Jumpchain simulations to study the effects of CpG hypermutability on inferences of positive selection

by

Kassandra Dickson

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Biology Specialization in Bioinformatics

Department of Biology
Faculty of Science
Carleton University

© Kassandra Dickson, Ottawa, Canada, 2021

Abstract

Detecting signals of positive selection in protein-coding DNA sequences is of central importance in the study of molecular evolution. The statistical methods available for such a task are based on a suite of simplifying assumptions, many of which are known to be violated in real data. However, the extent to which violations impact inferences of positive selection remains poorly characterized. One main hurdle in the study of this problem is the difficulty in simulating artificial data, which would incorporate some of the complexities assumed absent by inference systems. In this work, we explore the use of an under-exploited simulation approach that allows the realization of normally intractable continuous-time Markov chains. The method, known as *jump-chain* simulation, can be used to produce artificial data with sophisticated evolutionary models, such as those that relax the assumption of independence between the positions of a gene. We use this simulation method to study a context-dependent mutational feature, highly prevalent in mammalian genomes. Referred to as *CpG hypermutability*, this feature has the potential to mislead widely used statistical models for detecting positive selection. Indeed, our analyses suggest that when CpG hypermutability is present at average intensities, traditional tests are misled into the conclusion of the presence of positive selection. Altogether, the present work emphasizes the usefulness of jump-chain simulation to improving our understanding of the limitations of current inference tools, while helping to prioritize future methodological developments.

Acknowledgements

I would like to thank Dr. Nicolas Rodrigue, for mentoring me throughout my thesis and my work towards this degree and my undergraduate degree. I would also like to thank Thomas Bujaki for always helping me with coding problems and always asking if I needed help. I would like to also thank, Simon Laurin-Lemay, for the simulation software provided for this research project.

Table of Contents

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Fundamentals of phylogenetics	1
1.1.1 Nucleotide and amino acid substitution models	2
1.1.2 Codon substitution models	5
1.2 CpG Hypermutability	7
1.3 Objectives of thesis	8
2 Methods	10
2.1 Codon evolution with CpG hypermutability	10
2.1.1 Substitution rate specification	11
2.1.2 Parameter values for simulations	12
2.2 Jump-chain simulation of Markov processes	13
2.2.1 Sampling the root state	14
2.2.2 Sampling dwell times and substitutions	14

2.2.3	Software	15
2.3	Tests of adaptive evolution	15
3	Results	17
3.1	Model fitting	17
3.1.1	Effects of CpG hypermutability on global ω parameter	17
3.1.2	Effects of CpG hypermutability on site-models	18
3.2	Sites with strong evidence of positive selection	20
4	Discussion and future directions	26
4.1	Model violations	26
4.1.1	CpG hypermutability and site-models	26
4.1.2	Other model violations	27
4.2	Jump-chain simulation for prospecting more realistic models	27
	APPENDICES	29
A	Producing and Analysis of Control File	30
A.1	Control Files	30
A.2	Producing the Control File Needed for CODEML	30
A.3	CODEML Analysis	32
A.4	Analysis Code for Likelihood Ratio Test	33
A.5	Analysis for Bayes Test	34
	References	36

List of Tables

3.1	VWA9 Positive Selection	20
3.2	BACE Positive Selection	21
3.3	FLT3 Positive Selection	22
3.5	VWA9 Bayes Analysis	23
3.7	BACE1 Bayes Analysis	24
3.9	FLT3 Bayes Analysis	25

List of Figures

1.1	Codon Chart	3
3.1	Distribution of inferred ω values under the M0 model over 100 simulations. The blue density curve represents the frequency of ω when the $\lambda_{CpG} = 1.0$, the pink density curve shows the frequency of ω when the $\lambda_{CpG} = 5.0$, and finally the green density curve shows the frequency of ω when the $\lambda_{CpG} =$ 10.0. The red dashed line indicates the true value of the ω	18
3.2	VWA9 Positive Selection Plot	19

Chapter 1

Introduction

1.1 Fundamentals of phylogenetics

It has long been a collective objective for many biologists to reconstruct the evolutionary history of all organisms on Earth and to express it in the form of a phylogenetic tree (Nei and Kumar, 2000). Classical methods deal mainly with physical or morphological features. However, morphological features are not readily discretized into character states. Modern methods makes use of the information from molecular data, mainly from DNA and protein sequences. Given that DNA is made of four nucleotides, adenine (A), thymine (T), cytosine (C), and guanine (G), it becomes much easier to contrast the difference between species. Such data lends itself more readily to the development of mathematical models of the evolutionary process, in turn used for phylogenetics inference. This approach also benefits from the vast body of statistical theory, enabling the quantification of support for one phylogeny over another through probability calculations.

In the next subsections, we describe the basic models of nucleotide substitution, and briefly describe the calculation of the *likelihood function*. We then describe the commonly used codon models of substitution, which recognize the protein-coding nature of (some) DNA.

1.1.1 Nucleotide and amino acid substitution models

Most models of molecular evolution are Markovian processes. They describe a rate of change from one state (usually a nucleotide, amino acid, or codon) to another, and these rates depend only on the current state of the process; a feature otherwise known as a first-order Markov process. It is important to note that with the common models, independence is assumed between the different character positions of the alignments. In other words, substitution rates at one position are not influenced by the states at other positions.

Starting with the most basic and simplest models to infer evolutionary relationships, a simple Poisson process was described by Jukes and Cantor (1969) (JC). This model treats all states the same way, with equal rates between all. However, with this simplistic JC model, there were many downfalls and this gave rise to many more models that aimed to better reflect properties of real data. The most obvious downfall of the JC model is that it assumes equal frequency of all states. It is well known, for instance, that some species have GC-rich genomes. As another example at the amino acid level, the JC model assumes that a protein has equal amounts of the twenty amino acids, whereas, say, tryptophan is generally much more rare than serine.

In 1981, Felsenstein (1981) proposed a model that explicitly introduces parameters controlling the relative frequency of states in the Markov process. This model, eventually known as F81, still assumes even exchangeability between states. The assumption is obviously violated. At the nucleotide level, there would be more changes between states with similar chemical structure such as a change between a purine and another purine or a pyrimidine and another pyrimidine (changes referred to as *transitions*). It is less common for a pyrimidine to mutate to a purine or vice versa (changes referred to as *transversions*). This is further reinforced by the fact that the genetic code itself (1.1) allows for more transitions than transversions without amino acid replacements Kimura (1980). An amino acid-level F81 model also has obvious violations. Many amino acids are readily exchanged

First Position	Second Position				Third Position
	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	STOP	STOP	A
	Leu	Ser	STOP	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Gln	Gly	A
	Val	Ala	Gln	Gly	G

Figure 1.1: The universal codon chart which is used to decipher mRNA sequences into amino acids.

with one another (e.g., L/I/V) given their similar biophysical features, whereas others differ too dramatically to exchange very often. The basic idea behind this is that replacing an amino acid with a similar one is likely to leave the overall protein functional (and hence can happen over evolutionary time without being purified away by selective forces), whereas radical amino acid replacements typically disrupt the protein's structure or function (and hence, such replacements are almost never observed, being purified away by selection).

