The Dissertation Committee for Stephanie Jill Spielman
certifies that this is the approved version of the following dissertation:

Investigating the behaviors and limitations of phylogenetic models of protein-coding sequence evolution

Committee:

Claus O. Wilke, Supervisor

Jeffrey E. Barrick

James J. Bull

David M. Hillis

Hans A. Hoffmann

Investigating the behaviors and limitations of phylogenetic models of protein-coding sequence evolution

by

Stephanie Jill Spielman, B.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

To my family and friends.

Acknowledgments

I thank my advisor Claus Wilke for the most unexpected and rewarding graduate experience. Dr. Wilke's consistent support, encouragement, and respect for me as an independent scholar have made the past five years remarkably less painful than everyone warned me a Ph.D. would be. In fact, my Ph.D. has not only been bearable, but also (dare I say) enjoyable. I am confident when I say that I could not have done this without him, and indeed I would not have wanted to.

I also thank the Wilke Lab members (in particular Dakota Derryberry, Eleisha Jackson, Austin Meyer, and Ben Jack) for their helpful feedback and support over the years. Finally, I thank my family whose unwavering love, encouragement, and support for my career aspirations has kept me resolute in pursuing my goals.

Investigating the behaviors and limitations of phylogenetic models of protein-coding sequence evolution

Stephanie Jill Spielman, Ph.D.
The University of Texas at Austin, 2016

Supervisor: Claus O. Wilke

Probabilistic models which infer the strength and direction of natural selection from protein-coding sequences are among the most widely-used tools in comparative sequence analysis. A variety of phylogenetic models of coding-sequence evolution have been developed. However, these models have been produced independently from one another. As a consequence, it has been entirely unknown whether inferences from different models reveal similar or incompatible information about the evolutionary process. In this dissertation, I derive and study the mathematical relationship between two probabilistic models of protein-coding sequence evolution: dN/dS-based models, which estimate evolutionary rates, and mutation–selection models, which estimate site-specific amino-acid fitnesses. I demonstrate how this relationship reveals the behavioral properties, limitations, and applicabilities of different inference frameworks, which leads to concrete recommendations for how these models should best be employed in evolutionary sequence analysis. In Chapter 2, I develop a flexible and extendable software, implemented as a module in the Python programming langauge, for simulating sequences along phylogenies according to standard evolutionary models. This software platform provides an independent and user-friendly platform for testing model behavior, or indeed developing novel evolutionary models, thus enabling robust comparisons of modeling frameworks. In Chapter 3, I derive a mathematical relationship between dN/dS and amino-acid fitness values, and I show that mutation–selection models fully encompass information encoded in dN/dS models, provided that sequences are evolving under purifying selection. I further use this relationship to show that certain commonly-used dN/dS-based models are strongly and systematically biased. I additionally show that standard metrics used for model selection in phylogenetics (e.g. Akaike Information Criterion) may be positively misleading and indicate strong support for incorrect models. Finally, in Chapter 4, I apply the mathematical relationship developed in Chapter 3 to study the accuracy of two competing mutation–selection inference implementations, whose relative merits have been heavily debated in the literature. My approach demonstrates that mutation–selection inference platforms that treat amino-acid fitnesses as fixed-effect variables precisely estimate site-specific evolutionary constraints. By contrast, inference platforms that treat fitnesses as random-effect variables systematically underestimate the strength of natural selection across sites. Taken together, the work presented

in this dissertation yields novel insights into how these popular evolutionary models can best be applied to sequence data, how their results should be interpreted, and finally how future model development should be conducted in order to yield robust and reliable inference methods.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Comparative sequence analysis has taken a central role in modern biological research. With a wealth of sequence data being generated on a daily basis, it is critically important that the scientific community have powerful, verified tools to extract biologically relevant information from this data. For this purpose, probabilistic models that characterize the evolutionary dynamics of protein-coding sequences along a phylogeny have been particularly valuable. Such methods have seen wide-ranging applications, in basic biological research and medical and epidemiological fields. For instance, studies of retroviral sequences using evolutionary models have revealed mechanisms of increased virulence in West Nile Virus [19] and evolved drug resistance in HIV [18, 76]. Moreover, evolutionary sequence analysis has greatly facilitated vaccine development and therapeutic target identification by revealing genetic signatures of viral immune escape and antigenic shifts, particular in the influenza virus [31, 61, 69, 120].

1.1 Background: Markov Models

Today, the molecular evolution of protein-coding sequences is most commonly studied using continuous-time Markov models [6, 126]. Analysis is performed on a multiple sequence alignment and a corresponding phylogeny, and the substitution process at each alignment position is modeled as an independent Markov chain. Markov chains are memoryless processes, meaning that probability of transitioning to a new state depends only on the current state, not the past states. In the context of coding-sequence evolution, the 61 sense codons represent the states in each Markov chain.

The substitution rates between states are given by an *instantaneous rate matrix* (also known as the transition matrix) Q, describing the probability with which each state transitions to each other state in an infinitesimally small amount of time. In codon models, it is commonly assumed that instantaneous changes only occur between codons with a single nucleotide difference [6]. In other words, the instantaneous rate of change from codon ATT to AGC, for example, would be 0. For computational feasibility, models of sequence evolution are often assumed to be time-reversible, formally indicated as $\pi_i Q_{ij} = \pi_j Q_{ji}$, where π_i represents the *equilibrium frequency* (also known as stationary or steady-state frequency) of state (codon) i, and Q_{ij} is the entry in the transition matrix Q giving the probability of changing from codon i to j.

To incorporate time into this process, a *transition-probability* matrix is calculated $P = \exp(Qt)$, indicating the substitution probability between states over a certain time, represented by the parameter t. In the context of phylogenetic data, t is conveniently interpreted as the branch length along a given edge of a phylogenetic tree. To ensure that t is meaningful with respect to physical time, the rate matrix Q is usually scaled such that the mean substitution rate is 1: $-\sum_{i=1} \pi_i Q_{ii} = 1$ [42, 125], where Q_{ii} represents the diagonal elements of the rate matrix. Through this procedure, the t parameter for each branch explicitly represents the expected number of substitutions per unit (in this case, codon).

An important property of the matrix P is that it takes into consideration the possibility of "hidden" changes along a given branch, thus accounting for multiple and/or convergent substitutions. It is also important to note that, when $t \to \infty$, the distribution of states in the Markov chain will precisely equal the stationary frequencies π, indicating the limiting behavior of the process.

For any model of sequence evolution, the matrix Q will be populated by certain parameters, for example a rate of change r, which must be calculated. Obtaining analytical solutions for these parameters is not usually possible, and hence numerical approaches are used to optimize parameter values for a given model. As such, parameters for these models are commonly estimated using *maximum-likelihood* (ML). The *likelihood* is the probability of observing the data (e.g. a multiple sequence alignment) under an assumed model of sequence evolution. Specifically, the likelihood is given by $L = P(D|\mathcal{T};\theta)$, where D represents the data (a multiple sequence alignment), θ are the model parameters, and $\mathcal{T}$ represents the phylogeny (including both topology and branch lengths) which, for computational tractability, is often assumed to be fixed. As stated, most Markov models of sequence evolution treat each alignment site as independent, and hence the likelihood for a full dataset is the product of individual-site likelihoods: $\prod_{k=1}^{M} L = P(D^{(k)}|\mathcal{T};\theta)$, where k is an alignment site, $D^{(k)}$ is the data at alignment site k, and the product sums over all sites M (note that "site", in the case of codon models, refers to a codon site comprised of a nucleotide triplet). The likelihood is calculated along the phylogeny using Felsenstein's pruning algorithm [40], and it is optimized using numerical methods. This procedure provides estimates of the model parameters which best describe the data D.

As an alternative to ML, some Bayesian approaches have additionally been popularized. Rather than optimizing the likelihood function, the Bayesian approach seeks to optimize the posterior probability $P(\theta|X)$, or in other words the model given the observed data. This probability can be calculated with Bayes Theorem:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}. \tag{1.1}$$

$P(D|\theta)$ is the likelihood function, $P(\theta)$ represents the *prior probability* of this model, and $P(D)$ is the *marginal likelihood* of the data. The marginal likelihood is notoriously difficult to calculate, which has precluded the use of Bayesian approaches for many years. Modern applications estimate this probability using Markov Chain Monte Carlo (MCMC) approaches that integrate over the parameter space. Again, this process is assumed to be independent at each site. Although Bayesian approaches have been well-developed for phylogenetic reconstruction [35, 64], they are not yet widely used to model selection pressures in coding sequences (with the notable exception of ref. [98]). Future algorithmic developments may facilitate more widespread use in the coming years.

