 2 1	344
1 3 1	275
2 1 3	263
3 1 2	267
1 2 1 3	263
1 3 1 2	266
2 1 3 1	263
3 1 2 1	265
1 2 1 3 1	263
1 3 1 2 1	264
2 1 3 1 2	255
3 1 2 1 3	224
1 2 1 3 1 2	255
1 3 1 2 1 3	223
2 1 3 1 2 1	253
3 1 2 1 3 1	224

Table 4.3 Patterns with >200 frequency in the song of wood pewee.

Model	Number of Parameters	Log-Likelihood	BIC
Independence	2	-1349.4	2713.3
MC 1	5	-694.1	1424.2
MC 2	9	-368.6	801.9
MC 3	14	-354.0	808.6
MC 4	19	-315.8	768.3
HMM 2 (1)	7	-1086.4	2223.2
DCMM 2 (1;2)	17	-367.5	857.2
DCMM 2 (1;M2)	7	-384.1	818.4
DCMM 2 (2;2)	**17**	**-305.4**	**733.0**
DCMM 2 (2;M2)	9	-383.8	832.2
DCMM 3 (1;2)	15	-344.0	795.8
DCMM 3 (2;2)	23	-304.6	774.5
MMMHS 2 (1)	7	-650.9	1352.1
MMMHS 2 (2)	10	-634.1	1340.1
MMMHS 3 (1)	13	-337.5	768.4
MMMHS 4 (1)	**22**	**-301.7**	**761.5**
MMMHS 5 (1)	**23**	**-289.8**	**744.9**

Table 4.4 Different types of Markov model for the song of wood pewee.

and independence with only two hidden states. MMMHS 3 (1) shows significantly improved results from MMMHS 2 with a third hidden state that assumes 2^{nd}-order dependency in the data. Based on the dominating patterns shown on Table 4.3, it is not surprising to see that a conventional Markov model of 4^{th}-order could fit very well to the data. The fact that the two lowest log-likelihoods resulted from MMMHS 4 (1) and MMMHS 5 (1) also indicates that 3rd and 4th-order dependence prevails in the data.

4.6 Conclusion

In this part of the study, we developed a stochastic model MMMHS for representing heterogeneous sequences. MMMHS is very similar to the conventional HMM and DCMM in terms of using hidden states to describe the non-homogeneity of a sequence, but it presents a more flexible dependency structure by changing the order of Markov dependency under different hidden states. We also extended the forward-backward procedure to MMMHS and provided the complete estimation procedure based on EM algorithm. Applications of MMMHS to different types of data showed that MMMHS provides better representation for some data sets than the other models.

5. Conclusion

5.1 Summary

This dissertation introduced two statistical methodologies developed on the same mathematical background, multi-order Markov models. At first, we presented a new alignment-free method MTM for sequence comparison. The main feature of MTM method is that it takes into consideration the compositional structure of DNA sequences, and represents the DNA sequences in a comprehensive manner so that less information is lost when comparing sequences. Applications on both simulated and real data sets show that the MTM method can be successfully applied to phylogeny inference using either the whole genomes or the genomic segments.

The idea of multi-order dependency carried by the MTM method extended to the second part of the dissertation, where we also developed a stochastic model called MMMHS to represent non-homogeneous sequence. MMMHS can be seen as an extension of traditional HMM and DCMM with variable visible orders under different hidden states. We also provided the complete algorithm for model estimation. Applications of MMMHS on three types of real data sets showed improved results over the existing models. Especially, in the analysis of wind speed time series, the MMMHS significantly improved the log-likelihood and BIC value. Analysis of the song of the wood pewee showed that the MMMHS can also be effective in modeling data with repeating patterns.

5.1 Future Research

While the results presented provide strong support that the MTM can be successful applied to phylogeny inference, this research can be extended in different directions. First of all, the MTM method adopts an angle cosine based distance measure. A natural extension in the future would be to investigate the behavior of different distance metrics with the MTM, such as the correlation-based and information theory-based metrics. Second, the order selection would include all transition probabilities of same order.

Therefore, it is of interest to explore the possibility of a second level of selection, which would lead to more accurate representation of the sequence and more efficient computation.

A major concern with MMMHS is that the parameter space could be very large when high order dependence is considered among the hidden and visible states. Raftery and Tavare(1994) and Berchtold (2002) have shown that, for high order Markov model and double chain Markov model, the number of parameters can be reduced by replacing each transition matrix with a Mixture Transition Distribution (MTD) model. It is therefore of interest to explore the application of MTD model on MMMHS. In addition, initiative should be taken to further investigate the application of the corresponding hidden state sequence in the area of sequence decomposition and classification.

Appendix I: Data Sets for Section 3.7 & 3.8

1	dk/Laos/3295/06, DQ845348
2	commonmagpie/HK/645/2006, DQ992839
3	JapaneseWhiteEye/HK/1038/06, DQ992842
4	ck/Guiyang/3055/05, DQ992755
5	dk/Guiyang/3009/05, DQ992754
6	gs/Guiyang/3422/05, DQ992757
7	gs/Guangxi/3017/05, DQ992716
8	gs/Guangxi/345/05, DQ320896
9	ck/YN/374/04, AY651371
10	ck/YN/115/04, AY651372
11	gs/Guangxi/3316/05, DQ992719
12	dk/Hunan/149/05, DQ320904
13	dk/Hunan/139/05, DQ320903
14	dk/Hunan/127/05, DQ320902
15	dk/Egypt/22533/06, DQ862002
16	Egypt/0636NAMRU3/2007, EF382359
17	Egypt/14724NAMRU3/2006, EF200512
18	turkey/Turkey1//05, DQ407519
19	Turkey/15/06, EF619989
20	ck/Kyoto/3/04, AB188824
21	crow/Kyoto/53/04, AB189053
22	Indonesia/CDC1046/07, CY019408
23	Indonesia/CDC887/06, CY017688
24	Indonesia/CDC1047/07, CY019424
25	Indonesia/CDC940/06, CY017654
26	Indonesia/5/05, EF541394
27	Indonesia/546bH/06, EU146793
28	Indonesia/CDC625/06, CY014433
29	ck/Indonesia/11/03, EF473081
30	ck/Indonesia/7/03, EF473080
31	VN/1194/04, EF541402
32	TH/16/04, EF541408
33	TH/676/05, DQ360835
34	HK/213/03, EF541401
35	ck/Henan/12/04, AY950232
36	ck/Henan/16/04, AY950234
37	ck/HK/YU22/02, AY651349
38	ck/HK/YU777/02, AY575877
39	dk/Hubei/wg/02, DQ997094
40	sw/Anhui/ca/04, DQ997392
41	blbird/Hunan/1/04, AY741213
42	dk/Guangxi/2396/04, DQ320892

43	dk/Guangxi/1378/04, DQ320884
44	ck/Guiyang/1218/06, DQ992772
45	ck/Guiyang/441/06, DQ992766
46	ck/Guiyang/846/06, DQ992769
47	dk/Guiyang/504/06, DQ992918
48	ck/HK/891.1/01, AF509034
49	ck/HK/879.1/01, AF509031
50	gs/Guangdong/1/96, AF144305
51	HK/156/97, AF028709
52	Iraq/207NAMRU3/06, DQ435202

Table A1. List of 52 influenza A viral HA sequences.

1	human (Homo sapiens, V00662)
2	common chimpanzee(Pan troglodytes, D38116)
3	pigmy chimpanzee (Pan paniscus,D38113)
4	gorilla (Gorilla gorilla, D38114)
5	orangutan (Pongo pygmaeus, D38115)
6	gibbon (Hylobates lar, X99256)
7	baboon (Papio hamadryas, Y18001)
8	horse (Equus caballus,X79547)
9	white rhinoceros (Ceratotherium simum, Y07726)
10	harbor seal (Phoca vitulina, X63726)
11	gray seal (Halichoerus grypus, X72004)
12	cat (Felis catus, U20753)
13	fin whale (Balenoptera physalus, X61145)
14	blue whale (Balenoptera musculus, X72204)
15	cow (Bos taurus, V00654)
16	rat (Rattus norvegicus, X14848)
17	mouse (Mus musculus, V00711)
18	opossum (Didelphis virginiana, Z29573)
19	wallaroo (Macropus robustus, Y10524)
20	platypus (Ornithorhyncus anatinus,X83427)
21	squirrel (Sciurus vulgaris, AJ238588)
22	guinea pig (Cavia porcellus,AJ222767)
23	donkey (Equus asinus, X97337)
24	Indian rhinoceros (Rhinoceros unicornis, X97336)
25	dog (Canis familiaris,U96639)
26	sheep (Ovis aries, AF010406)
27	pig (Sus scrofa,AJ002189)
28	hippopotamus (Hippopotamus amphibius,AJ010957)
29	Dormouse (Glis glis, AJ001562)

Table A2. List of 29 complete mitochondrial genome sequences.