The model proposed by Hasegawa et al. (1985) was the first to jointly recognize uneven state composition and uneven state exchangeability. Proposed as a nucleotide-level model, it introduces one more parameter than the F81 model, acting as a rate multiplier for transitions (leaving transversions rates as specified under F81); given that transitions are far more frequent than transversions, the new parameter is almost universally found to be greater than 1. Amino acid models recognizing uneven exchangeabilities between states

were also proposed shortly thereafter, with the first Dayhoff-inspired model of Jones et al. (1992).

The general time-reversible model, or GTR (Tavaré, 1986), allows for different exchangeability parameters for all pairs of states, as well as frequency parameters for all states. The GTR model is often written in matrix form as shown below.

$$Q = \begin{bmatrix} -(\alpha\pi_C + \beta\pi_G + \gamma\pi_T) & \alpha\pi_C & \beta\pi_G & \gamma\pi_T \\ \alpha\pi_A & -(\alpha\pi_A + \delta\pi_G + \epsilon\pi_T) & \delta\pi_G & \epsilon\pi_T \\ \beta\pi_A & \delta\pi_C & -(\beta\pi_A + \delta\pi_C + \eta\pi_T) & \eta\pi_T \\ \gamma\pi_A & \epsilon\pi_C & \eta\pi_G & -(\gamma\pi_A + \epsilon\pi_C + \eta\pi_G) \end{bmatrix} \quad (1.1)$$

Here, α , β , γ , δ , ϵ , and η represent the nucleotide exchangeabilities, whereas π_A , π_C , π_G , and π_T are the frequencies. The name of the model refers to the fact that it is the most general possible version of time-reversible models. This latter property of time-reversibility is a short-hand for the fact that this class of Markov processes involves probability calculations that are invariant to the orientation of the nucleotide substitution or amino acid replacement process. For simplicity, suppose a two species “phylogeny”; this amounts to a single branch (of length λ) connecting sequences of DNA or amino acids. Also suppose that at one site the state of one of the sequences is A, while the other sequence is in state T at this site. The probability of the data at this site is computed as:

$$\begin{aligned} p(D \mid \theta) &= p(A \mid \theta)p(T \mid A, \theta) \\ &= \pi_A [e^{\lambda Q}]_{AT} \end{aligned} \quad (1.2)$$

where θ used to collectively denote all parameters of the model and the Q denotes the substitution rate matrix. This last equation reads as the probability of the starting state of the process, here chosen to be A, times the probability of ending the process in

state T when running it along the branch length λ . The feature of time-reversible Markov processes is that one can equivalently compute the probability of the data as:

$$\begin{aligned} p(D \mid \theta) &= p(T \mid \theta)p(A \mid T, \theta) \\ &= \pi_T[e^{\lambda Q}]_{TA}. \end{aligned} \tag{1.3}$$

The right-most probability factor in these equations is computed using matrix exponentiation. It is important to realize that such a calculation amounts to an integral over all possible substitution histories that could lead from T to A, in the latter case, over the distance λ . We will later utilize this fact to sample explicit substitution histories.

While these models can modulate rates of substitution from one particular state to another, they still assume equivalent processes across the sites of the alignment. Different selective pressures acting on different sites, however, can lead to major differences in overall rates. The *gamma-distributed-rates* model was the first attempt at accommodating this possibility, using a random-variable approach that effectively makes branches “seem” longer (or shorter) for different sites (Yang, 1993, 1994). Numerous other extensions have also been proposed, often with the motivation of recognizing more of the features that could govern evolutionary processes.

1.1.2 Codon substitution models

For some time, phylogenetic analyses were either done at the nucleotide level or, working with the translated sequences, at the amino acid level. A codon substitution model attempts to take into account both mutational tendencies at the nucleotide level as well as selective pressures on amino acid replacements. It is constructed in a manner that is informed of the genetic code connecting nucleotide-triplet states to their respective translated amino acid states (disallowing stop codons in the nucleotide-triplet state space). The

main idea behind such developments was to contrast synonymous (silent) and nonsynonymous (amino acid altering) rates (Mayrose et al., 2007). The comparison of rates between synonymous and nonsynonymous substitutions gives insights into the mechanisms behind molecular sequence evolution (Yang et al., 2000). The ratio between nonsynonymous and synonymous rates can be abbreviated into $\omega = dN/dS$. This rate ratio is a significant indicator of selective pressure at the protein level. Under the canonical interpretation of the parameter, when $\omega = 1$, this indicates a neutral process; when $\omega < 1$, this indicates negative, or purifying selection at the amino acid level; and finally, when $\omega > 1$, this indicates positive, or adaptive selection Yang et al. (2000). A popular form of codon substitution model has entries in a 61 by 61 rate matrix (from codon i to codon j) as follows:

$$Q_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ differ at two or three nucleotide positions,} \\ \pi_j & \text{if } i \text{ and } j \text{ differ by one synonymous transversion,} \\ \kappa\pi_j & \text{if } i \text{ and } j \text{ differ by one synonymous transition,} \\ \omega\pi_j & \text{if } i \text{ and } j \text{ differ by one nonsynonymous transversion,} \\ \omega\kappa\pi_j & \text{if } i \text{ and } j \text{ differ by one nonsynonymous transition.} \end{cases} \quad (1.4)$$

In this model, κ is the transition/transversion rate ratio and the π_j is the equilibrium frequency, or stationary probability, of codon i . The latter is often calculated from the nucleotide frequencies at the three codon positions. Alternatively, a form inspired by the GTR-level nucleotide substitution has the rates proportional to the frequencies of the target nucleotide that is different in codon j , since i and j differ only at one codon position. For simplicity, we refer to the overall nucleotide GTR rate specification from codon i to j as

μ_{ij} , while also distinguishing nonsynonymous and synonymous event:

$$Q_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ differ at two or three nucleotide positions,} \\ \mu_{ij} & \text{if } i \text{ and } j \text{ differ by one synonymous step,} \\ \mu_{ij}\omega & \text{if } i \text{ and } j \text{ differ by one nonsynonymous step.} \end{cases} \quad (1.5)$$

A well studied extension of the models given in 1.4 and 1.5 is to consider different sites of the alignment as having been generated under a mixture of different ω values (Yang et al., 2000). They have become known as *site-models* in the literature, and have formed one of a few central engine to the elucidation of adaptive molecular evolution. Again, numerous other extensions exists (Anisimova and Kosiol, 2009), each, again, with specific applications in mind, but often with the aim of more faithfully capturing the complex features governing molecular evolution.

1.2 CpG Hypermotability

While most efforts at modeling across-site heterogeneity have been focused on capturing the varying characteristics and intensities of selection, the mutational process itself also varies across the genome (Maharjan and Ferenci, 2017; Krašovec et al., 2017). An example of this is in the case of a cytosine followed by a guanine (CpG) in the genome of mammals (and in vertebrates in general), which are far more mutable (Bird, 1980). The cytosine of CpG is often methylated as a 5-methylcytosine, which in turn mutates to a thymine through deamination (Suzuki et al., 2009).