1.2 Markov Models of coding-sequence evolution

Traditionally, codon substitution models have been used to identify signatures of natural selection in protein-coding sequences [6]. First introduced in the 1990s, codon substitution models estimate dN/dS, the ratio of nonsynonymous to synonymous substitution rates. This metric indicates how quickly a protein's constituent amino acids change [42, 79, 81], allowing for the identification of positively-selected regions or sites in protein sequences [81, 131]. Various transition matrices have been proposed for codon substitution models, with the simplest one being either the Goldman-Yang (GY) matrix [42] or the Muse-Gaut (MG) matrix [79]. The GY transition matrix, as an example, is given by,

$$Q_{ij} = \begin{cases} \pi_j & \text{synonymous transversion} \\ \kappa\pi_j & \text{synonymous transition} \\ \omega\pi_j & \text{nonsynonymous transversion} \\ \omega\kappa\pi_j & \text{nonsynonymous transition} \\ 0 & \text{multiple nucleotide changes} \end{cases}, \qquad (1.2)$$

where π_j is the equilibrium frequency of codon j, κ is the ratio of transition-to-transversion mutation rates, and ω is the rate of nonsynonymous change. The focal parameter of this model is ω, which represents dN/dS (dS is implicitly assumed to equal 1 in this model).

In recent years, a complementary model of coding-sequence evolution, known as the *mutation–selection* (MutSel) model, has gained popularity [33, 46, 96, 100, 115, 116, 130]. The MutSel model describes how the presence of different amino acids across protein sites affects overall protein fitness. In particular, MutSel models consider the substitution process using fundamental population genetics principles [46, 53]. At each site, MutSel models infer the mutation rate (i.e. the rate of nucleotide change) as well as the fixation probability of that particular mutation. In this way, MutSel models estimate site-specific amino acid fitnesses, which is directly related to the probability of observing each amino acid at a given protein position [106]. This information indicates selective response to different mutations across sites. The transition matrix for the MutSel model is given by,

$$q_{ij} = \begin{cases} \mu_{ij} \frac{S_{ij}}{1-\exp(S_{ji})} & \text{single nucleotide change} \\ \\ 0 & \text{multiple nucleotide changes} \end{cases}, \tag{1.3}$$

for a substitution from codon i to j, where μ_{ij} is the nucleotide-level mutation rate, and S_{ij} is the *scaled selection coefficient*, which represents the fitness difference between codons j and i.

While first introduced in a seminal 1998 paper by Halpern and Bruno [46], the MutSel model has rarely been used due to its extremely high computational expense. Recent advancements in high-performance computing, however, have allowed for the development of two inference platforms for analyzing phylogenetic sequence data with this model [98, 116]. Thus, for the first time, the molecular evolution community now has the opportunity to apply the mutation–selection platform.

Although both the codon substitution and mutation–selection frameworks provide meaningful information about the evolutionary process at sites in protein-coding sequences, how these modeling frameworks relate to one another has been entirely unstudied. Indeed, codon substitution models and MutSel models have distinct focal parameters: Codon substitution models focus primarily on the evolutionary rate ratio dN/dS, and MutSel models focus on amino-acid fitness values. How dN/dS inferences and amino-acid fitness inferences relate to each other, however, is not immediately clear. Whether the inferences from these models will contain overlapping, opposing, or entirely distinct information about the strength and direction of natural selection is, therefore, an open question.

In this dissertation, I examine the relationship between these two modeling frameworks. In particular, this work is motivated by the concept that we can gain a more precise and nuanced understanding of how molecular evolution inference frameworks behave by considering how distinct models interpret the same data. In Chapter 2, I develop Pyvolve, a flexible and user-friendly Python module which simulates sequences along phylogenies according to continuous-time Markov models of sequence evolution [108]. This simulation platform provides one of the first open-source platforms for simulating sequences according to the mutation–selection modeling framework, thus enabling the work performed in Chapters 3 and 4. In Chapter 3, I present a mathematical relationship between dN/dS and codon fitness values, which allows me to directly compare dN/dS-based and MutSel model inferences. I use this framework to compare the relative performance of distinct dN/dS model parameterizations, and I uncover biases inherent to a commonly-used class of dN/dS models. Finally, in Chapter 4, I use the mathematical framework developed in 3 to compare performance between the two aforementioned mutation–selection inference platforms [96,98,115,116]. I find that fixed-effects modeling frameworks strongly outperform random-effects frameworks, which systematically underestimate the strength of selective constraint at individual sites. Together, this research represents a significant step towards integrating distinct modeling frameworks to develop a more unified approach to uncovering the signatures of natural selection from sequence data.

Chapter 2

Pyvolve: A Flexible Python Module for Simulating Sequences along Phylogenies

2.1 Introduction

This work has been previously published in the journal *PLOS ONE*.[1]

The Python programming language has become a staple in biological computing. In particular, the molecular evolution community has widely embraced Python as standard tool, in part due to the development of powerful bioinformatics modules such as Biopython [25] and DendroPy [111]. However, Python lacks a robust platform for evolutionary sequence simulation.

In computational molecular evolution and phylogenetics, sequence simulation represents a fundamental aspect of model development and testing. Through simulating genetic data according to a particular evolutionary model, one can rigorously test hypotheses about the model, examine the utility of analytical methods or tools in a controlled setting, and assess the interactions of different biological processes [7].

[1] S. J. Spielman and C. O. Wilke. Pyvolve: A Flexible Python Module for Simulating Sequences along Phylogenies. PLOS ONE, 10(9):e0139047, 2015. C. O. Wilke helped to write the manuscript.

To this end, we introduce Pyvolve, a sequence simulation Python module (with dependencies Biopython [25], SciPy, and NumPy [82]). Pyvolve simulates sequences along a phylogeny using continuous-time Markov models of sequence evolution for nucleotides, amino acids, and codons, according to standard approaches [126]. The primary purpose of Pyvolve is to provide a user-friendly and flexible sequence simulation platform that can easily be integrated into Python bioinformatics pipelines without necessitating the use of third-party software. Furthermore, Pyvolve allows users to specify and evolve custom evolutionary models and/or states, making Pyvolve an ideal engine for novel model development and testing.

2.2 Substitution models and frameworks in Pyvolve

Pyvolve is specifically intended to simulate gene sequences along phylogenies according to Markov models of sequence evolution. Therefore, Pyvolve requires users to provide a fixed phylogeny along which sequences will evolve. Modeling frameworks which are included in Pyvolve are detailed in Table 2.1.

Table 2.1: Substitution models included in Pyvolve.

Modeling Framework	**Available Models**
Nucleotide	GTR [117] and all nested variants (e.g. HKY85 [47] and TN93 [114])
Amino acid	JTT [51], WAG [121], LG [66], mtMAM [124], mtREV24 [3], DAYHOFF [29], AB [74]
Mechanistic codon	GY-style [42, 81] and MG-style [79]
Empirical codon	ECM (restricted and unrestricted) [62]
Mutation–selection	Halpern-Bruno model [46], for both nucleotides and codons

Pyvolve supports both site-wise and branch (temporal) heterogeneity.

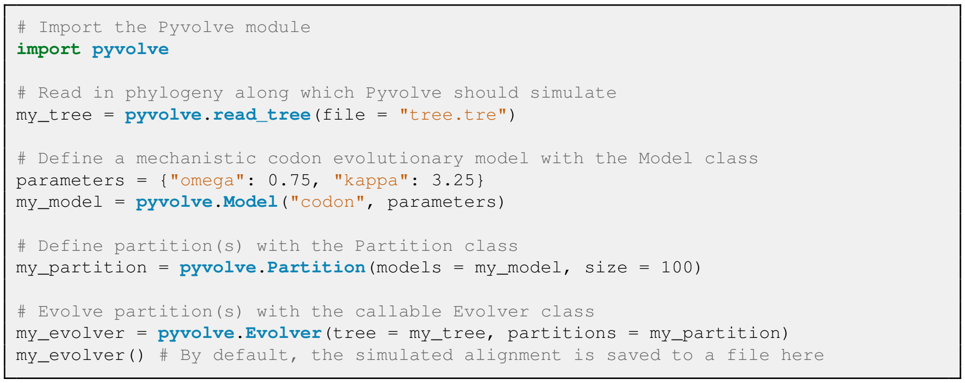
Site-wise heterogeneity can be modeled with Γ or Γ+I rates, or users can specify a custom rate-distribution. Further, users can specify a custom rate matrix for a given simulation, and thus they can evolve sequences according to substitution models other than those shown in Table 2.1. Similarly, users have the option to specify a custom set of states to evolve, rather than being limited to nucleotide, amino-acid, or codon data. Therefore, it is possible to specify arbitrary models with corresponding custom states, e.g. states 0, 1, and 2. This general framework will enable users to evolve, for instance, states according to models of character evolution, such as the Mk model [67].