Appendix II: Derivation of Forward-Backward Procedures and Parameter Estimations

Eq. (4.18)

$$\alpha_1(j_0) = P(Y_{-M+1},...,Y_1, X_1 = j_0)$$
$$= P(Y_1 \mid Y_{-M+1},...,Y_0, X_1 = j_0)P(Y_{-M+1},...,Y_0, X_1 = j_0)$$
$$= P(Y_1 \mid Y_{-j_0+1},...,Y_0, X_1 = j_0)P(X_1 = j_0)$$
$$= b^{j_0}_{y_{-j_0+1},...,y_1}\pi_1(j_0)$$

Eq. (4.19)

$$\alpha_2(j_1, j_0) = P(Y_{-M+1},...,Y_2, X_1 = j_1, X_2 = j_0)$$
$$= P(Y_2 \mid Y_{-M+1},...,Y_1, X_1 = j_1, X_2 = j_0)P(X_2 = j_0 \mid Y_{-M+1},...,Y_1, X_1 = j_1)P(Y_{-M+1},...,Y_1, X_1 = j_1)$$
$$= P(Y_2 \mid Y_{-j_0+2},...,Y_1, X_2 = j_0)P(X_2 = j_0 \mid X_1 = j_1)\alpha_1(j_1)$$
$$= b^{j_0}_{y_{-j_0+2},...,y_2}\pi_{2|1}(j_1, j_0)\alpha_1(j_1)$$

Eq. (4.20)

$$\alpha_t(j_{t-1},..., j_0) = P(Y_{-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0)$$
$$= P(Y_t \mid Y_{-M+1},...,Y_{t-1}, X_1 = j_{t-1},..., X_t = j_0)P(X_t = j_0 \mid Y_{-M+1},...,Y_{t-1}, X_1 = j_{t-1},..., X_{t-1} = j_1)$$
$$\quad P(Y_{-M+1},...,Y_{t-1}, X_1 = j_{t-1},..., X_{t-1} = j_1)$$
$$= P(Y_t \mid Y_{t-j_0},...,Y_{t-1}, X_t = j_0)P(X_t = j_0 \mid X_1 = j_{t-1},..., X_{t-1} = j_1)P(Y_{-M+1},...,Y_{t-1}, X_1 = j_{t-1},..., X_{t-1} = j_1)$$
$$= b^{j_0}_{y_{t-j_0},...,y_t}\pi_{t|1,...,t-1}(j_{t-1,...,}j_0)\alpha_{t-1}(j_{t-1},..., j_1)$$

Eq. (4.21)

$$\alpha_t(j_{l-1},..., j_0) = P(Y_{-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$= P(Y_t \mid Y_{-M+1},...,Y_{t-1}, X_{t-l+1} = j_{l-1},..., X_t = j_0)P(Y_{-M+1},...,Y_{t-1}, X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$= P(Y_t \mid Y_{t-j_0},...,Y_{t-1}, X_t = j_0)\sum_{j_l \in S(X)} P(Y_{-M+1},...,Y_{t-1}, X_{t-l} = j_l,..., X_t = j_0)$$
$$= b^{j_0}_{y_{t-j_0},...,y_t}\sum_{j_l \in S(X)} P(X_t = j_0 \mid Y_{-M+1},...,Y_{t-1}, X_{t-l} = j_l,..., X_{t-1} = j_1)P(Y_{-M+1},...,Y_{t-1}, X_{t-l} = j_l,..., X_{t-1} = j_1)$$
$$= b^{j_0}_{y_{t-j_0},...,y_t}\sum_{j_l \in S(X)} P(X_t = j_0 \mid X_{t-l} = j_l,..., X_{t-1} = j_1)\alpha_{t-1}(j_l,..., j_1)$$
$$= b^{j_0}_{y_{t-j_0},...,y_t}\sum_{j_l \in S(X)} a_{j_{l,...,}j_0}\alpha_{t-1}(j_l,..., j_1)$$

Eq. (4.25)

$$\beta_t(j_{l-1},..., j_0) = P(Y_{t+1},...,Y_T \mid Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$= \frac{P(Y_{t-M+1},...,Y_T, X_{t-l+1} = j_{l-1},..., X_t = j_0)}{P(Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)}$$

$$= \frac{\sum_{j \in S(X)} P(Y_{t-M+1},...,Y_T, X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j)}{P(Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)}$$

$$= \frac{1}{P(Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)} \sum_{j \in S(X)} P(Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$P(X_{t+1} = j \mid Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0) P(Y_{t+1} \mid Y_{t-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j)$$
$$P(Y_{t+2},...,Y_T \mid Y_{t-M+1},...,Y_{t+1}, X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j)$$

$$= \sum_{j \in S(X)} P(X_{t+1} = j \mid X_{t-l+1} = j_{l-1},..., X_t = j_0) P(Y_{t+1} \mid Y_{t-j+1},..., Y_t, X_{t+1} = j)$$
$$P(Y_{t+2},...,Y_T \mid Y_{t-M+1},...,Y_{t+1}, X_{t-l+2} = j_{l-2},..., X_t = j_0, X_{t+1} = j)$$

$$= \sum_{j \in S(X)} a_{j_{l-1},... j_0, j} b^j_{y_{t-j+1},...,y_{t+1}} \beta_{t+1}(j_{l-2},..., j_0, j)$$

Eq. (4.26)

$$\beta_t(j_{t-1},..., j_0) = P(Y_{t+1},...,Y_T \mid Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0)$$

$$= \frac{P(Y_{t-M+1},...,Y_T, X_1 = j_{t-1},..., X_t = j_0)}{P(Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0)}$$

$$= \frac{\sum_{j \in S(X)} P(Y_{t-M+1},...,Y_T, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)}{P(Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0)}$$

$$= \frac{1}{P(Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0)} \sum_{j \in S(X)} P(Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0)$$
$$P(X_{t+1} = j \mid Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0) P(Y_{t+1} \mid Y_{t-M+1},...,Y_t, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)$$
$$P(Y_{t+2},...,Y_T \mid Y_{t-M+1},...,Y_{t+1}, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)$$

$$= \sum_{j \in S(X)} P(X_{t+1} = j \mid X_1 = j_{t-1},..., X_t = j_0) P(Y_{t+1} \mid Y_{t-j+1},..., Y_t, X_{t+1} = j)$$
$$P(Y_{t+2},...,Y_T \mid Y_{t-M+1},...,Y_{t+1}, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)$$

$$= \sum_{j \in S(X)} \pi_{t+1|1,...,t}(j_{t-1},... j_0, j) b^j_{y_{t-j+1},...,y_{t+1}} \beta_{t+1}(j_{t-1},..., j_0, j)$$

Eq. (4.30)

$$\varepsilon_t(j_{l-1},..., j_0, j) = P(X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j \mid Y_{-M+1},...,Y_T)$$

$$= \frac{P(Y_{-M+1},...,Y_T, X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j)}{P(Y_{-M+1},...,Y_T)}$$

$$= \frac{1}{P(Y_{-M+1},...,Y_T)} P(Y_{-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0) P(X_{t+1} = j \mid Y_{-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$P(Y_{t+1} \mid Y_{-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j) P(Y_{t+2},...,Y_T \mid Y_{-M+1},...,Y_{t+1}, X_{t-l+1} = j_{l-1},..., X_t = j_0, X_{t+1} = j)$$

$$= \frac{1}{P(Y_{-M+1},...,Y_T)} P(Y_{-M+1},...,Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0) P(X_{t+1} = j \mid X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$P(Y_{t+1} \mid Y_{t-j+1},..., Y_t, X_{t+1} = j) P(Y_{t+2},..., Y_T \mid Y_{t-M+2},..., Y_{t+1}, X_{t-l+2} = j_{l-2},..., X_t = j_0, X_{t+1} = j)$$
$$= \frac{\alpha_t(j_{l-1},..., j_0) a_{j_{l-1},..., j_0, j} b^j_{y_{t-j+1},..., y_{t+1}} \beta_{t+1}(j_{l-2},..., j_0, j)}{L(Y_{-M+1},..., Y_T)}$$

Eq. (4.31)

$$\varepsilon_t(j_{t-1},..., j_0, j) = P(X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j \mid Y_{-M+1},..., Y_T)$$
$$= \frac{P(Y_{-M+1},..., Y_T, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)}{P(Y_{-M+1},..., Y_T)}$$
$$= \frac{1}{P(Y_{-M+1},..., Y_T)} P(Y_{-M+1},..., Y_t, X_1 = j_{t-1},..., X_t = j_0) P(X_{t+1} = j \mid Y_{-M+1},..., Y_t, X_1 = j_{t-1},..., X_t = j_0)$$
$$P(Y_{t+1} \mid Y_{-M+1},..., Y_t, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j) P(Y_{t+2},..., Y_T \mid Y_{-M+1},..., Y_{t+1}, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)$$

$$= \frac{1}{P(Y_{-M+1},..., Y_T)} P(Y_{-M+1},..., Y_t, X_1 = j_{t-1},..., X_t = j_0) P(X_{t+1} = j \mid X_1 = j_{t-1},..., X_t = j_0)$$
$$P(Y_{t+1} \mid Y_{t-j+1},..., Y_t, X_{t+1} = j) P(Y_{t+2},..., Y_T \mid Y_{t-M+2},..., Y_{t+1}, X_1 = j_{t-1},..., X_t = j_0, X_{t+1} = j)$$
$$= \frac{\alpha_t(j_{t-1},..., j_0) \pi_{t+1|1,...,t}(j_{t-1},..., j_0, j) b^j_{y_{t-j+1},..., y_{t+1}} \beta_{t+1}(j_{t-1},..., j_0, j)}{L(Y_{-M+1},..., Y_T)}$$