In detecting natural selection at the amino acid sequence level by comparing the rates of nonsynonymous/synonymous substitution, the mutation process is assumed to be homogeneous across sites. However, in reality, the rates may not be the same if different proportions of synonymous and nonsynonymous events overlap with CpG sites; several

studies suggest that the phenomenon is widespread in mammals (Krawczak et al., 1998; Hellmann et al., 2003; Bird, 1999; Subramanian and Kumar, 2006). This particular model violation could thus lead to erroneous inferences. Indeed, Laurin-Lemay et al. (2018a) found that a test for uneven codon usage devised by Yang and Nielsen (2008) was misled by the effect of CpG hypermutability. Laurin-Lemay et al. (2018b) used a simulation method allowing for context-dependent mutation rates, in particular, with high rates in the CpG context, even if it overlaps codons. By producing artificial codon alignments under a model that assumes even codon usage, but elevated CpG mutation rates, and then subjecting these artificial data sets to the test by Yang and Nielsen, the authors showed how all of the uneven codon usage detected in mammals could be explained by the well-known feature of CpG hypermutability.

1.3 Objectives of thesis

While Laurin-Lemay et al. highlighted how CpG hypermutability could mislead a specialized model for detecting uneven codon usage, its effect on the more broadly used models based on ω remain unexplored. Here, we leverage the simulation techniques proposed by Laurin-Lemay et al. (2018b) to study how the CpG hypermutability model violation could impact inferences of adaptive evolution with common models of codon substitution. We first present a new model used to simulate, then describe the simulation procedure in detail. We then use the simulation method to generate different artificial data sets, with varying degrees of CpG hypermutability. Our hypothesis is that when CpG hypermutability is sufficiently pronounced classical site models that are used to detect adaption will lead to false positives. Therefore, the null hypothesis states that CpG hypermutability does not lead to false positives. Finally, we analyze these artificial data sets with CODEML (Yang, 2007), using a few different models for detecting adaptation based on findings where $\omega > 1$ (explained later in this document). Our results serve as a caution to those applying stan-

dard codon models with data sets having context-dependent mutational features, which, under certain conditions, could lead to very high rates of false-positives.

Chapter 2

Methods

2.1 Codon evolution with CpG hypermutability

Implementing a system that allows for inferences based on real data under a model that acknowledges CpG hypermutability, in the maximum likelihood or Bayesian framework, is a highly challenging problem. Because CpG contexts can span two adjacent codons, the process is not site-independent, leading to an intractable likelihood function, and the need for elaborate Monte Carlo approaches (Robinson et al., 2003; Rodrigue et al., 2005, 2006, 2009). However, as we describe in the later section, simulating artificial data under such a model is relatively trivial with the *jump-chain* approach. Such simulated data can provide important information about statistical methods utilized in molecular evolution, since they give full control of the data generating process. In particular, they allow us to test the effects of assumptions of commonly applied methods, when we know these assumptions are in fact violated by the data being analyzed. In this chapter we first describe the form of the model of interest, follow by an explanation of our method of simulating artificial data with it.

2.1.1 Substitution rate specification

The first step in our model construction is to envision the substitution process directly in the space of all possible sequences of a given length. We still focus on a point-mutation-based process, which means that rates are only assigned to events where the initial and final sequence differ only by one nucleotide; the rates between two sequences with multiple nucleotide differences are set to zero. For simplicity, one can still write the essential ideas of the model in a simple 61 by 61 rate matrix, but with the understanding that a site-dependent parameter is also invoked, which we write as λ_{CpG} , which requires knowledge of the states at adjacent codons. In other words, a CpG context can span two different codons, making the model site interdependent. With this parameter, the rate specified for a particular event at a particular site can change if the states at the adjacent sites change. For now, the λ_{CpG} parameter is only applied to C to T events (transitions) when this initial C is followed by a G in the sequence. For any given site at any given point along the tree, the rates between possible pairs of codons can be expressed as an extension of the rates given in (1.5) as follows:

$$Q_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ differ at more than one position,} \\ \mu_{ij} & \text{if synonymous, and non-CpG transition or transversion,} \\ \mu_{ij}\omega & \text{if nonsynonymous, and non-CpG transition, or transversion,} \\ \mu_{ij}\lambda_{CpG} & \text{synonymous, CpG transition,} \\ \mu_{ij}\omega\lambda_{CpG} & \text{nonsynonymous, CpG transition.} \end{cases} \quad (2.1)$$

We note that this amounts to a special case of the model described in Laurin-Lemay et al. (2018a), which was concerned with a mutation-selection formulation for a specialized application. As explained by Laurin-Lemay et al. (2018a), the model is time reversible, which does not mean the rates are equal in either direction but rather that the likelihood

function is invariant to the orientation of the Markov substitution process (as discussed and explained in equation 1.2 and 1.3).

2.1.2 Parameter values for simulations

With this formulation, we can control the extent of CpG hypermutability, by setting $\lambda_{CpG} > 1$, within a model very close to the form used in the CODEML software. We can also control the value of ω , along with nucleotide-level parameters governing nucleotide frequencies and exchangeabilities. For the latter sets of parameters, all of our simulations were based on one of three different sets of values, as obtained from three mammalian protein-coding alignments randomly chosen from the set of genes studied in Laurin-Lemay et al. (2018b); these genes have the designations *BACE1*, *FLT3*, and *VWA9*, which we use to refer to the nucleotide-level setting of our simulations. Working with three genes settings keeps the computation requirements manageable while providing a first level of replication for simulations. We chose to work at the mammalian scale to reflect the most widely studied context for CpG hypermutability. We used the mammalian tree topology and branch lengths obtained from Laurin-Lemay et al. (2018b) for all simulations.

In a first set of simulations, we used a model with a single global ω parameter. We conducted simulations with this ω parameter set to several different values, ranging from 0.1 to 0.9, increasing by 0.1 over this range. For each of these simulation settings on ω , we explored the effect of different values of λ_{CpG} , ranging from 1 (i.e., no hypermutability in the CpG context) to 10 (i.e., strong hypermutability, with rates 10 times higher in this context). Based on previous studies Laurin-Lemay et al. (2018a), these parameter values are considered realistic. For each of set of conditions, we repeated the simulation 100 times. Altogether, this amounts to a high number of replicates: three sets of nucleotide parameter values, 9 sets of ω values, 10 sets of λ_{CpG} values, and 100 replicates for each condition.

In a second series of simulations, we used three different ω values across the positions

of the gene (with equal weights, i.e., simulating 1/3 of the dataset with each) to produce the alignments. The idea here is to emulate heterogeneous intensities of purifying across sites, generally considered to be the case for most real genes. Importantly, none of our simulations were conducted with ω values greater than 1. We arbitrarily chose the five sets of three ω values, and for each, conducted simulations with a range of λ_{CpG} values. Our choices are listed below.