Similar to other simulation platforms (e.g. Seq-Gen [89], indel-Seq-Gen [110], and INDELible [41]), Pyvolve simulates sequences in groups of *partitions*, such that different partitions can have unique evolutionary models and/or parameters. Although Pyvolve enforces that all partitions within a single simulation evolve according to the same model family (e.g. nucleotide, amino-acid, or codon), Python's flexible scripting environment allows for straight-forward alignment concatenation. Therefore, it is readily possible to embed a series of Pyvolve simulations within a Python script to produce highly-heterogeneous alignments, for instance where coding sequences are interspersed with non-coding DNA sequences. Moreover, Pyvolve allows users to specify, for a given partition, the ancestral sequence (MRCA) to evolve.

In addition, we highlight that Pyvolve is among the first open-source simulation tools to include the mutation–selection modeling framework introduced by Halpern and Bruno in ref. [46] (we note that the simulation software SGWE [10] also includes this model). Importantly, although these models were originally developed for codon evolution [46, 130], Pyvolve implements mutation–selection models for both codons and nucleotides. We expect that this implementation will foster the continued development and use of this modeling framework, which has seen a surge of popularity in recent years [33, 49, 98, 100, 109, 115, 116].

2.3 Basic Usage of Pyvolve

The basic framework for a simple simulation with the Pyvolve module is shown in Figure 2.1. To simulate sequences, users should input the phylogeny along which sequences will evolve, define evolutionary model(s), and assign model(s) to partition(s). Pyvolve implements all evolutionary models in their most general forms, such that any parameter in the model may be customized. This behavior stands in contrast to several other simulation platforms of comparable scope to Pyvolve. For example, some of the most commonly used simulation tools that implement codon models, including INDELible [41], EVOLVER [127], and PhyloSim [107], do not allow users to specify dS rate variation in codon models. Pyvolve provides this option, among many others.



```
# Import the Pyvolve module
import pyvolve

# Read in phylogeny along which Pyvolve should simulate
my_tree = pyvolve.read_tree(file = "tree.tre")

# Define a mechanistic codon evolutionary model with the Model class
parameters = {"omega": 0.75, "kappa": 3.25}
my_model = pyvolve.Model("codon", parameters)

# Define partition(s) with the Partition class
my_partition = pyvolve.Partition(models = my_model, size = 100)

# Evolve partition(s) with the callable Evolver class
my_evolver = pyvolve.Evolver(tree = my_tree, partitions = my_partition)
my_evolver() # By default, the simulated alignment is saved to a file here
```

Figure 2.1: Example code for a simple codon simulation in Pyvolve. This example will simulate an alignment of 100 codons with a dN/dS of 0.75 and a κ (transition-tranversion mutational bias) of 3.25. By default, sequences will be output to a file called "simulated_alignment.fasta", although this file name can be changed, as described in Pyvolve's user manual.

In the example shown in Figure 2.1, stationary frequencies are not explicitly specified. Under this circumstance, Pyvolve will assume equal frequencies, although they would normally be provided using the key `"state_freqs"` in the dictionary of parameters. Furthermore, Pyvolve contains a convenient module to help specify state frequencies. This module can read in frequencies from an existing sequence and/or alignment file (either globally or from specified alignment columns), generate random frequencies, or constrain frequencies to a given list of allowed states. In addition, this module will convert frequencies between alphabets, which is useful, for example, if one wishes to simulate amino-acid data using the state frequencies as read from a file of codon sequence data.

2.4 Validating Pyvolve

We carefully assessed that Pyvolve accurately simulates sequences. To this end, we simulated several data sets under a variety of evolutionary models and conditions and compared the observed substitution rates with the simulated parameters.

To evaluate Pyvolve under the most basic of conditions, site-homogeneity, we simulated both nucleotide and codon data sets. We evolved nucleotide sequences under the JC69 model [52] across several phylogenies with varying branch lengths (representing the substitution rate), and we evolved codon sequences under a MG94-style model [79] with varying values of dN/dS. All alignments were simulated along a two-taxon tree and contained 100,000 positions. We simulated 50 replicates for each branch length and/or dN/dS parameterization. As shown in Figures 2.2A and B, the observed number of changes agreed precisely with the specified parameters.

We additionally validated Pyvolve's implementation of site-wise rate heterogeneity. We simulated an alignment of 400 codon positions, again under an MG94-style model [79], along a balanced tree of 2^{14} taxa with all branch lengths set to 0.01. This large number of taxa was necessary to achieve accurate estimates for site-specific measurements. To incorporate site-specific rate heterogeneity, we specified four dN/dS values of 0.2, 0.4, 0.6, and 0.8, to be assigned in equal proportions to sites across this alignment. We counted the observed dN/dS values for each resulting alignment column using a version of the Suzuki-Gojobori algorithm [113] adapted for phylogenetic data [57]. Figure 2.2C demonstrates that Pyvolve accurately implements site-specific rate heterogeneity. The high variance seen in Figure 2.2C is an expected result of enumerating substitutions on a site-specific basis, which, as a relatively small data set, produces substantial noise.

Finally, we confirmed that Pyvolve accurately simulates branch heterogeneity. Using a four-taxon tree, we evolved codon sequences under an MG94-style model [79] and specified a distinct dN/dS ratio for each branch. We simulated 50 replicate alignments of 100,000 positions, and we computed the observed dN/dS value along each branch. Figure 2.2D shows that observed branch dN/dS values agreed with the simulated values.

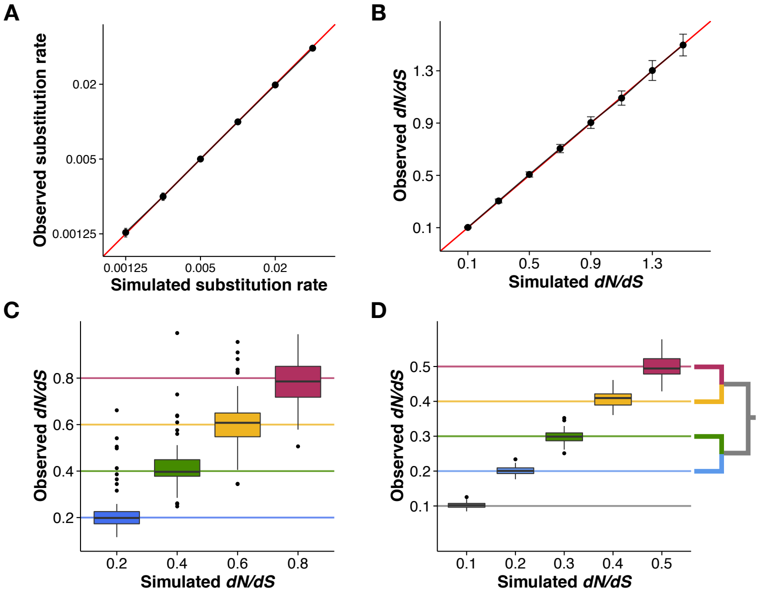


Figure 2.2: Pyvolve accurately evolves sequences under homogenous, site-wise rate heterogeneity, and branch-specific rate heterogeneity. A) Nucleotide alignments simulated under the JC69 [52] model along two-taxon trees with varying branch lengths, which represent the substitution rate. Points represent the mean observed substitution rate for the 50 alignment replicates simulated under the given value, and error bars represent standard deviations. The red line indicates the $x = y$ line. B) Codon alignments simulated under an MG94-style [79] model with varying values for the dN/dS parameter. Points represent the mean dN/dS inferred from the 50 alignment replicates simulated under the given dN/dS value, and error bars represent standard deviations. The red line indicates the $x = y$ line. C) Site-wise heterogeneity simulated with an MG94-style [79] model with varying dN/dS values across sites. Horizontal lines indicate the simulated dN/dS value for each dN/dS category. D) Branch-wise heterogeneity simulated with an MG94-style [79] model with each branch evolving according to a distinct dN/dS value. Horizontal lines indicate the simulated dN/d value for each branch, as shown in the inset phylogeny. The lowest dN/dS category ($dN/dS = 0.1$) was applied to the internal branch (shown in gray).