Eq. (4.32)

$$\gamma_t(j_{l-1},..., j_0) = P(X_{t-l+1} = j_{l-1},..., X_t = j_0 \mid Y_{-M+1},..., Y_T)$$
$$= \frac{P(Y_{-M+1},..., Y_T, X_{t-l+1} = j_{l-1},..., X_t = j_0)}{P(Y_{-M+1},..., Y_T)}$$
$$= \frac{1}{P(Y_{-M+1},..., Y_T)} P(Y_{-M+1},..., Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0) P(Y_{t+1},..., Y_T \mid Y_{-M+1},..., Y_t, X_{t-l+1} = j_{l-1},..., X_t = j_0)$$
$$= \frac{\alpha_t(j_{l-1},..., j_0) \beta_t(j_{l-1},..., j_0)}{L(Y_{-M+1},..., Y_T)}$$

Eq. (4.33)

$$\gamma_t(j_{t-1},..., j_0) = P(X_1 = j_{t-1},..., X_t = j_0 \mid Y_{-M+1},..., Y_T)$$
$$= \frac{P(Y_{-M+1},..., Y_T, X_1 = j_{t-1},..., X_t = j_0)}{P(Y_{-M+1},..., Y_T)}$$
$$= \frac{1}{P(Y_{-M+1},..., Y_T)} P(Y_{-M+1},..., Y_t, X_1 = j_{t-1},..., X_t = j_0) P(Y_{t+1},..., Y_T \mid Y_{-M+1},..., Y_t, X_1 = j_{t-1},..., X_t = j_0)$$
$$= \frac{\alpha_t(j_{t-1},..., j_0) \beta_t(j_{t-1},..., j_0)}{L(Y_{-M+1},..., Y_T)}$$

Eq. (4.38)

$$\hat{\pi}_{t|1,\ldots,t-1}(j_{t-1},\ldots,j_0) = P(X_t = j_0 \mid Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_{t-1} = j_1)$$
$$= \frac{P(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_t = j_0)}{P(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_{t-1} = j_1)}$$
$$= \frac{P(X_1 = j_{t-1},\ldots,X_t = j_0 \mid Y_{-M+1},\ldots,Y_T)P(Y_{-M+1},\ldots,Y_T)}{P(X_1 = j_{t-1},\ldots,X_{t-1} = j_1 \mid Y_{-M+1},\ldots,Y_T)P(Y_{-M+1},\ldots,Y_T)}$$
$$= \frac{\gamma_t(j_{t-1},\ldots,j_0)}{\gamma_{t-1}(j_{t-1},\ldots,j_1)}$$

Eq. (4.39)

$$\hat{a}_{j_{l-1},\ldots,j_0,j} = \frac{\sum_{t=l}^{T-1} P(Y_{-M+1},\ldots,Y_T, X_{t-l+1} = j_{l-1},\ldots,X_t = j_0, X_{t+1} = j)}{\sum_{t=l}^{T-1} P(Y_{-M+1},\ldots,Y_T, X_{t-l+1} = j_{l-1},\ldots,X_t = j_0)}$$
$$= \frac{\sum_{t=l}^{T-1} P(X_{t-l+1} = j_{l-1},\ldots,X_t = j_0, X_{t+1} = j \mid Y_{-M+1},\ldots,Y_T)P(Y_{-M+1},\ldots,Y_T)}{\sum_{t=l}^{T-1} P(X_{t-l+1} = j_{l-1},\ldots,X_t = j_0 \mid Y_{-M+1},\ldots,Y_T)P(Y_{-M+1},\ldots,Y_T)}$$
$$= \frac{\sum_{t=l}^{T-1} \varepsilon_t(j_{l-1},\ldots,j_0,j)}{\sum_{t=l}^{T-1} \gamma_t(j_{l-1},\ldots,j_0)}$$

Eq. (4.40)

$$\hat{b}^{j_0}_{i_{j_0},\ldots,i_0} = \frac{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} P(Y_{-M+1},\ldots,Y_T, X_t = j_0)}{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_{t-1}=i_1}^{T} P(Y_{-M+1},\ldots,Y_T, X_t = j_0)}$$
$$= \frac{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} P(Y_{-M+1},\ldots,Y_T, X_{t-j+1} = j_{l-1},\ldots,X_{t-1} = j_1, X_t = j_0)}{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_{t-1}=i_1}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} P(Y_{-M+1},\ldots,Y_T, X_{t-j+1} = j_{l-1},\ldots,X_{t-1} = j_1, X_t = j_0)}$$
$$= \frac{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} \gamma_t(j_{l-1},\ldots,j_0)}{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_{t-1}=i_1}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} \gamma_t(j_{l-1},\ldots,j_0)}$$

Appendix III: Derivation of Reestimation Formulas

The reestimation formulas for π, A and B are derived by maximizing Baum's auxiliary function

$$Q(\mu,\bar{\mu}) = \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)\log P_{\bar{\mu}}(Y_{-M+1},...,Y_T,X_1,...,X_T) \tag{A.1}$$

over $\bar{\mu}$. First of all, we need to rewrite the likelihood $P(Y_{-M+1},...,Y_t,X_1,...,X_t)$ in terms of the model parameters π, A and B. For $t=1$, the likelihood is

$$P(Y_{-M+1},...,Y_1,X_1)$$

$$= P(Y_1 \mid Y_{-M+1},...,Y_0,X_1)P(Y_{-M+1},...,Y_0,X_1)$$

$$= b^{X_1}_{y_{-X_1+1},...,y_1} P(X_1)$$

$$= b^{X_1}_{y_{-X_1+1},...,y_1} \pi_1(X_1)\,. \tag{A.2}$$

For $t=2$, it is

$$P(Y_{-M+1},...,Y_2,X_1,X_2)$$

$$= P(Y_2 \mid Y_{-M+1},...,Y_1,X_1,X_2)P(X_2 \mid Y_{-M+1},...,Y_1,X_1)P(Y_{-M+1},...,Y_1,X_1)$$

$$= P(Y_2 \mid Y_{-X_2+1},...,Y_1,X_2)P(X_2 \mid X_1)P(Y_{-M+1},...,Y_1,X_1)$$

$$= b^{X_2}_{y_{-X_2+2},...,y_2} \pi_{2|1}(X_1,X_2)P(Y_{-M+1},...,Y_1,X_1)$$

$$= \pi_1(X_1)\pi_{2|1}(X_1,X_2) b^{X_1}_{y_{-X_1+1},...,y_1} b^{X_2}_{y_{-X_2+2},...,y_2}\,. \tag{A.3}$$

For $t=3,...,l$,

$$P(Y_{-M+1},...,Y_t,X_1,..,X_t)$$

$$= P(Y_t \mid Y_{-M+1},...,Y_{t-1},X_1,...,X_t)P(X_t \mid Y_{-M+1},...,Y_{t-1},X_1,...,X_{t-1})P(Y_{-M+1},...,Y_{t-1},X_1,...,X_{t-1})$$

$$= P(Y_t \mid Y_{-X_t+t},...,Y_{t-1},X_t)P(X_t \mid X_1,...,X_{t-1})P(Y_{-M+1},...,Y_{t-1},X_1,...,X_{t-1})$$

$$= b^{X_t}_{y_{-X_t+t},...,y_t} \pi_{t|1,...,t-1}(X_1,...,X_t)P(Y_{-M+1},...,Y_{t-1},X_1,...,X_{t-1})$$

$$= \pi_1(X_1)\pi_{2|1}(X_1, X_2),...,\pi_{t|1,...,t-1}(X_1,...,X_t)\prod_{i=1}^{t} b^{X_i}_{y_{-X_i+i},...,y_i} \ . \tag{A.4}$$

For $t > l$,

$$P(Y_{-M+1},...,Y_t, X_1,..,X_t)$$

$$= P(Y_t \mid Y_{-M+1},...,Y_{t-1}, X_1,...,X_t)P(X_t \mid Y_{-M+1},...,Y_{t-1}, X_1,...,X_{t-1})P(Y_{-M+1},...,Y_{t-1}, X_1,...,X_{t-1})$$

$$= P(Y_t \mid Y_{-X_t+t},...,Y_{t-1}, X_t)P(X_t \mid X_{t-l},...,X_{t-1})P(Y_{-M+1},...,Y_{t-1}, X_1,...,X_{t-1})$$

$$= b^{X_t}_{y_{-X_t+t},...,y_t} a_{X_{t-l},...,X_t} P(Y_{-M+1},...,Y_{t-1}, X_1,...,X_{t-1})$$

$$= \pi_1(X_1)\pi_{2|1}(X_1, X_2),...,\pi_{l|1,...,l-1}(X_1,...,X_l)\prod_{t=l+1}^{t} a_{X_{t-l},...,X_t}\prod_{i=1}^{t} b^{X_i}_{y_{-X_i+i},...,y_i} \ . \tag{A.5}$$