1. $\omega = 0.1, 0.2$ and 0.3 at $\lambda_{CpG} = 1-10$
2. $\omega = 0.2, 0.5$ and 0.7 at $\lambda_{CpG} = 1-10$
3. $\omega = 0.4, 0.5$ and 0.6 at $\lambda_{CpG} = 1-10$
4. $\omega = 0.5, 0.7$ and 0.9 at $\lambda_{CpG} = 1-10$
5. $\omega = 0.7, 0.8$ and 0.9 at $\lambda_{CpG} = 1-10$

Note that, the three ω values are always between 0 and 1, thus emulating simulation under the M7 model. Again, we repeated the simulations with three sets of nucleotide-level parameters, and each possible configuration was used to generate 100 alignments. All simulations were conducted to produce alignments of 750 codons length.

2.2 Jump-chain simulation of Markov processes

For historical reasons, most phylogenetic simulation software is based on producing artificial alignments by sampling from the integrated likelihood function (Felsenstein, 1981). However, a far simpler (and strictly equivalent) method, called *jump-chain* simulation, is based on a full realization of the Markov process along the branches of the tree. This method, first utilized in phylogenetics by Nielsen (2002), allows one to simulate under far richer models, since it does not require any of the costly numerical matrix exponentiation calculations. We give a detailed description of the method below.

2.2.1 Sampling the root state

As previously mentioned, the models we study are time-reversible, such that we may arbitrarily choose any node of the tree to be designated as the root node. The jump-chain simulation method starts by sampling a state at the root node from the stationary distribution of the model. In the case of the model of interest to us here, the stationary distribution is simply based on the nucleotide frequency parameters, which are multiplied across the three positions of a each codon. Thus, we simply need to draw from a 61-state (disallowing stop codons) discrete distribution. This can be achieved in a few different ways, but one is to simply draw a random number uniformly distributed on the unit interval, denoted U , and, since the discrete distribution sums to 1, finding within which interval the cumulative probability of states first surpasses U . We repeat the process until 750 codons are drawn, forming the entire length of the sequence, that we then set as the root state.

2.2.2 Sampling dwell times and substitutions

Having sampled the starting state, the process then diverges independently along the branches connected to the root node, but using the same starting state. Following along one of these branches, we first sample a waiting, or dwell, time to a substitution event. Under a continuous-time Markov process, a dwell time follows an exponential distribution. An exponential distribution has a single parameter, in this particular case, corresponding to the *rate away* from the current state, which we will denote R . The value of R is the sum of rates to all “nearest neighbor” sequences, that is, those that differ from the current state by a single point mutation. The rates to these nearest neighbors are specified in (2.1), given the root state as the initial state. Sampling a dwell time t from an exponential distribution in this case is done by setting $t = -\ln(1 - U)/R$ (see Nielsen (2002), for example). Note that if the t drawn goes beyond the length of the branch, the descendant node state is simply set to match the root node state (i.e., no substitutions are realized

along that branch). If the t sampled is less than the branch length, however, a new state is drawn from the discrete distribution of nearest neighbor state rates; the new state is sampled with a probability proportional to the rate towards it, again calculated from (2.1).

From here, we repeat the procedure: a new dwell time is drawn, with the exponential distribution's parameter being the rate away from the new state. If the time drawn (when added to the previous time drawn) is greater than the branch length, the state at the descendent node is set to this new state. If the time drawn still leaves us within the branch length, a new state is drawn, and so on, until having reached the end of the branch and setting the descendent node state. At this point, the process diverges again into two independent processes, with what was our descendent node becoming the new ancestral node. The process continues to bifurcate over the tree, until the times drawn along the terminal branches go beyond the tips of the tree, thereby setting the states of the simulated alignment.

2.2.3 Software

The simulation software utilized is available at <https://github.com/Simonll/LikelihoodFreePhylogenetics>. Our modifications and scripts used for running it are described in the appendix, as well as <http://github.com/kassandradickson/cpghypermutability>.

2.3 Tests of adaptive evolution

We used the CODEML software to analyze our simulated data sets. CODEML is a part of the PAML package (Yang, 2007). We applied three different choices with regards to ω : first, we invoked a single global ω parameter, just as written in 1.5, often referred to as the M0 setting (Yang et al., 2000); next, we used the site-models referred to as M7 and M8.

Under the M0 setting, the “test” for adaptive evolution is simply whether the maximum likelihood estimate of the global ω parameter is greater than 1. This is almost never the case, but applications of this setting can still help highlight how $\lambda_{CpG} > 1$ influences inferences on ω values. The more powerful tests of adaptive evolution are based on the site-models. As mentioned, we used two different site models for our tests of adaptive evolution. The M7 model considers the values of ω across sites to follow a beta distribution (which is constrained in the range 0 to 1). The M8 model builds upon the M7 model, allowing for an extra possible ω value, greater than one; the likelihood function is computed as a weighted average of the likelihood under the beta distribution and the extra ω value. Note that the M7 setting is nested within the M8 setting: by giving a null weight to components with the additional $\omega > 1$ parameter, the M8 model falls back to the M7 model. As such, by fitting both models via maximum likelihood, one can use a likelihood ratio test to determine if M8 is warranted over M7, which would then constitute evidence of adaptive evolution (i.e., that the gene has some sites with $\omega > 1$). In practice, tests that fail to reject M8 usually have a relatively high weight associated with the extra ω parameter, or have this extra parameter at a relatively high value. It is important to note that since our simulation conditions never included any values of $\omega > 1$, any evidence of adaptive evolution would be due to a model violation, in this case the effect of CpG hypermutability.

We conducted the likelihood ratio test at several levels of significance. Specifically, the tests were done at the 75%, 90%, 95%, 97% and 99% significance levels. We also sought to establish if any particular sites had strong evidence of adaptive evolution, using the post-treatment analysis (based on maximum likelihood parameter estimates) known as Naive Empirical Bayes (NEB) and Bayes Empirical Bayes (BEB) analysis (see Anisimova and Kosiol (2009) for details). Briefly, these approaches use the inferred parameters to calculate a probability for each site of being assigned to $\omega > 1$ with the M8 model.

Chapter 3

Results

3.1 Model fitting

3.1.1 Effects of CpG hypermutability on global ω parameter

We first applied the M0 model on our simplest simulations where $\lambda_{CpG} = 1$. Under these conditions, the model used for simulation and the model used for inference are the same. Inferences should therefore “land on their feet”, whereby the parameters inferred should match closely those of the generating process. Of course, since the continuous-time Markov chain we are simulating is a stochastic process, the match may not be exact. We show examples of our results with *BACE1* nucleotide settings, and various ω values, in figure (3.1). We generated histograms of the ω values inferred over 100 replicates, and marked the true ω value (used in the simulation) with a dashed red line. As the blue histograms show, the true value of ω is indeed well recovered when the simulation and inference model match.

When applying the M0 model on data produced with $\lambda_{CpG} > 1$, we find that the ω value inferred is inflated, with the effect more pronounced for higher λ_{CpG} values: the red histogram, where $\lambda_{CpG} = 5$, is shifted to the right of the true value of ω , and the green

Figure 3.1: Distribution of inferred ω values under the M0 model over 100 simulations. The blue density curve represents the frequency of ω when the $\lambda_{CpG} = 1.0$, the pink density curve shows the frequency of ω when the $\lambda_{CpG} = 5.0$, and finally the green density curve shows the frequency of ω when the $\lambda_{CpG} = 10.0$. The red dashed line indicates the true value of the ω

histogram, where $\lambda_{CpG} = 10$, even more so. For the latter conditions, we found several cases where all 100 simulations yielded inferences of ω that were higher than the true value, as exemplified by the leftmost panel.