2.5 Conclusions

Because Pyvolve focuses on simulating the substitution processes using continuous-time Markov models along a fixed phylogeny, it is most suitable for simulating gene sequences, benchmarking inference frameworks, and for developing and testing novel Markov models of sequence evolution. For example, we see a primary application of Pyvolve as a convenient simulation platform to benchmark dN/dS and mutation–selection model inference frameworks such as the ones provided by PAML [127], HyPhy [58], Phylobayes [98], or swMutSel [116]. Indeed, the Pyvolve engine has already successfully been applied to investigate the relationship between mutation–selection and dN/dS modeling frameworks and to identify estimation biases in certain dN/dS models [109]. Moreover, we believe that Pyvolve provides a convenient tool for easy incorporation of complex simulations, for instance those used in approximate Bayesian computation (ABC) or MCMC methods [8], into Python pipelines.

Importantly, Pyvolve is meant primarily as a convenient Python library for simulating simple Markov models of sequence evolution. For more complex evolutionary scenarios, including the simulation of entire genomes, population processes, or protein folding and energetics, we refer the reader to several more suitable platforms. For example, genomic process such as recombination, coalescent-based models, gene duplication, and migration, may be best simulated with softwares such as ALF [28], CoalEvol and SGWE [10], or EvolSimulator [12]. Simulators which consider the influence of structural and/or biophysical constraints in protein sequence evolution include CASS [43] or ProteinEvolver [9]. Similarly, the software REvolver [54] simulates protein sequences with structural domain constraints by recruiting profile hidden Markov models (pHMMs) to model site-specific substitution processes.

We additionally note that Pyvolve does not currently include the simulation of insertions and deletions (indels), although this functionality is planned for a future release. We refer readers to the softwares indel-Seq-Gen [110] and Dawg [22] for simulating nucleotide sequences, and we suggest platforms such as INDELible [41], phyloSim [107], or πBuss [14] for simulating coding sequences with indels.

In sum, we believe that Pyvolve's flexible platform and user-friendly interface will provide a helpful and convenient tool for the biocomputing Python community. Pyvolve is freely available from `https://github.com/sjspielman/pyvolve`, conveniently packaged with a comprehensive user manual and several example scripts demonstrating various simulation conditions. In addition, Pyvolve is distributed with two helpful Python scripts that complement Pyvolve simulations: one which implements the Suzuki-Gojobori [113] dN/dS counting algorithm adapted for phylogenetic data [57], and one which calculates dN/dS from a given set of mutation–selection model parameters as described in ref. [109]. Pyvolve is additionally available for download from the Python Package Index.

Chapter 3

The relationship between *dN/dS* and scaled selection coefficients

3.1 Introduction

This work was published previously in the journal *Molecular Biology and Evolution*.[1]

The oldest and most-widely used method to infer selection pressure in protein-coding genes calculates the evolutionary rate ratio dN/dS, which represents the ratio of non-synonymous to synonymous substitution rates. This metric indicates how quickly a protein's constituent amino acids change, relative to synonymous changes, and it is commonly used to identify protein sites that experience purifying selection ($dN/dS < 1$), evolve neutrally ($dN/dS \approx 1$), or experience positive, diversifying selection ($dN/dS > 1$) [50, 57, 81, 131]. In phylogenetic contexts, dN/dS is typically calculated using a maximum likelihood (ML) approach [42, 79, 81, 126]. ML methods assume a continuous time Markov model of sequence evolution, and since the introduction of Markov codon models in the 1990s, they have become a sta-

[1] S. J. Spielman and C. O. Wilke. The relationship between dN/dS and scaled selection coefficients. Mol. Biol. Evol., 32(4):1097-1108, 2015. C. O. Wilke helped to design the project and write the manuscript.

ple of comparative sequence analysis [see ref. [6] for a comprehensive review]. Throughout this paper, we will refer to these models as dN/dS-based models.

A second class of Markov models, known as mutation–selection (MutSel) models, are increasingly being viewed as a viable alternative to the dN/dS framework. While dN/dS-based models describe how quickly a protein's constituent amino acids change, MutSel models assess the strength of natural selection acting on specific mutations. Couched firmly in population-genetic theory, the MutSel framework estimates site-specific scaled selection coefficients $S = 2N_e s$, which indicate the extent to which natural selection favors, or disfavors, particular codon and/or amino acid changes [46, 100, 115, 130]. Although first introduced over 15 years ago [46], MutSel models have seen little use due to their high computational expense. Recently, however, several computationally tractable model implementations have emerged [98,116], allowing for the first time the potential for widespread adoption.

Over the course of twenty years development, dN/dS-based models have advanced to a high level of sophistication. These models can accommodate a variety of evolutionary scenarios, including synonymous rate variation [60,79,101] and episodic [59,78] and/or lineage-specific selection [56,129,134], and they can also incorporate information regarding protein structure and epistatic interactions [72,94,97,104,118]. This flexibility, along with accessible software implementations [30, 58, 127], makes dN/dS-based models an attractive analysis choice. On the other hand, some have argued that MutSel models, given their explicit basis in population-genetics theory and attention to site-specific amino-acid fitness differences, offer a more mechanistically realistic approach to studying coding-sequence evolution [46, 100, 115, 119]. Moreover, a growing body of literature has demonstrated that dN/dS estimates are particularly sensitive to violations in model assumptions, calling into question the general utility of dN/dS-based models [63, 75, 92, 95].

Although both MutSel and dN/dS-based models describe the same fundamental process of coding-sequence evolution along a phylogeny, it is unknown how these two modeling frameworks relate to one another. In particular, as these inference methods have been developed independently, it remains an open question whether or not parameter estimates from one model are comparable to those of the other model. As a consequence, although certain rhetorical arguments may be made in favor of using one method over another, there is currently no formalized, concrete rationale to guide researchers in their methodological choices. Elucidating the relationship between these complementary modeling frameworks will more precisely reveal under which circumstances the use of these models is justified and has great potential to reveal previously unrecognized model behaviors, limitations, and capabilities.

Here, we formalize the relationship between these two modeling frameworks by examining the extent to which their respective focal parameters, dN/dS and scaled selection coefficients, yield overlapping information about the evolutionary process. To this end, we derive a mathematical relationship between dN/dS and scaled selection coefficients. We find that dN/dS values can be precisely calculated from scaled selection coefficients, and that

dN/dS accurately captures the selective pressures indicated by a given distribution of scaled selection coefficients. Furthermore, we prove that, when synonymous mutations are neutral, it is only possible to recover $dN/dS \leq 1$ from scaled selection coefficients, demonstrating that MutSel models, which commonly assume a static fitness landscape, are inherently only able to model purifying selection. Therefore, these models would be an inappropriate analysis method if positive selection is expected. However, we also find that, when synonymous codons have different fitnesses and hence purifying selection acts on synonymous changes, it is possible to recover dN/dS values above 1, even though classical positive, diversifying selection is not occurring. Therefore, the dN/dS framework cannot distinguish between positive, diversifying selection on amino acids and purifying selection on synonymous changes.

Finally, this relationship provides a uniquely rigorous platform to examine the performance of dN/dS-based models. Typically, researchers evaluate performance of a given inference framework through simulations that adhere to the underlying model's assumptions [but see refs. [49, 73, 101, 105, 132]]. In particular, simulated data is usually generated according to the same model as the inference framework, allowing for a direct comparison between the true and estimated parameter values. While this strategy is critical for testing whether a model implementation behaves as expected, it cannot assess model performance when the data are generated under a different process than the one modeled in the inference framework. However, in real-world sequence analysis, the inference framework will never exactly match the data-generation process. Therefore, a more sensitive test of model performance would examine how a given method performs when data are simulated under different mechanistic processes, and how sensitive the method is to violations of its assumptions. Unfortunately, such an approach is typically infeasible, because the relationships between parameters of interest among distinct model classes are generally not known.

The relationship we establish here between dN/dS and selection coefficients allows us to overcome this limitation, as we can determine the true dN/dS value directly from MutSel model parameters. Thus, we can assess performance of dN/dS-based inference frameworks by simulating data with a MutSel model and then comparing inferred dN/dS ML estimates (MLEs) to dN/dS values computed from selection coefficients. Using this strategy, we find, for sequences evolved under a symmetric mutation model, that dN/dS values inferred in an ML framework agreed precisely with those calculated from scaled selection coefficients. However, as mutational asymmetry increases, dN/dS MLEs become increasingly biased away from their true values, under a variety of ML model parameterizations. Surprisingly, the ML model parameterization which produced the most accurate dN/dS estimates was never the model which exhibited the best fit to the data (measured by AIC and BIC), ultimately revealing that relying on model fit as a litmus-test for model performance can be an ineffective and misleading strategy.