Therefore, the complete likelihood based on the reestimation model $\bar{\mu}$ can be written as

$$P_{\bar{\mu}}(Y_{-M+1},...,Y_T, X_1,..,X_T)$$

$$= \pi_1(X_1)\pi_{2|1}(X_1, X_2),...,\pi_{l|1,...,l-1}(X_1,...,X_l)\prod_{t=l+1}^{T} a_{X_{t-l},...,X_t}\prod_{t=1}^{T} b^{X_t}_{y_{-X_t+t},...,y_t} \ . \tag{A.6}$$

Substituting Eq. (A.6) into the Baum's auxiliary function $Q(\mu, \bar{\mu})$ [Eq. (A.1)], we get

$$Q(\mu, \bar{\mu}) = \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T, X_1,...,X_T)\Big[\log \pi_1(X_1) + \log \pi_{2|1}(X_1, X_2)+,...,+\log \pi_{l|1,...,l-1}(X_1,...,X_l)$$

$$+\sum_{t=l+1}^{T} \log a_{X_{t-l},...,X_t} + \sum_{t=1}^{T} \log b^{X_t}_{y_{-X_t+t},...,y_t} \ \Big]. \tag{A.7}$$

To find the set of parameters that maximizes Eq. (A.7), we take the partial derivative with respect to each parameter and set it equal to zero.

To get the reestimation formula of $\pi_1(X_1 = j_0)$ subject to $\sum_{X_1 \in S(X)} \pi_1(X_1) = 1$, we rewrite Eq. (A.7) as

$$Q = ... + \sum_{X_2,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T, X_1 = j_0, X_2,...,X_T)\log \pi_1(X_1 = j_0) - \lambda(\sum_{X_1 \in S(X)} \pi_1(X_1) - 1), \tag{A.8}$$

where all terms that do not contain $\pi_1(X_1 = j)$ are excluded for the convenience of derivation. It can be shown that

$$\frac{\partial Q}{\partial \pi_1(X_1 = j_0)} = \frac{1}{\pi_1(X_1 = j_0)} \sum_{X_2,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T, X_1 = j_0, X_2,..., X_T) - \lambda . \tag{A.9}$$

Set Eq. (A.9) equal to zero, we get

$$\hat{\pi}_1(X_1 = j_0) = \frac{\sum_{X_2,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T, X_1 = j_0, X_2,..., X_T)}{\lambda}$$

$$= \frac{P_\mu(Y_{-M+1},...,Y_T, X_1 = j_0)}{\lambda} . \tag{A.10}$$

Based on the constraint $\sum_{X_1 \in S(X)} \hat{\pi}_1(X_1) = 1$, it can be shown that $\lambda = P_\mu(Y_{-M+1},...,Y_T)$. Therefore, the reestimation formula for $\pi_1(X_1 = j_0)$ is given as

$$\hat{\pi}_1(X_1 = j_0) = \frac{P_\mu(Y_{-M+1},...,Y_T, X_1 = j_0)}{P_\mu(Y_{-M+1},...,Y_T)}$$

$$= P_\mu(X_1 = j_0 \mid Y_{-M+1},...,Y_T)$$

$$= \gamma_1(j_0) . \tag{A.11}$$

For estimation of $\pi_{t|1,...,t-1}(X_1 = j_{t-1},..., X_t = j_0)$, Eq. (A.7) is rewritten as

$$Q = ... + \sum_{X_{t+1},...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T, X_1 = j_{t-1},..., X_t = j_0, X_{t+1},..., X_T) \log \pi_{t|1,...,t-1}(X_1 = j_{t-1},..., X_t = j_0)$$

$$- \lambda \left[\sum_{X_t \in S(X)} \pi_{t|1,...,t-1}(X_1 = j_{t-1},..., X_{t-1} = j_1, X_t) - 1 \right], \tag{A.12}$$

then the partial derivative is

$$\frac{\partial Q}{\partial \pi_{t|1,...,t-1}(X_1 = j_{t-1},..., X_t = j_0)}$$

$$= \frac{1}{\pi_{t|1,\ldots,t-1}(X_1 = j_{t-1},\ldots,X_t = j_0)} \sum_{X_{t+1},\ldots,X_T \in S(X)} P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_t = j_0, X_{t+1},\ldots,X_T) - \lambda \ . \qquad \text{(A.13)}$$

Set Eq. (A.13) equal to zero, we have

$$\hat{\pi}_{t|1,\ldots,t-1}(X_1 = j_{t-1},\ldots,X_t = j_0) = \frac{\sum_{X_{t+1},\ldots,X_T \in S(X)} P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_t = j_0, X_{t+1},\ldots,X_T)}{\lambda} . \qquad \text{(A.14)}$$

Eq. (A.14) is subject to $\sum_{X_t \in S(X)} \hat{\pi}_{t|1,\ldots,t-1}(X_1 = j_{t-1},\ldots,X_{t-1} = j_1, X_t) = 1$, so it can be shown that

$$\lambda = \sum_{X_t,\ldots,X_T \in S(X)} P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_{t-1} = j_1, X_t,\ldots,X_T) . \qquad \text{(A.15)}$$

Therefore, the reestimation formula is

$$\hat{\pi}_{t|1,\ldots,t-1}(X_1 = j_{t-1},\ldots,X_t = j_0)$$

$$= \frac{\sum_{X_{t+1},\ldots,X_T \in S(X)} P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_t = j_0, X_{t+1},\ldots,X_T)}{\sum_{X_t,\ldots,X_T \in S(X)} P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_{t-1} = j_1, X_t,\ldots,X_T)}$$

$$= \frac{P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_t = j_0)}{P_\mu(Y_{-M+1},\ldots,Y_T, X_1 = j_{t-1},\ldots,X_{t-1} = j_1)}$$

$$= \frac{P_\mu(X_1 = j_{t-1},\ldots,X_t = j_0 \mid Y_{-M+1},\ldots,Y_T) P_\mu(Y_{-M+1},\ldots,Y_T)}{P_\mu(X_1 = j_{t-1},\ldots,X_{t-1} = j_1 \mid Y_{-M+1},\ldots,Y_T) P_\mu(Y_{-M+1},\ldots,Y_T)}$$

$$= \frac{\gamma_t(j_{t-1},\ldots,j_0)}{\gamma_{t-1}(j_{t-1},\ldots,j_1)} . \qquad \text{(A.16)}$$

In Eq. (A.7), the part of the equation that contains $a(X_{t-l},\ldots,X_t)$

is $\sum_{X_1,\ldots,X_T \in S(X)} P_\mu(Y_{-M+1},\ldots,Y_T, X_1,\ldots,X_T) \sum_{t=l+1}^{T} \log a_{X_{t-l},\ldots,X_t}$. $\qquad \text{(A.17)}$

Each term in the summation of Eq. (A.17) can be transformed to

$$P_\mu(Y_{-M+1},\ldots,Y_T, X_1,\ldots,X_T) \sum_{j_{l-1},\ldots,j_0,j} n_{j_{l-1}\ldots,j_0,j}(X_1,\ldots,X_T) \log a(j_{l-1}\ldots,j_0,j) , \qquad \text{(A.18)}$$

where $n_{j_{l-1}...,j_0,j}(X_1,...,X_T)$ is the number of times $j_{l-1},...,j_0,j$ observed in the sequence $X_1,...,X_T$.
To estimate $a(j_{l-1}...,j_0,j)$, we rewrite Eq. (A.7) as

$$Q = ... + \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T) \log a_{j_{l-1}...,j_0,j} - \lambda\left[\sum_j a_{j_{l-1}...,j_0,j} - 1\right], \quad \text{(A.19)}$$

according to the transformation in Eq. (A.18). The partial derivative with respect to $a(j_{l-1}...,j_0,j)$ is given as

$$\frac{\partial Q}{\partial a(j_{l-1}...,j_0,j)}$$

$$= \frac{1}{a(j_{l-1}...,j_0,j)} \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T) - \lambda \ . \quad \text{(A.20)}$$

Set Eq. (A.20) equal to zero, we get

$$\hat{a}(j_{l-1}...,j_0,j) = \frac{\sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T)}{\lambda}. \quad \text{(A.21)}$$

Since $\sum_j \hat{a}(j_{l-1}...,j_0,j) = 1$, it is clear that

$$\lambda = \sum_j \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T). \quad \text{(A.22)}$$