3.1.2 Effects of CpG hypermutability on site-models

We next applied the site-models M7 and M8 to the simulations with heterogeneous values of ω across sites, and with a range of values of λ_{CpG} . As described in the methods chapter, these models can be used to perform a likelihood ratio test, and failure to reject M8 is considered as evidence of positive selection (Anisimova and Kosiol, 2009). We performed such a test on the sets of 100 alignments simulated under various settings, and counted how many of these data sets were indicative of positive selection. A graphical summary of results with one set of nucleotide-level parameter settings, and with different levels of significance, is shown in figure (3.2).

As expected with $\lambda_{CpG} = 1$, none of the simulated data sets were considered to exhibit evidence of positive selection, since none of them were generated with values of ω greater than 1. However, the frequency of alignments considered to be under positive selection increases as the value λ_{CpG} increases.

Figure 3.2: The number of alignments analyzed that had evidence of positive selection under the M7 and M8 model comparison (y-axis) within the 100 simulated datasets of the *VWA9* nucleotide settings with the corresponding λ_{CpG} value (x-axis) at significance levels: 75%, 90%, 95%, 97% and 99%.

Even at the strictest of significance levels (99%), when λ_{CpG} goes beyond 7.5, most of the simulations lead to results indicative of positive selection under the M7 versus M8 test. A more detailed reporting is presented in tables (3.1), (3.2), and (3.3). As these tables show, when $\lambda_{CpG} = 10$, most simulated alignments yield results indicative of positive selection.

ω Values	λ Value	Significance Level				
		75%	90%	95%	97.5%	99%
0.1, 0.2, 0.3	1	0	0	0	0	0
	5	8	2	2	1	0
	10	63	57	53	48	33
0.2, 0.5, 0.7	1	0	0	0	0	0
	5	33	12	9	6	4
	10	75	62	53	40	26
0.4, 0.5, 0.6	1	0	0	0	0	0
	5	40	20	11	9	3
	10	86	84	83	75	70
0.5, 0.7, 0.9	1	0	0	0	0	0
	5	85	72	65	59	41
	10	91	87	84	81	72
0.7, 0.8, 0.9	1	0	0	0	0	0
	5	82	61	50	39	29
	10	93	89	83	75	57

Table 3.1: The number of simulated alignments showing evidence of positive selection under the M7 and M8 model comparisons found within the 100 datasets of the *VWA9* nucleotide parameter settings. Simulations were done with three different ω parameters across sites and repeated for three different λ_{CPG} values, as shown in first and second columns.

3.2 Sites with strong evidence of positive selection

While the likelihood ratio test between M7 and M8 can show evidence of positive selection in a multiple sequence alignment, this should be interpreted with caution Anisimova and Kosiol (2009). Loosely speaking, such a test suggests that enough sites exhibit enough signal of having an ω greater than 1. The test does not in itself provide any direct information as to which sites might have an ω greater than 1. To explore this question in a maximum likelihood framework, the traditional approach has been to use the inferred parameter values to calculate, for each site of the alignment, the probability of having an ω greater than 1. Since this calculation is done after having already analyzed the data, it is considered a *posterior* probability. Several variations on these methods exist (reviewed in Yang (2013)), and we focus on those referred to as *naive* and *empirical* Bayes methods.

As shown in tables (3.5), (3.7), and (3.9), the two analyses produced slightly different

ω Values	λ Value	Significance Level				
		75%	90%	95%	97.5%	99%
0.1, 0.2, 0.3	1	0	0	0	0	0
	5	11	4	2	1	0
	10	77	71	63	57	53
0.2, 0.5, 0.7	1	0	0	0	0	0
	5	46	23	13	11	7
	10	94	92	90	86	77
0.4, 0.5, 0.6	1	0	0	0	0	0
	5	56	44	34	26	15
	10	95	95	95	95	95
0.5, 0.7, 0.9	1	6	4	4	4	4
	5	93	88	84	74	69
	10	91	91	91	91	91
0.7, 0.8, 0.9	1	7	5	5	5	5
	5	89	81	71	61	52
	10	96	96	96	96	94

Table 3.2: The number of simulated alignments showing evidence of positive selection under the M7 and M8 model comparisons found within the 100 datasets of the *BACE1* nucleotide parameter settings. Simulations were done with three different ω parameters across sites and repeated for three different λ_{CPG} values, as shown in first and second columns.

ω Values	λ Value	Significance Level				
		75%	90%	95%	97.5%	99%
0.1, 0.2, 0.3	1	0	0	0	0	0
	5	32	20	11	7	3
	10	87	87	86	84	83
0.2, 0.5, 0.7	1	0	0	0	0	0
	5	61	43	29	24	15
	10	79	74	67	58	39
0.4, 0.5, 0.6	1	2	1	1	1	1
	5	79	75	68	62	54
	10	87	87	87	87	87
0.5, 0.7, 0.9	1	22	9	4	4	3
	5	89	89	88	88	87
	10	94	94	94	93	93
0.7, 0.8, 0.9	1	2	1	1	1	1
	5	90	90	88	81	70
	10	89	87	83	74	67

Table 3.3: The number of simulated alignments showing evidence of positive selection under the M7 and M8 model comparisons found within the 100 datasets of the *FLT3* nucleotide parameter settings. Simulations were done with three different ω parameters across sites and repeated for three different λ_{CPG} values, as shown in first and second columns.

results. However, when the value of λ_{CPG} reaches 10, both approaches indicate a non-negligible proportion of sites having strong signals of positive selection. In the case of the empirical Bayes method, this proportion reaches 0.12 for the *VWA9* and *FLT3* nucleotide settings, and 0.16 for the *BACE1* nucleotide settings.

It would be interesting to repeat these analyses with a slight variation on the M8 model, known as the M8a model. With the latter model, an extra ω fixed at 1 is included in addition to the $\omega > 1$. This would account for cases where some sites evolve under a regime $\omega = 1$.