3.2 Results and Discussion

3.2.1 Theoretical model

This section contains a re-derivation of results presented in ref. [46], reproduced here to introduce notation and to place the remainder of our work into context. We model sequence evolution using the Halpern-Bruno Mut-Sel modeling framework under the assumptions of a fixed effective population size N_e and constant selection pressure over time [46, 115, 119, 130]. This continuous-time reversible Markov process is governed by the 61×61 transition matrix $T(t) = e^{Qt}$, where the matrix $Q = q_{ij}$ gives the instantaneous substitution probabilities between all 61 sense codons, and diagonal elements of Q satisfy $q_{ii} = -\sum_{i \neq j} q_{ij}$. We assume that only single-nucleotide substitutions occur instantaneously.

Let f_i^{codon} be the fitness of codon i, and let the selection coefficient acting on a mutation from codon i to codon j be $s_{ij} = f_j^{\text{codon}} - f_i^{\text{codon}}$ [106,130]. The fixation probability for this mutation is [46, 53]

$$u_{ij} \approx \frac{2s_{ij}}{1 - e^{-2N_e s_{ij}}} = \frac{1}{N_e} \frac{2N_e s_{ij}}{1 - e^{-2N_e s_{ij}}}. \tag{3.1}$$

We further define $S_{ij} = 2N_e s_{ij}$ (although note that this value would equal $4N_e s_{ij}$ in diploids) as the scaled selection coefficient for this change [130]. The probability of a substitution from codon i to j is therefore

$$q_{ij} = N_e m_{ij} u_{ij} = m_{ij} \frac{S_{ij}}{1 - e^{-S_{ij}}}, \tag{3.2}$$

where m_{ij} is the codon mutation rate, which represents the rate at which codon i transitions to codon j [46, 106]. If we assume that m_{ij} only has nonzero entries for single-nucleotide changes, we can write it as $m_{ij} = \mu_{o_i t_j}$, where μ_{kl} is the per-nucleotide mutation rate, o_i is the origin (i.e., before mutation) nucleotide in codon i, and t_j is the target (i.e., after mutation) nucleotide in codon j.

We now show how S_{ij} can be written in terms of mutation rates and stationary (equilibrium) codon frequencies P_i. As this system satisfies detailed balance (reversibility) [46], we have

$$q_{ij} P_i = q_{ji} P_j. \tag{3.3}$$

From equations (3.2) and (3.3), we can write the ratio of substitution probabilities as

$$\frac{P_i}{P_j} = \frac{m_{ji} S_{ji} (1 - e^{-S_{ij}})}{m_{ij} S_{ij} (1 - e^{-S_{ji}})}. \tag{3.4}$$

Using $S_{ij} = -S_{ji}$, we find that

$$S_{ij} = \ln \left(\frac{P_j m_{ji}}{P_i m_{ij}} \right). \tag{3.5}$$

This equation, previously derived in ref. [46], establishes a relationship between scaled selection coefficients and the stationary codon frequencies of the Markov chain. Moreover, in the specific case of symmetric mutation rates $m_{ij} = m_{ji}$, we have $S_{ij} = \ln \left(P_j / P_i \right)$ [106].

3.2.2 Predicting *dN/dS* from scaled selection coefficients

We now derive respective expressions for average nonsynonymous and synonymous evolutionary rates, which we can divide to obtain the evolutionary rate ratio dN/dS. We write the mean nonsynonymous rate K_{N} as

$$K_{\mathrm{N}} = \sum_i \sum_{j \in \mathcal{N}_i} P_i q_{ij} \,, \tag{3.6}$$

where $\mathcal{N}_i$ is the set of codons that are nonsynonymous to codon i and differ from it by one nucleotide, and the substitution probability q_{ij} is defined in equation (3.2). To normalize K_{N}, we divide it by the number of nonsynonymous sites, which we calculate according to the mutational opportunity definition of a site [42, 126] as

$$L_{\mathrm{N}} = \sum_i \sum_{j \in \mathcal{N}_i} P_i m_{ij} \,. \tag{3.7}$$

Thus, we find that

$$dN = \frac{K_{\mathrm{N}}}{L_{\mathrm{N}}} = \frac{\sum_i \sum_{j \in \mathcal{N}_i} P_i q_{ij}}{\sum_i \sum_{j \in \mathcal{N}_i} P_i m_{ij}} \,. \tag{3.8}$$

Similarly, for dS, the mean synonymous evolutionary rate K_{S} per synonymous site L_{S}, we find

$$dS = \frac{K_{\mathrm{S}}}{L_{\mathrm{S}}} = \frac{\sum_i \sum_{j \in \mathcal{S}_i} P_i q_{ij}}{\sum_i \sum_{j \in \mathcal{S}_i} P_i m_{ij}} \,, \tag{3.9}$$

where $\mathcal{S}_i$ is the set of codons that are synonymous to codon i and differ from it by one nucleotide substitution. The quantities K_{S} and L_{S} are defined as in Eqs. (3.6) and (3.7) but sum over $j \in \mathcal{S}_i$ instead of $j \in \mathcal{N}_i$.

Moreover, under certain simplifying conditions, we can simplify the ratio given by equations (3.8) and (3.9) to a more intuitive interpretation. If we assume that all synonymous codons have equal fitness (i.e. synonymous mutations are neutral), the synonymous fixation rate satisfies $u_{ij|j \in \mathcal{S}_i} = 1/N_e$ [26], and hence the synonymous substitution probability becomes $q_{ij} = m_{ij}$.

If we further assume symmetric mutation rates, the value for dS reduces to 1, and dN/dS thus reduces to the mean nonsynonymous substitution rate. We additionally note that, if we further assume uniform mutation rates, dN/dS becomes simply the average nonsynonymous fixation rate.

3.2.3 MutSel models specifically describe purifying selection

We examined the relationship between dN/dS and scaled selection coefficients by simulating 200 distributions of amino acid scaled fitness values, $F_a^{\text{aa}} = 2Nf_a^{\text{aa}}$, from a normal distribution $\mathcal{N}(0, \sigma^2)$. We drew a unique σ^2 for each fitness distribution from a uniform distribution $\mathcal{U}(0, 4)$. Higher values for σ^2 correspond to larger fitness differences among amino acids, causing selection to act more strongly against nonsynonymous changes. Thus, high σ^2 values indicate strong purifying selection, low values indicate weak purifying selection, and $\sigma^2 = 0$ indicates that all amino acids are equally fit. We note that these F_a^{aa} quantities correspond exactly to the amino-acid propensity parameters estimated by currently available site-specific MutSel inference methods [98, 116].

We then converted each amino-acid fitness distribution to a corresponding set of codon fitnesses, as described in *Methods*. Briefly, for 100 of the distributions, we assumed that all synonymous codons had the same fitness, but for the other 100 distributions we allowed synonymous codons to have different fitnesses. In other words, the former 100 distributions do not incorporate purifying selection on synonymous changes whereas the latter 100 distributions do. Using equations (3.6) – (3.9), we computed dN/dS for each distribution of codon fitnesses. For these calculations, we assumed the symmetric mutation model HKY85 [47], which is specified by the parameters μ, the nucleotide mutation rate, and κ, the ratio of transitions to tranversions. Specifically, transitions occur at a rate $\mu\kappa$ and tranversions at a rate μ. We used $\mu = 10^{-6}$ for all simulations, while we selected a unique value for κ for each simulation from $\mathcal{U}(1, 6)$.

Under neutral evolution, we expect that $dN/dS = 1$, and as purifying selection increases in strength, dN/dS should correspondingly decrease. Therefore, we expect that dN/dS will decline with the variance (σ^2) of the distribution of amino acid fitness values. Indeed, we observed a strong, negative correlation between these quantities (Figure 3.1). The larger the fitness differences among amino acids (higher σ^2), the lower dN/dS, properly reflecting increased purifying selection. This correlation was much stronger for fitness distributions without synonymous selection (Figure 3.1A) than for those with synonymous selection (Figure 3.1B). This difference emerged because fitness differences among synonymous codons obscured underlying amino-acid fitness differences. Even so, selection on synonymous codons did not negate the significant correlation between dN/dS and overall selection strength.