Thus,

$$\hat{a}(j_{l-1}...,j_0,j) = \frac{\sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T)}{\sum_j \sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T)}. \quad \text{(A.23)}$$

It can be proved that

$$\sum_{X_1,...,X_T \in S(X)} P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T) n_{j_{l-1}...,j_0,j}(X_1,...,X_T)$$

$$= \sum_{t=l}^{T-1} P_\mu(Y_{-M+1},...,Y_T,X_{t-l+1} = j_{l-1},...,X_t = j_0,X_{t+1} = j), \quad \text{(A.24)}$$

so Eq. (A.23) can be written as

$$\hat{a}(j_{l-1}...,j_0,j)=\frac{\sum_{t=l}^{T-1}P_\mu(Y_{-M+1},...,Y_T,X_{t-l+1}=j_{l-1},...,X_t=j_0,X_{t+1}=j)}{\sum_{t=l}^{T-1}P_\mu(Y_{-M+1},...,Y_T,X_{t-l+1}=j_{l-1},...,X_t=j_0)}$$

$$=\frac{\sum_{t=l}^{T-1}P_\mu(X_{t-l+1}=j_{l-1},...,X_t=j_0,X_{t+1}=j\mid Y_{-M+1},...,Y_T)P_\mu(Y_{-M+1},...,Y_T)}{\sum_{t=l}^{T-1}P_\mu(X_{t-l+1}=j_{l-1},...,X_t=j_0\mid Y_{-M+1},...,Y_T)P_\mu(Y_{-M+1},...,Y_T)}$$

$$=\frac{\sum_{t=l}^{T-1}\varepsilon_t(j_{l-1},...,j_0,j)}{\sum_{t=l}^{T-1}\gamma_t(j_{l-1},...,j_0)}. \qquad \text{(A.25)}$$

Similar to Eq. (A.18), $P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)\sum_{t=1}^{T}\log b^{X_t}_{y_{-X_t+t},...,y_t}$ in Eq. (A.7) can be transformed to

$$P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)\sum_{j_0}\sum_{i_{j_0},...,i_0}n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)\log b^{j_0}_{i_{j_0},...,i_0}, \qquad \text{(A.26)}$$

where $n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)$ is the frequency of observing j_0 at X_t and $i_{j_0},...,i_0$ at $Y_{t-j_0},...,Y_t$.

According to Eq. (A.26), Eq. (A.7) can be written as

$$Q=...+\sum_{X_1,...,X_T\in S(X)}P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)\log b^{j_0}_{i_{j_0},...,i_0}$$
$$-\lambda\left[\sum_{i_0}\log b^{j_0}_{i_{j_0},...,i_0}-1\right]. \qquad \text{(A.27)}$$

Taking derivative with respect to $b^{j_0}_{i_{j_0},...,i_0}$ and set it equal to zero, we

$$\text{get}\,\hat{b}^{j_0}_{i_{j_0},...,i_0}=\frac{\sum_{X_1,...,X_T\in S(X)}P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)}{\lambda}.$$

(A.28)

It can be shown that

$$\lambda=\sum_{i_0}\sum_{X_1,...,X_T\in S(X)}P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T). \qquad \text{(A.29)}$$

Substituting Eq. (A.29) into Eq. (A.28), we have

$$\hat{b}^{j_0}_{i_{j_0},...,i_0}=\frac{\sum_{X_1,...,X_T\in S(X)}P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)}{\sum_{i_0}\sum_{X_1,...,X_T\in S(X)}P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)}. \qquad \text{(A.30)}$$

Since $\sum_{X_1,...,X_T\in S(X)}P_\mu(Y_{-M+1},...,Y_T,X_1,...,X_T)n^{j_0}_{i_{j_0},...,i_0}(Y_{-M+1},...,Y_T,X_1,...,X_T)$ is equivalent to

$$\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} P(Y_{-M+1},\ldots,Y_T,X_t=j_0)\,. \tag{A.31}$$

Eq. (A.30) can be written as

$$\begin{aligned}
\hat{b}_{i_{j_0},\ldots,i_0}^{j_0} &= \frac{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} P(Y_{-M+1},\ldots,Y_T,X_t=j_0)}{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_{t-1}=i_1}^{T} P(Y_{-M+1},\ldots,Y_T,X_t=j_0)} \\
&= \frac{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} P(Y_{-M+1},\ldots,Y_T,X_{t-j+1}=j_{l-1},\ldots,X_{t-1}=j_1,X_t=j_0)}{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_{t-1}=i_1}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} P(Y_{-M+1},\ldots,Y_T,X_{t-j+1}=j_{l-1},\ldots,X_{t-1}=j_1,X_t=j_0)} \\
&= \frac{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_t=i_0}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} \gamma_t(j_{l-1},\ldots,j_0)}{\sum_{t=1\,:\,Y_{t-j_0}=i_{j_0},\ldots,Y_{t-1}=i_1}^{T} \sum_{j_{l-1}\in S(X)} \cdots \sum_{j_1\in S(X)} \gamma_t(j_{l-1},\ldots,j_0)}\,.
\end{aligned} \tag{A.32}$$

Appendix IV: Scaling of the Forward and Backward Variables

According to the definitions, both forward and backward variables are product of a large number of probabilities, each of which is significantly less than 1. Computationally, both variables will become too small such that the computer does not have the precision to handle. One way to avoid this problem is to scale the forward and backward variables at each step *t*. For $t = 1,$ the scaled forward variable is defined as

$$\tilde{\alpha}_1(j_0) = \frac{\alpha_1(j_0)}{\bar{\alpha}_1}, \tag{B.1}$$

where $\bar{\alpha}_1 = \dfrac{\sum_{j_0=0}^{M} \alpha_1(j_0)}{M+1}$. For $t = 2,$

$$\tilde{\alpha}_2(j_1, j_0) = \frac{b^{j_0}_{y_{-j_0+2},\ldots,y_2} \pi_{2|1}(j_1, j_0)\tilde{\alpha}_1(j_1)}{\bar{\alpha}_2}, \tag{B.2}$$

where $\bar{\alpha}_2 = \dfrac{\sum_{j_1, j_0=0}^{M} b^{j_0}_{y_{-j_0+2},\ldots,y_2} \pi_{2|1}(j_1, j_0)\tilde{\alpha}_1(j_1)}{(M+1)^2}$. Substituting Eq. (B.1) into Eq. (B.2), we get

$$\tilde{\alpha}_2(j_1, j_0) = \frac{b^{j_0}_{y_{-j_0+2},\ldots,y_2} \pi_{2|1}(j_1, j_0)\alpha_1(j_1)}{\bar{\alpha}_1\bar{\alpha}_2}$$

$$= \frac{\alpha_2(j_1, j_0)}{\bar{\alpha}_1\bar{\alpha}_2}. \tag{B.3}$$

Similarly, we can show that

$$\tilde{\alpha}_t(j_{t-1},\ldots, j_0) = \frac{\alpha_t(j_{t-1},\ldots, j_0)}{\bar{\alpha}_1,\ldots,\bar{\alpha}_t} \tag{B.4}$$

for $t = 3,\ldots,l$ and

$$\tilde{\alpha}_t(j_{l-1},\ldots, j_0) = \frac{\alpha_t(j_{l-1},\ldots, j_0)}{\bar{\alpha}_1,\ldots,\bar{\alpha}_t} \tag{B.5}$$

for $t = l+1,...,T$.

For $t = T$, the scaled backward variable is defined as

$$\tilde{\beta}_T(j_{l-1},...,j_0) = \frac{\beta_T(j_{l-1},...,j_0)}{\overline{\beta}_T}, \tag{B.6}$$

where $\overline{\beta}_T = \frac{\sum_{j_{l-1},...,j_0=0}^{M} \beta_T(j_{l-1},...,j_0)}{(M+1)^l}$. For $t = T-1$,

$$\tilde{\beta}_{T-1}(j_{l-1},...,j_0) = \frac{\sum_{j=0}^{M} a_{j_{l-1},...j_0,j} b^j_{y_{t-j+1},...,y_{t+1}} \tilde{\beta}_T(j_{l-2},...,j_0,j)}{\overline{\beta}_{T-1}}, \tag{B.7}$$

where $\overline{\beta}_{T-1} = \frac{\sum_{j_{l-1},...j_0}^{M} \sum_{j=0}^{M} a_{j_{l-1},...j_0,j} b^j_{y_{t-j+1},...,y_{t+1}} \tilde{\beta}_T(j_{l-2},...,j_0,j)}{(M+1)^l}$. Substituting Eq. (B.6) into Eq. (B.7), we get

$$\tilde{\beta}_{T-1}(j_{l-1},...,j_0) = \frac{\sum_{j=0}^{M} a_{j_{l-1},...j_0,j} b^j_{y_{t-j+1},...,y_{t+1}} \beta_T(j_{l-2},...,j_0,j)}{\overline{\beta}_{T-1}\overline{\beta}_T}$$

$$= \frac{\beta_{T-1}(j_{l-1},...,j_0)}{\overline{\beta}_{T-1}\overline{\beta}_T}. \tag{B.8}$$

Similarly, it can be shown that for $t = T-1,...,l$

$$\tilde{\beta}_t(j_{l-1},...,j_0) = \frac{\beta_t(j_{l-1},...,j_0)}{\overline{\beta}_t,...,\overline{\beta}_T}, \tag{B.9}$$

and for $t = 1,...,l-1$

$$\tilde{\beta}_t(j_{t-1},...,j_0) = \frac{\beta_t(j_{t-1},...,j_0)}{\overline{\beta}_t,...,\overline{\beta}_T}. \tag{B.10}$$

Therefore, the likelihood in terms of the scaled variables is

$$L = \sum_{j_{l-1},...,j_0 \in S(X)} \tilde{\alpha}_t(j_{l-1},...,j_0)\tilde{\beta}_t(j_{l-1},...,j_0)\prod_{i=1}^{t}\overline{\alpha}_i\prod_{i=t}^{T}\overline{\beta}_i \tag{B.11}$$

for $t = l+1,...,T$. Likewise, we can rewrite ε_t and γ_t using the scaled forward and backward variables.