ω Values	λ Value	Empirical Bayes			Naive Bayes		
		Mean PSP	PSPs >95%	Proportion	Mean PSP	PSPs >95%	Proportion
0.1, 0.2, 0.3	1	0	0	0	0.01	0	0
	5	0.40	0	0	0.36	2	0.0556
	10	2.97	10	0.0337	3.51	69	0.1966
0.2, 0.5, 0.7	1	8.30	3	0.0036	0.02	0	0
	5	21.30	74	0.0347	4.39	6	0.0137
	10	27.53	256	0.0930	10.97	77	0.0702
0.4, 0.5, 0.6	1	3.59	0	0	0.09	0	0
	5	14.85	28	0.0189	4.80	15	0.0313
	10	23.98	180	0.0751	25.07	255	0.1017
0.5, 0.7, 0.9	1	60.88	176	0.0289	2.27	2	0.0088
	5	154.17	1365	0.0885	90.18	258	0.0286
	10	160.67	1744	0.1085	45.87	169	0.0368
0.7, 0.8, 0.9	1	178.42	649	0.0364	0.29	1	0.0345
	5	300.92	32.15	0.1068	69.39	87	0.0125
	10	312.45	3714	0.1189	34.06	150	0.0440

Table 3.5: The Empirical and the Naive Bayes averages over 100 simulations of the *VWA9* nucleotide settings. The number of positively selected sites (PSP) were found within the simulations. The mean of positively selected sites denotes how many sites within the alignment had evidence of positive selection. The positively selected sites greater than 95% significance level denotes the total amount of positively selected sites at that significance level throughout the entirety of the 100 simulations. The proportion column represents the proportion of positively selected sites in one simulations. Simulations were done with three different ω parameters across sites and repeated for three different λ_{CpG} values, as shown in first and second columns.

ω Values	λ Value	Empirical Bayes			Naive Bayes		
		Mean PSP	PSPs >95%	Proportion	Mean PSP	PSPs >95%	Proportion
0.1, 0.2, 0.3	1	0	0	0	0.01	0	0
	5	0.16	0	0	0.63	3	0.0476
	10	1.86	6	0.0323	4.50	116	0.2578
0.2, 0.5, 0.7	1	6.75	9	0.0133	0.05	0	0
	5	21.47	77	0.0359	7.72	12	0.0155
	10	35.37	378	0.1069	27.21	185	0.0680
0.4, 0.5, 0.6	1	3.33	0	0	0.12	0	0
	5	15.28	42	0.0275	7.05	32	0.0454
	10	31.35	309	0.0986	37.20	579	0.1556
0.5, 0.7, 0.9	1	47.85	132	0.0276	0.70	3	0.0429
	5	143.73	1348	0.0938	90.32	244	0.0270
	10	166.57	2419	0.1452	77.35	519	0.0671
0.7, 0.8, 0.9	1	152.31	540	0.0355	0.61	3	0.0492
	5	265.84	3132	0.1178	92.89	80	0.0083
	10	296.47	4743	0.1596	57.72	378	0.0655

Table 3.7: The Empirical and the Naive Bayes averages over 100 simulations of the *BACE1* nucleotide settings. The number of positively selected sites (PSP) were found within the simulations. The mean of positively selected sites denotes how many sites within the alignment had evidence of positive selection. The positively selected sites greater than 95% significance level denotes the total amount of positively selected sites at that significance level throughout the entirety of the 100 simulations. The proportion column represents the proportion of positively selected sites in one simulations. Simulations were done with three different ω parameters across sites and repeated for three different λ_{CpG} values, as shown in first and second columns

ω Values	λ Value	Empirical Bayes			Naive Bayes		
		Mean PSP	PSPs >95%	Proportion	Mean PSP	PSPs >95%	Proportion
0.1, 0.2, 0.3	1	0.02	0	0	0.06	0	0
	5	0.21	0	0	0.78	10	0.1282
	10	1.89	11	0.0582	11.25	272	0.2418
0.2, 0.5, 0.9	1	12.05	15	0.0124	0.03	0	0
	5	41.59	233	0.0560	8.38	15	0.0179
	10	46.11	420	0.0911	12.90	77	0.0637
0.4, 0.5, 0.6	1	5.79	2	0.0035	0.14	0	0
	5	27.14	94	0.0346	20.20	73	0.0361
	10	37.48	343	0.0915	35.42	377	0.1064
0.5, 0.7, 0.9	1	69.43	160	0.0230	4.46	1	0.0022
	5	286.90	3098	0.1080	196.65	431	0.0219
	10	302.76	3326	0.1099	133.60	204	0.0153
0.7, 0.8, 0.9	1	281.66	1236	0.0439	2.99	0	0
	5	541.83	6780	0.1251	123.73	123	0.0100
	10	579.86	6918	0.1193	73.94	108	0.0146

Table 3.9: The Empirical and the Naive Bayes averages over 100 simulations of the *FLT3* nucleotide settings. The number of positively selected sites (PSP) were found within the simulations. The mean of positively selected sites denotes how many sites within the alignment had evidence of positive selection. The positively selected sites greater than 95% significance level denotes the total amount of positively selected sites at that significance level throughout the entirety of the 100 simulations. The proportion column represents the proportion of positively selected sites in one simulations. Simulations were done with three different ω parameters across sites and repeated for three different λ_{CpG} values, as shown in first and second columns

Chapter 4

Discussion and future directions

4.1 Model violations

4.1.1 CpG hypermutability and site-models

The phenomenon of CpG hypermutability is known to be pervasive and pronounced in mammalian genomes (Hodgkinson and Eyre-Walker, 2011). Previous work by Laurin-Lemay et al. (2018b) indicates that in real mammalian data sets, λ_{CpG} can reach values of up to 20, whereas the average gene has a value around 8. Given the results of the simulations presented herein, this suggests that many studies utilizing site-models to detect positive selection (e.g. Kosiol et al. (2008)) may in part simply be detecting the effects of context-dependent CpG hypermutability. As such, great caution should be taken before concluding that positive selection has played an important role in shaping a given mammalian gene. More simulations could be done exploring a greater range of parameter values and simulation settings. For instance, we have not explored the effect of sequence length on the simulation results. Another potential variation to the simulations would have λ_{CpG} changing across the branches of the phylogeny. Until more work is done to estimate the extent of context-dependent mutational features in other taxonomic groups, it will remain

difficult to know the general reliability of the widely used tests of positive selection such as those available in CODEML.

4.1.2 Other model violations

Moreover, recent work has highlighted numerous other model violations that could have misleading effects on tests of positive selection. Still focusing on the mammalian context, GC-biased gene conversion is also considered pervasive; this phenomenon is the result of favoring the transmission of G:C alleles over A:T alleles during meiotic recombination (Duret and Galtier, 2009). Other work has emphasized on classical site models fail to make any distinction between the nature of amino acids being replaced for nonsynonymous events (Rodrigue et al., 2010), or the extreme variation in effective population size across the mammalian phylogeny (Lanfear et al., 2014). Many invertebrates, and certainly bacterial taxonomic groups, are also known to exhibit strong codon usage bias (Ermolaeva et al., 2001). These and other model violations are highly likely to mislead the classical approaches to detecting positive selection.

4.2 Jump-chain simulation for prospecting more realistic models

Recent work by Rodrigue and Lartillot (2017) has proposed a re-framing of approaches to detecting adaptive evolution from protein coding gene, based on contrasting of the nonsynonymous rates expected under the mutation-selection balance relative to the measured nonsynonymous rates. While these approach invoke sophisticated statistical methods to infer site-specific amino acid profiles governing purifying selection, they also, like the traditional models, ignore the potential effects of context-dependent mutational features. Indeed, the general trend into proposing novel models has been to focus on one particular

aspect of interest, but always building within the simplest traditional models of codon substitution. The rationale behind such an incremental approach is usually technical: almost all novel models require the development of novel computational and/or statistical methodologies.