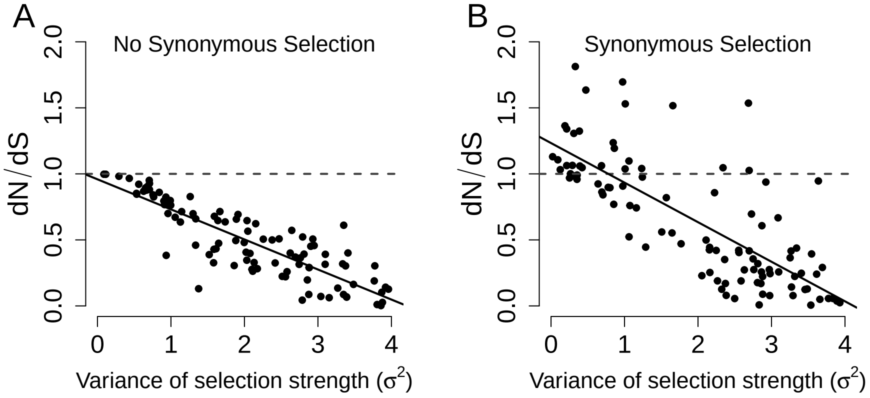


Figure 3.1: dN/dS decreases in proportion to amino-acid level selection strength. dN/dS is plotted against the variance (σ^2) of the simulated distribution of amino-acid scaled fitness values. Higher variances indicate larger fitness differences among amino acids, whereas the limiting value of $\sigma^2 = 0$ indicates that all amino acids have the same fitness. (A) Synonymous codons have equal fitness values ($r^2 = 0.83, P < 2^{-16}$). (B) Synonymous codons have different fitness values ($r^2 = 0.45, P < 2^{-16}$). Note that panel B, but not A, shows dN/dS values greater than 1, in spite of the steady-state evolutionary process. In each panel, the dashed line indicates the $y = 1$ line, and the solid line indicates the regression line.

Importantly, Figure 3.1A demonstrates that, in the limiting case when σ^2 approaches 0, and thus all codons have virtually the same fitness, dN/dS converges to 1. In other words, when the protein-coding sequence evolved neutrally, selection coefficients correctly yielded a $dN/dS \approx 1$. Furthermore, we never recovered $dN/dS > 1$ when synonymous changes were neutral, revealing a key property of Halpern-Bruno style MutSel models: they inherently cannot describe positive, diversifying selection. Indeed, in Appendix 1, we prove that, under the assumptions that synonymous changes are neutral and mutation is symmetric, scaled selection coefficients strictly yield $dN/dS \leq 1$. This proof formalizes this MutSel model's underlying assumption that selection pressure is constant over the phylogeny and that the protein evolves under equilibrium conditions. Although this proof assumes symmetric mutation rates, we have found numerically that dN/dS remains bounded from above by 1 even when mutations rates are asymmetric (Figure 3.4).

3.2.4 Purifying selection on synonymous changes can produce dN/dS > 1

The restriction $dN/dS \leq 1$ does not hold when synonymous changes are not neutral, as seen in Figure 3.1B. Even though the Halpern-Bruno model explicitly assumes that the system is at equilibrium [46, 119], we find that dN/dS can readily be greater than 1 under strong synonymous selection. In fact, it is theoretically possible to achieve arbitrarily high dN/dS values when synonymous codon substitutions carry fitness changes. In the most extreme case of synonymous selection, where only a single codon per amino acid is selectively tolerated, the number of synonymous changes becomes $K_{\text{S}} = 0$, and thus the value for dN/dS approaches infinity. Therefore, we find that $dN/dS > 1$ may indicate either positive, diversifying selection on amino acids or strong purifying selection on synonymous codons.

Given that the MutSel model framework assumes an overarching regime of purifying selection, this finding might seem paradoxical. However, the logical argument that $dN/dS > 1$ represents positive, diversifying selection assumes that the rate of synonymous change may be used as a neutral benchmark, an assumption clearly violated when selection acts on synonymous changes. Thus, the traditional signal of positive, diversifying selection, a dN/dS value in excess of one, can result simply from strong synonymous fitness differences.

That sequences evolving under purifying selection can spuriously bear the hallmark of positive, diversifying selection highlights the pitfalls of naively interpreting dN/dS values. Indeed, evolutionary constraints which induce synonymous selection are pervasive and affect virtually all domains of life [45], from viruses [27, 133, 136] to plants [44] to Metazoa [24, 38, 48, 65, 85]. Recent work has shown that synonymous rate variation is common across myriad proteins, and contributing to evolutionary rate heterogeneity in up to 42% of known protein families [32]. For example, exonic splicing enhancers [83, 84, 102], regions contributing to mRNA secondary structure such as translation-initiation sites [23, 27, 45, 102, 133], and DNA- and RNA-binding sites [83] all experience moderate to strong synonymous selection. It has additionally been suggested that up to 18% of mutational fitness effects in RNA viruses, whose genomes frequently feature sites with $dN/dS > 1$ [13, 21, 71, 72, 112], are caused by selection acting on synonymous changes [27]. Finally, both selection against protein mis-folding and for translation efficiency tend to induce synonymous selection in a gene-specific manner [4, 123], most notably in highly expressed genes [36, 65]. Therefore, while synonymous selection may not dominate genomes in organisms with relatively small effective population sizes [24, 85], it certainly acts strongly at specific sites and/or small, local regions. As dN/dS ratios are typically measured on a per-site basis, we expect that some sites with $dN/dS > 1$ are false positives in the detection of positive selection and instead represent cases of strong purifying selection on synonymous changes. We offer several approaches to ease this concern in *Conclusions*.

3.2.5 Relationship between dN/dS and scaled selection coefficients provides a novel benchmarking approach

The relationship we have established between dN/dS and scaled selection coefficients offers a unique opportunity to assess the robustness of dN/dS-based inference methods. It is conventional practice in model development to benchmark models against data simulated according to the model itself. While crucial for testing whether a given model has been correctly implemented, this strategy inherently cannot discern how the model behaves when data arose from a different mechanistic process. To overcome this limitation, we applied a novel benchmarking approach which used the theoretical relationship among modeling frameworks to assess the accuracy and specific utility of those models. Outlined in Figure 3.2A, this approach entails comparing dN/dS values calculated from selection coefficients to those inferred by a dN/dS-based model. Through this approach, we are able to simulate data which explicitly does not conform to the model used for inference, but we can still compare inferred parameter values to their true, simulated values using the relationship derived in the present work.

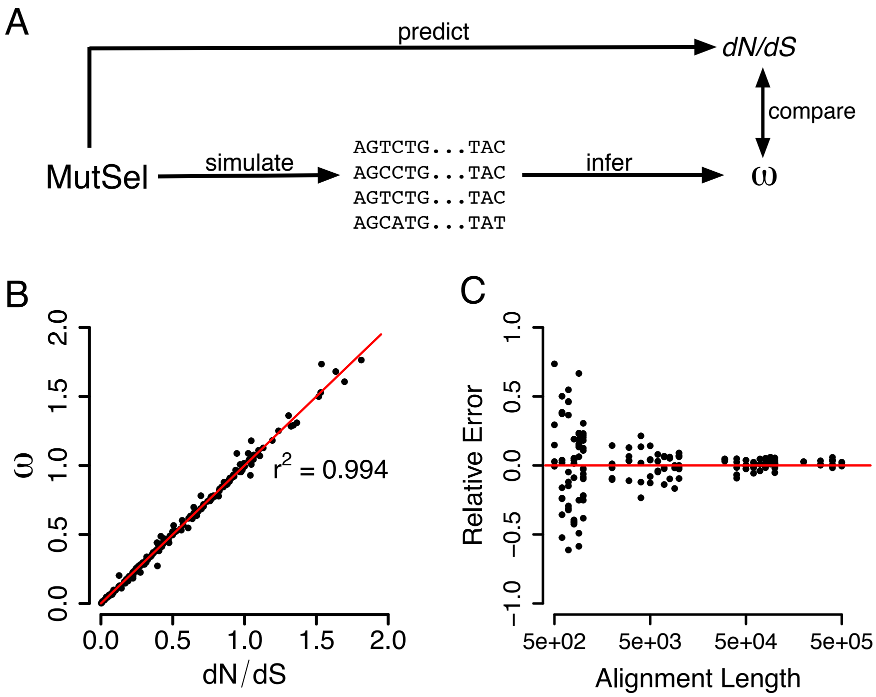


Figure 3.2: Combined modeling approach to assess performance of dN/dS inference frameworks. (A) Protein-coding alignments are simulated in the MutSel modeling framework. dN/dS can then be calculated ("predict") from scaled selection coefficients as well as through an ML inference framework ("infer"). Comparing resulting quantities reveals the accuracy of the chosen inference framework. (B) Regression between predicted dN/dS values and inferred ω MLEs. Each point corresponds to a single simulated alignment, and the solid line is the $x = y$ line. (C) Convergence of ω MLEs to the true dN/dS value. The y-axis indicates the relative error of the ω MLE, and the x-axis indicates the number of positions in the simulated alignment. As the number of positions and hence the size of the data set increases, ω converges to the predicted dN/dS value. The solid line is the $y = 0$ line, indicating no error.