Appendix V: R Program for MMMHS

```
#####################################################

# Sequence based starting values for C matrix
#k equals to vmax.order+1

C.starting.value.seqbased<-function(data, k, h)
{
n<-length(data)
c.vector<-rep(0, sum(h^(1:k)))
for (i in 1:k)
{
  multiplier<-c(h^((i-1):0))
  for (j in 1:(n-i+1))
  {
    s.ind<-t(data[j:(j+i-1)])%*%multiplier
    c.vector[s.ind]<-c.vector[s.ind]+1
  }
}
c.mat<-apply(matrix(c.vector, nrow=h, byrow=FALSE), 2, function(x) x/sum(x))
c.mat.nona<-ifelse(is.na(c.mat), 0, c.mat)
return(as.vector(c.mat.nona))
}

#EM starts
proc.em<-function(data, pi, A, C, h.order, h.state, v.state)
{

#################################################

#Function to generate C diagnal matrix at t th pos (largest t is T-h.state)

C.diagmat.gen<-function(data, t, h.s, v.s)
{
  real.t<-t+h.s-1
  s.value<-0
  pos<-rep(NA, h.s)
  for (i in 1:h.s)
  {
    pos[i]<-data[real.t-i+1]*v.s^(i-1)+s.value
    s.value<-pos[i]
  }
  return(diag(C[pos]))
}

#################################################

#5 Forward procedure

v.maxorder<-h.state-1
seq.length<-length(data)
T<-seq.length-v.maxorder
alpha.tilde<-array(NA, dim=c(h.state, h.state^(h.order-1), T))
alpha.bar<-rep(NA, T)
for (i in 1:T)
```

```
{
  if (i<=h.order)
  {
    n.col<-h.state^(i-1)
  } else {n.col<-h.state^(h.order-1)}

  if (i==1)
  {
    alpha.orig<-C.diagmat.gen(data, t=i, h.s=h.state, v.s=v.state)%*%pi[,1:n.col,i]
    alpha.bar[i]<-mean(alpha.orig)
    alpha.tilde[,1:n.col,i]<-alpha.orig/alpha.bar[i]
  }

  if (i>1 && i<h.order+1)
  {
    alpha.orig<-C.diagmat.gen(data, t=i, h.s=h.state,
v.s=v.state)%*%pi[,1:n.col,i]%*%diag(as.vector(t(alpha.tilde[,1:(n.col/h.state),i-1])))
    alpha.bar[i]<-mean(alpha.orig)
    alpha.tilde[,1:n.col,i]<-alpha.orig/alpha.bar[i]
  }

  if (i>=h.order+1)
  {
    alpha.orig<-C.diagmat.gen(data, t=i, h.s=h.state, v.s=v.state)%*%A%*%(kronecker(diag(1,ncol(A)/h.state),
as.vector(rep(1, h.state)))*as.vector(t(alpha.tilde[,,i-1])))
    alpha.bar[i]<-mean(alpha.orig)
    alpha.tilde[,1:n.col,i]<-alpha.orig/alpha.bar[i]
  }

}

#5 end

#####################################################

#6 Backward procedure

beta.tilde<-array(NA, dim=c(h.state, h.state^(h.order-1), T))
beta.bar<-rep(NA, T)
for (i in 1:T)
{
  j<-T+1-i
  if (j<=h.order-1)
  {
    n.col<-h.state^(j-1)
  } else {n.col<-h.state^(h.order-1)}

  if (j==T)
  {
    beta.orig<-1
    beta.bar[j]<-mean(beta.orig)
    beta.tilde[,1:n.col,j]<-beta.orig/beta.bar[j]
  }
  if (j>=h.order && j<T)
  {
```

```
   beta.orig<-matrix(t(rep(1, h.state))%*%(C.diagmat.gen(data, t=(j+1), h.s=h.state,
v.s=v.state)%*%A*kronecker(beta.tilde[,,j+1], t(rep(1, h.state)))), ncol=n.col, byrow=TRUE)
   beta.bar[j]<-mean(beta.orig)
   beta.tilde[,1:n.col,j]<-beta.orig/beta.bar[j]
  }
  if (j<h.order)
  {
   beta.orig<-matrix(t(rep(1, h.state))%*%(C.diagmat.gen(data, t=(j+1), h.s=h.state,
v.s=v.state)%*%(pi[,1:(n.col*h.state),j+1]*beta.tilde[,1:(n.col*h.state),j+1])), ncol=n.col, byrow=TRUE)
   beta.bar[j]<-mean(beta.orig)
   beta.tilde[,1:n.col,j]<-beta.orig/beta.bar[j]
  }
}

#6 end

#####################################################

#7 Logliklihood

log.liklihood<-rep(NA, T)
lik.mat<-alpha.tilde*beta.tilde
for (i in 1:T)
{
log.liklihood[i]<-log(sum(lik.mat[,,i][!is.na(lik.mat[,,i])]))+sum(log(alpha.bar[1:i]))+sum(log(beta.bar[i:T]))
}

#7 end
#####################################################

#8 Epsilon

eps<-array(NA, dim=c(h.state, h.state^h.order, T-1))
for (i in 1:(T-1))
{
  if (i<h.order)
  {
    n.col<-h.state^i
    eps[,1:n.col,i]<-(C.diagmat.gen(data, t=(i+1), h.s=h.state,
v.s=v.state)%*%pi[,1:n.col,i+1]%*%diag(as.vector(t(alpha.tilde[,1:(n.col/h.state),i]))))*beta.tilde[,1:n.col,i+1]*(exp(s
um(log(alpha.bar[1:i]))+sum(log(beta.bar[(i+1):T]))-log.liklihood[T]))
  }

  if (i>=h.order)
  {
    n.col<-h.state^h.order
    eps[,1:n.col,i]<-(C.diagmat.gen(data, t=(i+1), h.s=h.state,
v.s=v.state)%*%A%*%diag(as.vector(t(alpha.tilde[,1:(n.col/h.state),i]))))*kronecker(beta.tilde[,1:(n.col/h.state),i+1],
t(rep(1, h.state)))*(exp(sum(log(alpha.bar[1:i]))+sum(log(beta.bar[(i+1):T]))-log.liklihood[T]))
  }
}

#8 end

#####################################################
```

```
#9 Gamma

gamma<-array(NA, dim=c(h.state, h.state^(h.order-1), T))
for (i in 1:T)
{
  if (i<h.order)
  {
    n.col<-h.state^(i-1)
    gamma[,1:n.col,i]<-
alpha.tilde[,1:n.col,i]*beta.tilde[,1:n.col,i]*(exp(sum(log(alpha.bar[1:i]))+sum(log(beta.bar[i:T]))-log.liklihood[T]))
  }

  if (i>=h.order)
  {
    n.col<-h.state^(h.order-1)
    gamma[,1:n.col,i]<-
alpha.tilde[,1:n.col,i]*beta.tilde[,1:n.col,i]*(exp(sum(log(alpha.bar[1:i]))+sum(log(beta.bar[i:T]))-log.liklihood[T]))
  }
}


#9 end

#####################################################

#10 Pi reestimation

old.pi<-pi

for (i in 1:h.order)
{
  if (i==1)
  {
    pi[,,i]<-gamma[,,i]
  }
  if (i>1)
  {
    c.d<-h.state^(i-1)
    pi[,1:c.d,i]<-gamma[,1:c.d,i]/kronecker(t(as.vector(t(gamma[,1:(c.d/h.state), i-1]))), rep(1, h.state))
  }
}

#10 end

#####################################################

#11 A reestimation

old.A<-A

eps.sum<-apply(eps[,,h.order:(T-1)], c(1,2), sum)
if (h.order==1)
{
gamma.sum<-t(as.vector(t(apply(gamma[,,h.order:(T-1)], 1, sum))))
}
if (h.order>1)
```

```
{
gamma.sum<-t(as.vector(t(apply(gamma[,,h.order:(T-1)], c(1,2), sum))))
}
A<-eps.sum/kronecker(gamma.sum, rep(1, h.state))

#11 end

#####################################################

#12 C reestimation

C.nom<-as.vector(rep(0, sum(v.state^(1:h.state))))
C.denom<-as.vector(rep(0, sum(v.state^(1:h.state))))
for (i in 1:T)
{
  sq.ind<-0
  real.pos<-i+h.state-1
  for (j in 1:h.state)
  {
    sq.ind<-data[real.pos-j+1]*v.state^(j-1)+sq.ind
    C.nom[sq.ind]<-C.nom[sq.ind]+sum(gamma[j,,i][!is.na(gamma[j,,i])])
  }
}

for (i in 1:T)
{
  sq.ind<-0
  real.pos<-i+h.state-1
  for (j in 2:h.state)
  {
    sq.ind<-data[real.pos-j+1]*v.state^(j-2)+sq.ind
    C.denom[sq.ind]<-C.denom[sq.ind]+sum(gamma[j,,i][!is.na(gamma[j,,i])])
  }
}

C.den<-kronecker(c(sum(gamma[1,,][!is.na(gamma[1,,])]), C.denom[1:sum(v.state^(1:(h.state-1)))]), rep(1,
v.state))
C<-ifelse(is.na(C.nom/C.den), 0, C.nom/C.den)


#12 end

#####################################################

output<-list(pi=pi, A=A, C=C, log.lik=log.liklihood[T])

return(output)

}
#EM ends

#####################################################

#MMMHS program starts

MMMHS<-function(data, h.order, h.state, v.state, random.seed, csv.method, conv.criteria)
```

```
{

set.seed(random.seed)

# Hidden states are 0, 1,..., h.state-1.

v.maxorder<-h.state-1

#####################################################
#1 Generate starting values for Pi

pi<-array(NA, dim=c(h.state, h.state^(h.order-1), h.order))

#1.1 Transition probabilities generating function
prob.gen<-function(r.dim, c.dim)
{
  raw.num<-matrix(runif(r.dim*c.dim, 0, 1), nrow=r.dim, ncol=c.dim)
  end.num<-apply(raw.num, 2, function(x) x/sum(x))
  return(end.num)
}
#1.1 end

for (i in 1:h.order)
{
  c.d<-h.state^(i-1)
  pi[,1:c.d,i]<-prob.gen(r.dim=h.state, c.dim=c.d)
}
#1 end
#####################################################

#2 Generate starting values for A

A<-prob.gen(r.dim=h.state, c.dim=h.state^h.order)

#2 end

#####################################################

#3 Generate starting values for C

if (csv.method=="random")
{
  C<-as.vector(prob.gen(r.dim=v.state, c.dim=sum(v.state^(1:h.state))/v.state))
} else if (csv.method=="seqbased")
  {
    C<-C.starting.value.seqbased(data, k=h.state, h=v.state)
  } else {stop("csv.method must be 'random' or 'seqbased' ")}

#3 end

#####################################################

#4Function to generate C diagnal matrix at t th pos (largest t is T-h.state)

C.diagmat.gen<-function(data, t, h.s, v.s)
{
```

```
  real.t<-t+h.s-1
  s.value<-0
  pos<-rep(NA, h.s)
  for (i in 1:h.s)
  {
    pos[i]<-data[real.t-i+1]*v.s^(i-1)+s.value
    s.value<-pos[i]
  }
  return(diag(C[pos]))
}
#4 end

#####################################################

no.para<-h.state*sum(h.state^(0:(h.order-1)))+h.state*h.state^h.order+v.state*sum(v.state^(1:h.state))/v.state
em.out<-proc.em(data, pi=pi, A=A, C=C, h.order, h.state, v.state)

pi<-em.out$pi
A<-em.out$A
C<-em.out$C
old.loglik<-em.out$log.lik

seq.length<-length(data)
T<-seq.length-v.maxorder
diff.loglik<-10
iter<-1

while (diff.loglik>conv.criteria)
{
  s.time<-proc.time()
  em.out<-proc.em(data, pi=pi, A=A, C=C, h.order, h.state, v.state)
  pi<-em.out$pi
  A<-em.out$A
  C<-em.out$C
  diff.loglik<-em.out$log.lik-old.loglik
  old.loglik<-em.out$log.lik
  BIC<--2*em.out$log.lik+no.para*log(T)
  cat("Difference of Log Liklihood = ",diff.loglik, " || ", (proc.time()-s.time)[3], " second for No.", iter, "iteration.", "
BIC=", BIC, "\n")
  iter<-iter+1
}

final.output<-list(pi=pi, A=A, C=C, log.lik=em.out$log.lik, BIC)
return(final.output)

}

#MMMHS program ends
#####################################################

#csv.method--"seqbased" or "random".
#conv.criteria--0.000001
#h.order--hidden order >=1
#h.state--hidden state >=2
#v.state--visible state depends on data
```