While combining all the known features of molecular evolution into an all-encompassing inference model is likely to take many years of technical development, the present work, along with other previous work (Laurin-Lemay et al., 2018a,b), suggests that the jump-chain method of simulation could have an important role in prospecting the most pressing directions to develop. Future work along these lines could include simulating under a structurally informed mutation-selection-based model with CpG hypermutability and variable effective population size along the phylogeny. Moreover, by suppressing each complexity from the simulation model in turn (say, population size, then CpG hypermutability, etc.), and devising discrepancy statistics between simulated and real data, the approach could shed light on relative importance of each feature on shaping protein-coding genes.

APPENDICES

Appendix A

Code to Produce the Control File and Analysis

A.1 Control Files

Properties from the control files that are produced indicate the sequence file that was produced during the simulation process with the proper ω and λ_{CPG} value, as well as the gene name. In the control file, it also indicates the tree file and the the output file name, which will be used to analysis using the CODEML package. In the code, it produces all of the 100 sequence control file and analyses them on a cluster.

A.2 Producing the Control File Needed for CODEML

All values, as well as genes can be manipulated from the command line. Here is an example of the code that produces and analyses the DNA sequences:

```
#!/bin/bash
```

```

#Indicates how many values are needed for this script to work
argc=$#
requiredArgc=3

#This statement will show up in the command line, if there are not enough values and it reminds the person what values should
if [ $argc -ne $requiredArgc ]; then
echo "./MakeControlFile_research.sh <omegaValue> <lambdaValue> <geneName>"
exit 1
fi

#Shows the script which values correspond to which number it was inputted as
omegaValue=$1
lambdaValue=$2
geneName=$3

#This prepares the CODEML control file
function Control {

echo "preparing codeml control file..."

for i in {0..99} ; do
echo "seqfile = $geneName.Omega$omegaValue.L$lambdaValue.$i.fas" > /share/data/kdickson/simu/$geneName.O$omegaVa
lue.L$lambdaValue.$i.ctl
echo "treefile = Mammalia-39sp-CAT_unrooted_withoutsupport.tre" >> /share/data/kdickson/simu/$geneName.O$omegaVa
lue.L$lambdaValue.$i.ctl
echo " " >> /share/data/kdickson/simu/$geneName.O$omegaValue.L$lambdaValue.$i.ctl
echo "outfile = $geneName.O$omegaValue.L$lambdaValue.$i" >> /share/data/kdickson/simu/$geneName.O$omegaValue.L$1
ambdaValue.$i.ctl
cat /share/data/kdickson/simu/codemlTail_BNF.ctl >> /share/data/kdickson/simu/$geneName.O$omegaValue.L$lambdaVal
ue.$i.ctl
done

}

#This runs the CODEML program onto the cluster
function Cluster {
for i in {0..99} ; do
echo "running codeml"
echo "#!/bin/bash" > $geneName.O$omegaValue.L$lambdaValue.$i.sh
echo " " >> $geneName.O$omegaValue.L$lambdaValue.$i.sh
echo "/share/data/kdickson/simu/paml4.9d/bin/codeml /share/data/kdickson/simu/$geneName.O$omegaValue.L$lambdaVal

```

```

ue.$i.ct1" >> $geneName.O$omegaValue.L$lambdaValue.$i.sh
echo "submitting codeml..."
qsub -q all.q@compute-0-5.local -S /bin/bash -cwd $geneName.O$omegaValue.L$lambdaValue.$i.sh -j y -o /share/data
/kdickson/simu/paml4.9d/examples/simu/$geneName.O$omegaValue.L$lambdaValue.$i
echo "...done submitting codeml."

done
}
#Main
Control
Cluster

```

A.3 CODEML Analysis

In order to run the CODEML control file previous made, a code was implemented to analysis the sequences on the cluster found at Carleton University. Within the control file, the parameters and what was needed in the output is shown. The code ran just allowed the analysis for all sequences take place on a cluster. An example of the code ran is shown below:

```

#!/bin/bash

#This is the input needed to run CODEML

#for ((i=0; i<100; i++)); do
#for i in {1..10} ; do
echo "running codeml..."
echo "#!/bin/bash" > FLT3.00p2.L20.sh
echo " " >> FLT3.00p2.L20.sh
echo "/share/data/kdickson/simu/paml4.9d/bin/codeml /share/data/kdickson/simu/FLT3.00p2.L20.ct1" >> FLT3.00p2.L20.sh
echo "submitting codeml..."
qsub -q all.q@compute-0-15.local,all.q@compute-0-5.local,all.q@compute-0-8.local -S /bin/bash -cwd FLT3.00p2.L20.sh -j y -o /s
#done
echo "...done submitting codeml."

```

A.4 Analysis Code for Likelihood Ratio Test

The values in order to perform this test were found within the output of the sequences. Since, this was the case, the values were just searched in the output and compared to each other. This code searches for the needed values and compares the differences and outputs the differences in a different output file.

```
#!/bin/bash

argc=$#
requiredArgc=3

if [ $argc -ne $requiredArgc ]; then
echo "./program.sh <omegaValue> <lambdaValue> <geneName>"
exit 1
fi

omegaValue=$1
lambdaValue=$2
geneName=$3

#This takes out the M7 and M8 number needed for the likelihood ratio test
function grepLRTM7 {
for i in {0..99} ; do
grep "lnL" /share/data/kdickson/simu/gene.$geneName/$geneName.Omega$omegaValue.L$lambdaValue/$geneName.O$omegaValue.L$lambdaValue
done

sed -i -e "s/.*: //g" /share/data/kdickson/simu/$geneName.O$omegaValue.L$lambdaValue

}

#This shows the difference between the M7 and M8 output values
function Difference {
awk '{s=$0;getline;r=s-$0;if (r<0)r;print r;next}' $geneName.O$omegaValue.L$lambdaValue >> /share/data/kdickson/
simu/$geneName.O$omegaValue.L$lambdaValue.delta

}

#Main
grepLRTM7
Difference
```


A.5 Analysis for Bayes Test

Similar to the likelihood ratio test, the Bayes test were also in the output given by the CODEML analysis. The code searches the output for these values and outputs them into a different output file.

```
#!/bin/bash

#####

###This Script greps the Naive Bayes and Emperical Bayes and Compares
#####

argc=$#
requiredArgc=3

if [ $argc -ne $requiredArgc ]; then
echo "./grepCompareBayes.sh <omegaValue> <lambdaValue> <geneName>"
exit 1
fi

omegaValue=$1
lambdaValue=$2
geneName=$3

#This takes out the numbers for the Native Empirical Bayes Test
function SedNaive {
for i in {0..99} ; do
sed -e '1,/~Naive Empirical Bayes (NEB) analysis/d' -e '/~Bayes/, $d' /share/data/kdickson/simu/gene.$geneName/$geneName.Omega
done

sed -i -e "s/. *w //g" /share/data/kdickson/simu/Naive.$geneName.Omega$omegaValue.L$lambdaValue

}

#This takes out the numbers for the Empirical Bayes Test
function SedEmpirical {
for i in {0..99} ; do
sed -e '1,/~Bayes Empirical Bayes/d' -e '/~The grid/, $d' /share/data/kdickson/simu/gene.$geneName/$geneName.Omega
a$omegaValue.L$lambdaValue/$geneName.O$omegaValue.L$lambdaValue.$i >> /share/data/kdickson/simu/Empirical.$geneName.Omega
a$omegaValue.L$lambdaValue
```

done

```
sed -i -e "s/.w //g" /share/data/kdickson/simu/Empirical.$geneName.Omega$omegaValue.L$lambdaValue  
}
```