Using the selection coefficients and symmetric mutation rates from the previous two subsections, we simulated alignments using standard methods [126] according to the Halpern-Bruno MutSel model [46]. We then inferred a dN/dS value for each alignment using the GY94 matrix [42, 81], which estimates dN/dS with the parameter ω. Throughout the remaining text, we refer to dN/dS inferred using ML as ω or ω maximum likelihood estimate (MLE), and to dN/dS computed using equations (3.6) – (3.9) simply as dN/dS.

We found that dN/dS values agree nearly perfectly with ω MLEs (Figure 3.2B), and indeed is relationship was robust to both synonymous selection and uneven nucleotide composition (simulated alignments featured GC contents ranging from 0.21 – 0.89). Additionally, Figure 3.2C demonstrates that ω converged to the true dN/dS value as the size of the data set (i.e., simulated alignment length) increased. These results unequivocally show that, when nucleotide mutation is symmetric, dN/dS-based model-inference methods behave exactly as expected, yielding precisely accurate dN/dS estimates. This finding has important implications for modeling choices; although the MutSel framework might model the sequence evolution in a way that more mechanistically matches the evolutionary process, dN/dS-based models may suffice to model selective forces in phylogenetic data.

3.2.6 Biased *dN/dS* estimates under asymmetric mutation models

We next sought to test the accuracy of dN/dS-based models using more realistic parameter values. To this end, we determined codon fitness distributions from 498 unique distributions of experimentally-derived, site-specific amino-acid fitnesses for H3N2 influenza nucleoprotein (NP) [16]. We combined each of these fitness distributions with three sets of experimentally-determined mutation rates, either for NP [16], yeast [137], or polio virus [2], to determine 498 × 3 = 1494 distinct distributions of steady-state codon frequencies (see *Methods* for details). While all three mutation matrices were asymmetric, each featured a differing degree of mutational bias; specifically, the mean ratios μ_{ij}/μ_{ji} for NP, yeast, and polio mutation rates are 1.03, 1.69, and 5.25, respectively. For each resulting set of stationary codon frequencies, in combination with its respective set of mutation rates, we calculated dN/dS and simulated alignments from which we inferred ω. Note that we assumed no selection on synonymous codons for these calculations.

dN/dS-based model matrices account for nucleotide mutational bias by incorporating either target codon [42] or target nucleotide [79] frequencies; these frameworks are known, respectively, as GY-style and MG-style models [55]. For example, the instantaneous rate matrix element giving the substitution probability from codon AAA to AAG would contain the target codon frequency P_{AAG} in GY-style models but the target nucleotide frequency π_{G} in MG-style models. Moreover, the GY-style models conform explicitly to a general-time reversible (GTR) form, whereas MG-style matrices do not, at first glance, appear to follow the same framework. However, as we show in Appendix 2, it is indeed posible to write MG-style matrices such that they conform to the GTR framework. This insight explicitly justifies using a time-reversible Markov process to describe these models, and it additionally demonstrates that the F1x4 codon frequency estimator [79] represents the state frequencies of the MG-style framework.

Previous works have suggested that MG-style and GY-style models yield different ω estimates [60, 132], so we inferred ω according to both GY- and MG-style frameworks. For GY-style models, we used the frequency estimators F61 [42], F3x4 [42], CF3x4 [55], and F1x4 [79]. For MG-style models, we considered both a parameterization with four global nucleotide frequency parameters and a parameterization which employed twelve nucleotide frequency parameters to allow for different frequencies at each codon position. We term the former framework MG1 and the latter MG3. Note that our MG1 corresponds to the original MG-style model [79], whereas our MG3 corresponds to the so-called MG94×HKY85 model [60].

Figure 3.3 shows the resulting relationships between dN/dS and ω MLEs for each set of mutation rates (NP, yeast, and polio), across model frequency parameterizations. Figure 3.3A displays the estimator bias, defined as the average difference between the true dN/dS value and the ω MLEs. Figure 3.3B displays the precision in this relationship, measured by the squared correlation coefficient r^2 between dN/dS and ω. The exact bias and r^2 values are given in Tables 3.1 and 3.2, respectively, and full regression plots for dN/dS vs. ω are shown in Figure 3.4.

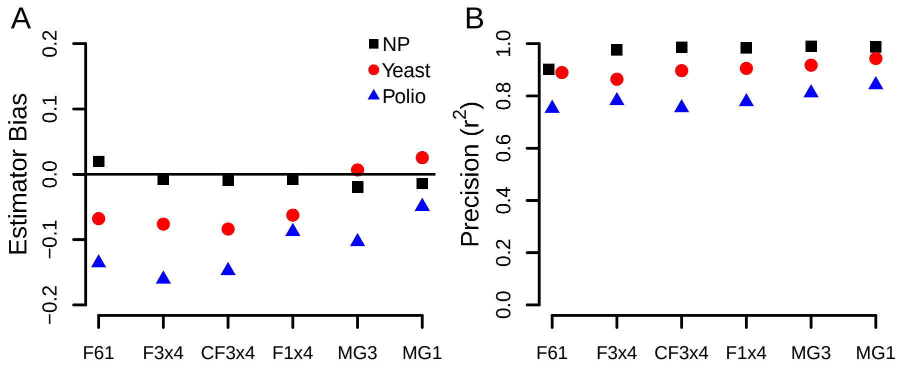


Figure 3.3: Estimator bias and precision of ω estimates for various model frequency parameterizations. (A) Estimator bias and (B) Precision (r^2) values between dN/dS and ω MLEs across model frequency parameterizations, for each set of nucleotide mutation rates. To calculate bias, we fit a linear model with ω as the response and dN/dS as the predictor, with a fixed slope of 1, and the resulting intercept value represents the bias. Negative biases indicate ω MLEs that are, on average, lower than dN/dS. Note that all standard errors for bias are smaller than the symbol size.

Two distinct trends emerge from Figure 3.3. First, asymmetry in the mutational process consistently induced significant bias in ω estimates. Most often, the model underestimated ω relative to the true dN/dS value. Based on simulations without any selection ($dN/dS = 1$), ref. [132] had previously suggested that GY-style models produce negatively biased ω estimates. Our results generalize this finding and show that this bias is pervasive, remains approximately constant through a wide range of dN/dS values, and is not limited to the GY-style framework (Figure 3.3A, Table 3.1, Figure 3.4). Furthermore, this bias systematically increased in magnitude as the underlying mutational process became more asymmetric. Indeed, for all frequency parameterizations, ω MLEs were most accurate under NP mutation rates, and both accuracy and precision tended to decrease as mutational bias progressed from yeast to polio mutation rates.

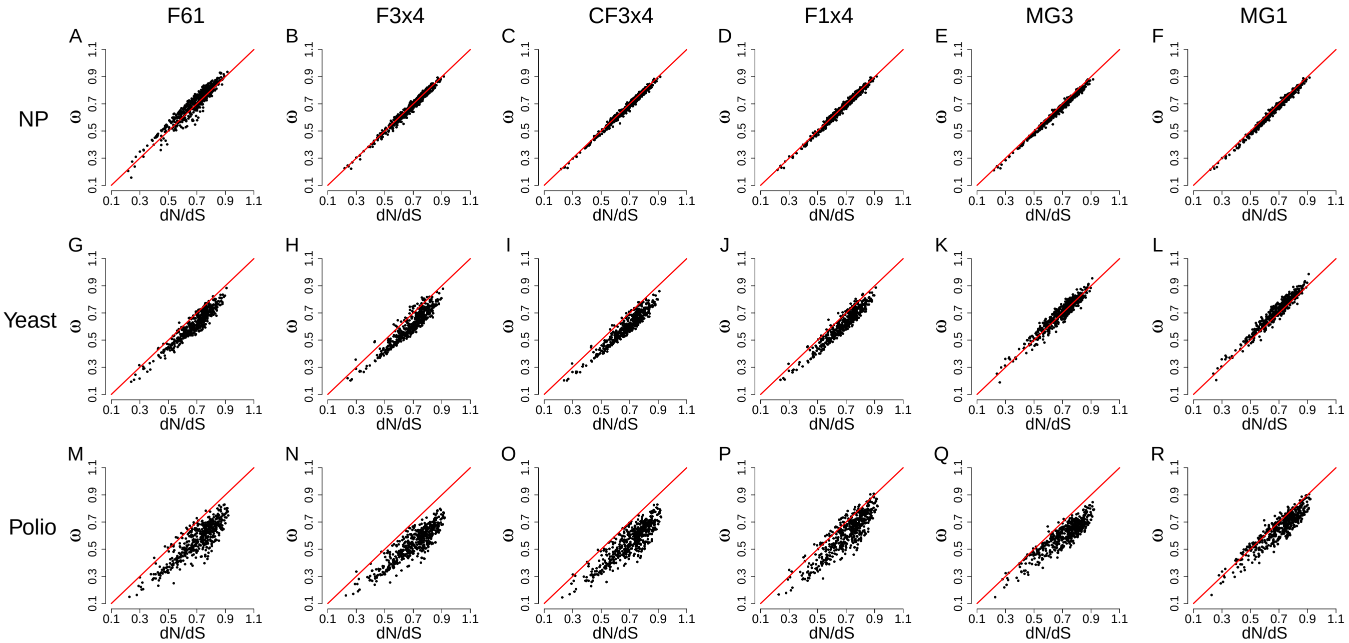


Figure 3.4: Regressions of ω MLEs on the true dN/dS values, as calculated from scaled selection coefficients, for datasets simulated using experimental fitnesses and mutation rates. Each point represents an alignment, and each red line is the $x = y$ line.