MMMHS(data, h.order=1, h.state=2, v.state=3, random.seed=1, csv.method="seqbased", conv.criteria=0.000001)

References:

Almagor, H. (1983) A Markov analysis of DNA sequences. Journal of Theoretical Biology, 104, 633-645.

Attwood,T.K. (2000) Genomics: the Babel of bioinformatics. Science, 290, 471-473.

Avery, P.J. (1987) The analysis of intron data and their use in the detection of short signals. Journal of Molecular Evolution, 26, 335-340.

Baum, L. E. (1972) An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities, 3, 1-8.

Baum, L. E. and Egon, J. A. (1967) An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. Bull. Amer. Meteorol. Soc., 73, 360-363.

Baum, L. E. and Petrie, T. (1966) Statistical inference for probabilistic function of finite state Markov chains. Ann. Math. Stat., 37, 1554-1563, 1966.

Baum, L. E. and Sell, G. R. (1968) Growth functions for transformations on manifolds. Pac. /. Math., 27(2), 211-227.

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Stat., 41(1), 164-171.

Berchtold, A. (2001) Estimation in the Mixture Transition Distribution Model. J. Time Ser. Anal. , 22 (4), 379–397.

Berchtold, A. (1999) The Double Chain Markov Model. Commun. Stat.: Theory Meth. 28(11), 2569-2589.

Berchtold, A. (2002) High-order Extensions of the Double Chain Markov Model. Stochastic Models. 18(2), 193-227.

Bishop, Y.M.M., Fienberg, S.E., Holland, P.W. (1975) Discrete Multivariate Analysis. MIT press: Cambridge.

Bize, L., Muri, F., Samson, F., Rodolphe, F., Ehrlich, S. D., Prum, B., Bessières, p. (1999) Searching gene transfers on Bacillus subtilis using hidden Markov models. Proceedings of the third annual international conference on Computational molecular biology, 43-49.

Blaisdell, E. B. (1985) Markov chain analysis finds a significant influence of neighboring bases on the occurrence of a base in eukaryotic nuclear DNA sequences both protein-coding and non-coding. Journal of Molecular Evolution, 21, 278-288.

Blaisdell,B.E. (1986) A measure of the similarity of sets of sequences not requiring sequence alignment. Proc. Natl Acad. Sci. USA, 83, 5155–5159.

Brendel, V, .Beckmann, J. S. and Trifonov, E. N. (1986) Linguistics of nucleotide sequences: Morphology and comparison of vocabularies. J Biomol Struct Dyn, 4, 11–21.

Burke, J., Davison, D. and Hide, W. (1999) d2_cluster: a validated method for clustering EST and full-length cDNA sequences. Genome Res., 9, 1135-1142.

Burke, J., Wang, H., Hide, W. and Davison, D. B. (1998) Alternative gene form discovery and candidate gene selection from gene indexing projects. Genome Res., 8, 276-290.

Cao,Y., Janke,A., Waddell,P.J., Westerman,M., Takenaka,O., Murata,S., Okada,N., Paabo,S. and Hasegawa,M. (1998) Conflict among individual mitochondrial proteins in resolving the phylogeny of eutherian orders. J Mol Evol, 47, 307–322.

Cao,Y., Okada,N. and Hasegawa,M. (1997) Phylogenetic position of guinea pigs revisited. Mol Biol Evol, 14, 461–464.

Chatfield, C. and Lemon, R.E. (1970) Analysing Sequences of Behavioral Events. J. Theor. Biol., 29, 427–445.

Churchill, A. G. (1992) Hidden Markov models and the analysis of genome structure. Computers and Chemistry, 16, 107-115.

Churchill,A.G. (1989) Stochastic models for heterogeneous DNA sequences. Bulletin of Mathematical Biology, 51, 79-94.

Cogburn, R. (1984) The ergodic theory of Markov chains in random environments. Z. Wahrscheinlichkeitsttheories Verw. Gebiete, 66, 109-128.

Craig, W. (1943) The Song of the Wood Peewee; University of the State of New York: Albany.

D'Erchia,A.M., Gissi,C., Pesole,G., Saccone,C. and Árnason,Ú. (1996) The guinea-pig is not a rodent. Nature, 381, 597-600.