#Main

SedNaive

SedEmpirical

References

- Anisimova, M. and Kosiol, C. (2009). Investigating protein-coding sequence evolution with probabilistic codon substitution models. *Molecular biology and evolution*, 26(2):255–271.
- Bird, A. (1999). Dna methylation de novo. *Science*, 286(5448):2287–2288.
- Bird, A. P. (1980). DNA methylation and the frequency of CpG in animal DNA. *Nucleic Acids Research*, 8(7):1499–1504.
- Duret, L. and Galtier, N. (2009). Biased gene conversion and the evolution of mammalian genomic landscapes. *Annual review of genomics and human genetics*, 10:285–311.
- Ermolaeva, M. D. et al. (2001). Synonymous codon usage in bacteria. *Current issues in molecular biology*, 3(4):91–97.
- Felsenstein, J. (1981). Evolutionary trees from dna sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17:1432–1432.
- Hasegawa, M., Kishino, H., and Yano, T.-a. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of Molecular Evolution*, 22:160–174.
- Hellmann, I., Ebersberger, I., Ptak, S. E., Pääbo, S., and Przeworski, M. (2003). A Neutral Explanation for the Correlation of Diversity with Recombination Rates in Humans. *The American Journal of Human Genetics*, 72(6):1527 – 1535.
- Hodgkinson, A. and Eyre-Walker, A. (2011). Variation in the mutation rate across mammalian genomes. *Nature reviews genetics*, 12(11):756–766.
- Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282.
- Jukes, T. and Cantor, C. (1969). *Mammalian protein metabolism*, volume 3. Academic Press.
- Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120.
- Kosiol, C., Vinař, T., da Fonseca, R. R., Hubisz, M. J., Bustamante, C. D., Nielsen, R., and Siepel, A. (2008). Patterns of positive selection in six mammalian genomes. *PLoS Genet*, 4(8):e1000144.
- Krašovec, R., Richards, H., Gifford, D. R., Hatcher, C., Faulkner, K. J., Belavkin, R. V., Channon, A., Aston, E., McBain, A. J., and Knight, C. G. (2017). Spontaneous mutation rate is a plastic trait associated with population density across domains of life. *PLOS Biology*, 15(8):1–19.

- Krawczak, M., Ball, E. V., and Cooper, D. N. (1998). Neighboring-Nucleotide Effects on the Rates of Germ-Line Single-Base-Pair Substitution in Human Genes. *The American Journal of Human Genetics*, 63(2):474 – 488.
- Lanfear, R., Kokko, H., and Eyre-Walker, A. (2014). Population size and the rate of evolution. *Trends in ecology & evolution*, 29(1):33–41.
- Laurin-Lemay, S., Philippe, H., and Rodrigue, N. (2018a). Multiple factors confounding phylogenetic detection of selection on codon usage. *Molecular biology and evolution*, 35(6):1463–1472.
- Laurin-Lemay, S., Rodrigue, N., Lartillot, N., and Philippe, H. (2018b). Conditional approximate bayesian computation: A new approach for across-site dependency in high-dimensional mutation–selection models. *Molecular biology and evolution*, 35(11):2819–2834.
- Maharjan, R. P. and Ferenci, T. (2017). A shifting mutational landscape in 6 nutritional states: Stress-induced mutagenesis as a series of distinct stress input–mutation output relationships. *PLOS Biology*, 15(6):1–22.
- Mayrose, I., Doron-Faigenboim, A., Bacharach, E., and Pupko, T. (2007). Towards realistic codon models: among site variability and dependency of synonymous and non-synonymous rates. *Bioinformatics*, 23(13):i319–i327.
- Nei, M. and Kumar, S. (2000). *Molecular evolution and phylogenetics*. Oxford University Press.
- Nielsen, R. (2002). Mapping mutations on phylogenies. *Systematic biology*, 51(5):729–739.
- Robinson, D. M., Jones, D. T., Kishino, H., Goldman, N., and Thorne, J. L. (2003). Protein Evolution with Dependence Among Codons Due to Tertiary Structure. *Molecular Biology and Evolution*, 20(10):1692–1704.
- Rodrigue, N. and Lartillot, N. (2017). Detecting adaptation in protein-coding genes using a bayesian site-heterogeneous mutation-selection codon substitution model. *Molecular biology and evolution*, 34(1):204–214.
- Rodrigue, N., Philippe, H., and Lartillot, N. (2006). Assessing site-interdependent phylogenetic models of sequence evolution. *Molecular biology and evolution*, 23:1762–1775.
- Rodrigue, N., Philippe, H., and Lartillot, N. (2010). Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proceedings of the National Academy of Sciences*, 107(10):4629–4634.
- Rodrigue, N., Philippe, H., Lartillot, N., and Bryant, D. (2005). Site interdependence attributed to tertiary structure in amino acid sequence evolution. *Gene*, 247:207–217.
- Rodrigue, N., Philippe, H., Lartillot, N., and Kleinman, C. (2009). Computational methods for evaluating phylogenetic models of coding sequence evolution with dependence between codons. *Molecular Biology and Evolution*, 26:1663–1676.
- Subramanian, S. and Kumar, S. (2006). Higher Intensity of Purifying Selection on >90% of the Human Genes Revealed by the Intrinsic Replacement Mutation Rates. *Molecular Biology and Evolution*, 23(12):2283–2287.
- Suzuki, Y., Gojobori, T., and Kumar, S. (2009). Methods for Incorporating the Hypermutability of CpG Dinucleotides in Detecting Natural Selection Operating at the Amino Acid Sequence Level. *Molecular Biology and Evolution*, 26(10):2275–2284.
- Tavaré, S. (1986). Some probabilistic and statistical problems in the analysis of dna sequences. *Lectures on mathematics in the life sciences*, 17(2):57–86.

- Yang, Z. (1993). Paml: a program package for phylogenetic analysis by maximum likelihood. *Molecular Biology and Evolution*, 24:1586–1591.
- Yang, Z. (1994). Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods. *Journal of Molecular evolution*, 39(3):306–314.
- Yang, Z. (2007). Paml 4: phylogenetic analysis by maximum likelihood. *Molecular biology and evolution*, 24(8):1586–1591.
- Yang, Z. (2013). *Computational molecular evolution*. Oxford University Press.
- Yang, Z., Nielsen, R., Goldman, N., and Pederson, A.-M. K. (2000). Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics*, 155(1):431–449.
- Yang, Z. and Nielsen, R. (2008). Mutation-Selection Models of Codon Substitution and Their Use to Estimate Selective Strengths on Codon Usage. *Molecular Biology and Evolution*, 25(3):568–579.