Davison, D. B. and Burke, J. (2001) Brute force estimation of the number of human gene using EST clustering as a measure. IBM J. Res. Dev., 45, 439-447.

Dempster, P. A., Laird, M. N. and Rubin, B. D. (1977) Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc., 39(1), 1-38.

Edgar, Robert, C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research, 32(5), 1792-97.

Elton, R. A. (1974) Theoretical models for heterogeneity of base composition in DNA. J. Theor. Biol.

45, 533-553.

Felsenstein, J. and Churchill, G. (1996) A hidden Markov model approach to variation among sites in rate of evolution. Mol Biol Evol, 13, 93-104.

Felsenstein,J. (1993) PHYLIP (Phylogeny Inference Package), version 3.6.4. Distributed by the Author. Department of Genetics, University of Washington, Seattle, WA

Fickett, J. W. and Tung, C-S. (1992) Assessment of protein coding measures. Nucleic Acids Research, 20, 6441-6450.

Forney, G. D. (1973) The Viterbi Algorithm. Proceedings of the IEEE, 61, 268-278.

Gentleman, F. J. and Mullin, C. R. (1989) The distribution of the frequency of occurrence of nucleotide subsequences, based on their overlap capability. Biometrics, 45(1), 35-52.

Gibbs,A.J., Dale,M.B., Kinns,H.R. and MacKenzie,H.G. (1971) The transition matrix method for comparing sequences; its use in describing and classifying proteins by their amino acids sequence. Systematic Zool., 20, 417–425.

Hao,B. and Qi, J. (2004) Prokaryote phylogeny without sequence alignment: from avoidance signature to composition distance. J Bioinform Comput Biol, 2, 1-19.

Haslett, J. and Raftery, A.E. (1989) Space-time modeling with long-memory dependence: Assessing ireland's wind power resource. Applied Statistics, 38(1):1-50.

Helden,V.J. (2004) Metrics for comparing regulatory sequences on the basis of pattern counts. Bioinformatics, 20, 399–406.

Hide, W., Burke, J. and Davison, D. B. (1994) Biological evaluation of d2, an algorithm for high-performance sequence comparison. J. Comput. Biol., 1, 199-215.

Kantorovitz,R,M,, Robinson, E.G. and Sinha,S. (2007) A statistical method for alignment-free comparison of regulatory sequences. Bioinformatics, 23, i249-i255.

Li,J. and Sayood,K. (2005) A genome signature based on Markov modeling. Engineering in Medicine and Biology Society, 27th Annual Conference of IEEE, IEEE-EMBS, 2832-2835.

Li,M., Badger,J.H., Chen,X., Kwong,S., Kearney,P. and Zhang,H. (2001) An information-based sequence distance and its application to whole mitochondrial genome phylogeny. Bioinformatics, 17, 149–154.

Li,W., Fang,W., Ling,L., Wang,.J, Xuan,Z. and Chen,R. (2002) Phylogeny based on whole genome as inferred from complete information set analysis. Journal of Biological Physics, 28, 439-447.

Lu, G., Rowley, T., Garten, R., Donis, R. (2007) FluGenome: a web tool for genotyping influenza A

virus. Nucleic Acids Research, 35(Suppl 2),W275-W279 doi: 10.1093/nar/gkm365

Lu,G., Zhang,S. and Fang,X. (2008) An improved string composition method for sequence comparison. BMC Bioinformatics, 9 (Suppl 6), S15.

Miller, R. T., Christoffels, A. G., Gopalakrishnan, C., Burke, J., Ptitsyn, A. A., Broveak, T. R. and Hide, W. A. (1999) A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base. Genome Res., 9, 1143-1155.

Moriyama, E., Kim, J. (2005) Protein family classification with discriminant function analysis. Genome Exploitation: Data Mining the Genome. Springer, New York, 121-132.

Muri, F. (1998) Modelling bacterial genomes using hidden Markov models. Compstat'98 Proceedings in Computational statistics. R. Payne and P. Green (Ed. Physics-Verlag.). 89-100.

Novacek, M. (1992) Mammalian phylogeny: shaking the tree. Nature 356:121–125.

Otu,H.H. and Sayood,K. (2003) A new sequence distance measure for phylogenetic tree construction. Bioinformatics, 19, 2122-2130.

Pearson,W.R. (2000) Protein sequence comparison and protein evolution. Tutorial-ISMB2000.

Petrilli, P. (1993) Classification of protein sequences by their dipeptide composition. Comput. Appl. Biosci., 9, 205-209.

Petrilli, P. and Tonukari, N. J. (1997) PFDB: a protein families database for Macintosh computers. The effectiveness of its organization in searching for protein similarity. J. Protein Chem., 16, 713-720.

Pham,D.T. and Zuegg,J. (2004) A probabilistic measure for alignment-free sequence comparison. Bioinformatics, 20, 3455-3461.

Phillips, G. J., Arnold, J. and Ivarie, R. (1987) Mono-through hexanucleotide composition of the Escherichia coli genome: a Markov chain analysis. Nucleic Acids Research, 15, 2611-2626.

Qi,J., Wang,B. and Hao,B. (2004) Whole proteome prokaryote phylogeny without sequence alignment: A K-string composition approach.. J Mol Evol, 58, 1-11.

Rabiner, R. L. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77(2), 257-286.

Raftery, A.E and Tavare. S. (1994) Estimation and Modelling Repeated Patterns in High Order Markov Chains with the Mixture Transition Distribution Model. Applied Statistics 43(1), 179-199.

Reyes,A., Gissi,C., Pesole,G., Catzeflis,F.M. and Saccone,C. (2000) Where do rodents fit? Evidence from the complete mitochondrial genome of Sciurus vulgaris. Mol Biol Evol, 17, 979–983.

Scherer, S., Mcpeek, M. S. and Speed, T. P. (1994) A typical regions in large genomic DNA sequences. Proceedings of the National Academy of Sciences USA, 91, 7134-7138.

Staden, R. (1984) Graphic methods to determine the function of nucleic acid sequences. Nucl. Acids Res. 12, 521-538.

Strope, P., Moriyama, E. (2007) Simple alignment-free methods for protein classification: a case study from G-protein-coupled receptors. Genomics, 89(5), 602-612.

Stuart,G., Moffet,K. and Baker,S. (2002a) Integrated gene and species phylogenies from unaligned whole genome sequence. Bioinformatics, 18, 100–108.

Stuart,G., Moffet,K. and Leader,J. (2002b) A comprehensive vertebrate phylogeny using vector representation of protein sequences from whole genomes. Mol Biol Evol, 19, 554–562.

Sueeka, N. (1959) A statistical analysis of deoxyribonucleic acid distribution in density gradient centrifugation. Proc. Natn. Acad. Sci. U.S.A. 45, 1480-1490.

Tamura, K., Dudley, J., Nei, M. and Kumar, S. (2007) MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) software version 4.0. Mol Biol Evol, 24, 1596-1599

Torney, D. C., Burks, C., Davison, D. and Sirotkin, K. M. (1990) Computation of d2: a measure of sequence dissimilarity. In George, I. and Bell, T. G. M. (eds), Computers and DNA: the proceedings of the Interface between Computation Science and Nucleic Acid Sequencing Workshop, held December 12 to 16, 1988 in Santa Fe.

Vinga,S. and Almeida,J. (2003) Alignment free sequence comparison-a review. Bioinformatics 2003, 19, 513-523.

Viterbi, A. J. (1967) Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Trans. Informat. Theory, IT-13, 260-269.

WHO/OIE/FAO H5N1 Evolution Working Group.(2008) Toward a unified nomenclature system for highly pathogenic avian influenza virus (H5N1). Emerg Infect Dis, 14(7), p. e1.

Wiens,J.J. and Servedio,M.R. (1998) Phylogenetic analysis and intraspecific variation: performance of parsimony, likelihood, and distance methods. Syst Biol, 47, 228-53.

Wu,T.J., Burke,J.P. and Davison,D.B. (1997) A measure of DNA sequence dissimilarity based on Mahalanobis distance between frequencies of words. Biometrics, 53, 1431–1439.

Wu,T.J., Hsieh,Y.C. and Li,L.A. (2001) Statistical measures of DNA dissimilarity under Markov chain models of base composition. Biometrics,, 57, 441–448.

Wu,X., Cai,Z., Wan,X., Hoang,T., Goebel,R. and Lin,G. (2007) Nucleotide composition string selection in HIV-1 subtyping using whole genomes. Bioinformatics, 23, 1744-1752.

Wu,X., Wan,X., Wu,G., Xu,D. and Lin,G. (2006) Phylogenetic analysis using complete signature information of whole genomes and clustered Neighbour-Joining method. Int J Bioinform Res Appl, 2, 219-248.

Zharkikh, A. A. and Rzhetsky, A. (1993) Quick assessment of similarity of two sequences by comparison of their L-tuple frequencies. Biosystems, 30, 93-111.