Table 3.1: Estimator bias of ω MLEs and the true dN/dS values, for all nucleotide mutation rates and model frequency parameterizations examined. All biases are statistically significant (different from 0), with all $P < 2 \times 10^{-16}$ except for the estimator bias associated with yeast mutation rates for MG3, where $P = 5.4 \times 10^{-5}$.

Mutation rate	MG1	F1x4	MG3	CF3x4	F3x4	F61
NP	-0.014	-0.02	-0.007	-0.009	-0.007	0.019
Yeast	0.025	0.007	-0.063	-0.084	-0.076	-0.068
Polio	-0.049	-0.103	-0.088	-0.148	-0.161	-0.136

Table 3.2: Precision, measured as the squared correlation coefficient r^2, of ω MLEs relative to the true dN/dS values, for all nucleotide mutation rates and model frequency parameterizations examined. All values shown are statistically significant, with all $P < 2 \times 10^{-16}$.

Mutation rate	MG1	F1x4	MG3	CF3x4	F3x4	F61
NP	0.988	0.989	0.985	0.986	0.977	0.902
Yeast	0.943	0.917	0.905	0.897	0.864	0.889
Polio	0.842	0.811	0.777	0.754	0.781	0.752

Second, frequency parameterizations which more closely matched the mechanistic process that generated the data (MG1 and MG3) generally outperformed all other frequency estimators. In particular, MG1 clearly performed the best of all frequency estimators considered, featuring by far the least amount of estimator bias for the highly asymmetric polio mutation rates. This result makes intuitive sense, as the MG-style framework most mechanistically matches the MutSel framework among all dN/dS-based frameworks examined here. Indeed, in the case of neutral evolution, $\omega = 1$ in an MG-style matrix, and the ratio of fixation probabilities in the MutSel matrix will also equal 1. Therefore, nucleotide mutation rates alone comprise each model's rate matrix, demonstrating that MG-style and MutSel models are virtually identical under neutral evolution. Importantly, this correspondence does not hold for GY-style matrices which, as they incorporate target codon frequencies, do not explicitly consider nucleotide mutation rates. Thus, we highly recommend that researchers employ MG-style matrices in their dN/dS inferences to minimize bias. We note that this modeling framework is available through HyPhy [58] and/or the Datamonkey server [30].

3.2.7 Model with best fit is not the model that yields the most accurate parameter estimates

Strikingly, when we examined model fit using AIC scores [5,20] for the different frequency parameterizations, we found that the F61 parameterization was unequivocally the best-performing model, on average, for all datasets (Table 3.3). This result dramatically juxtaposed the substantial inaccuracy and imprecision in ω that F61 frequently yielded. In particular, F61 had the most estimator bias for NP datasets as well as the least precision for both NP and polio datasets (Figure 3.3). Thus, we found AIC could not identify the model which produced the most accurate estimates for the parameter of interest.

Table 3.3: Mean ΔAIC for datasets simulated with NP, yeast, or polio virus mutation rates. The order of frequency models shown in the table corresponds to the model ranking for NP, and the ranking differs somewhat for yeast and polio datasets. AIC is computed as $\text{AIC} = 2(k - \ln L)$, where k is the number of free parameters of the model, and $\ln L$ is the log-likelihood [5,20]. Number of free parameters for each model is F61, 63; CF3x4, 12; MG1, 6; F1x4, 6; MG3, 12; and F3x4, 12. Note that, for each model, 3 of the parameters are ω, κ, and a global branch-length scaling parameter, and the remaining parameters are either empirical codon or nucleotide frequencies.

Frequencies	NP	Yeast	Polio
F61	0	0	0
CF3x4	-9519.53	-7843.77	-7975.94
MG1	-13207.5	-9924.05	-5147.57
F1x4	-13410.54	-13544.47	-15468.29
MG3	-14287.28	-12737.57	-8624.87
F3x4	-14699.22	-17277.3	-19384.58

Although this result may seem counterintuitive, it is important to note that AIC measures goodness-of-fit by approximating the Kullback-Leibler (KL) distance between a given candidate model and the true model. As the MutSel framework defines selection coefficients in terms of stationary frequencies, it indeed follows that the F61 estimator, which explicitly incorporates empirical codon frequencies into the rate matrix, should be selected as the best-fitting model, in spite of its biased parameter estimates. Therefore, we additionally assessed whether BIC might provide a more accurate indication of model performance. However, BIC scores, shown in Table 3.4, yielded the same overarching trend as did AIC scores in which F61 dramatically outperformed all other frequency parameterizations.

Table 3.4: Mean ΔBIC for datasets simulated with NP, yeast, or polio virus mutation rates. Note that the order of frequency models shown here corresponds to the model ranking for NP, and the ranking differs somewhat for yeast and polio datasets. BIC is computed as BIC $= -2 \ln L + k \ln n$, where k is the number of free parameters of the model, $\ln L$ is the log-likelihood, and n is the sample size [20]. For all models, $n = 500000$, which corresponds to the number of alignment columns. The number of free parameters for each model is F61, 63; CF3x4, 12; MG1, 6; F1x4, 6; MG3, 12; and F3x4, 12. Note that, for each model, 3 of the parameters are ω, κ, and a global branch-length scaling parameter, and the remaining parameters are either empirical codon or nucleotide frequencies.

Frequencies	NP	Yeast	Polio
F61	0	0	0
CF3x4	-8918.92	-7243.16	-7306.7
MG1	-12551.28	-9267.83	-4399.6
F1x4	-12776.56	-12910.5	-14720.32
MG3	-13653.31	-12103.59	-7955.63
F3x4	-14098.61	-16676.69	-18715.34

This finding has broad implications for practices in model selection. In particular, it appears that model fit can be confounded with model accuracy, such that the model with better model fit may produce less accurate parameter estimates. We find that, if the data are generated by a process distinct from the inference model, standard model selection quantities cannot necessarily identify which model produces the most precise and least biased parameter estimates. Good model fit, therefore, may not have any bearing on whether using that model is mechanistically justified, and selecting models based solely on fit may not guard effectively against spurious inferences but instead prove misleading. We suggest that the mechanism producing the data should be carefully considered, and an appropriate inference method which best approximates this process should then be selected.

Finally, these results provide a concrete example of previous theoretical suggestions that AIC might fail in phylogenetic model selection [68]. Indeed, previous work has suggested that Bayes Factors might serve as a better indication of model performance than AIC, albeit results were obtained in a Bayesian rather than frequentist framework [99]. Further investigation into the performance of various model fit criteria for model selection is strongly warranted.

3.3 Conclusions

By elucidating the relationship between dN/dS and scaled selection coefficients, we have shown that dN/dS-based and MutSel models convey consistent information regarding the strength of natural selection. Importantly, our proof that $dN/dS \leq 1$ (assuming symmetric mutation and no synonymous selection) indicates that the use of the Halpern-Bruno MutSel modeling framework is only justified under purifying selection. This restriction is in part indicated by this model's assumption of constant selection pressures over time, or in other words a static fitness landscape [46, 100, 118, 119]. Thus, if the aim is to identify positive, diversifying selection, of the two frameworks examined here, only dN/dS-based models are appropriate. However, different MutSel frameworks not examined here, which allow fitnesses to fluctuate over time, should serve as a promising avenue for future research extending the applicability of this modeling framework [80, 122].

We have also found that dN/dS values can readily be greater than 1 when selection acts on synonymous mutations, even though the protein sequence is evolving solely under purifying selection. This seemingly paradoxical finding actually reflects an assumption violation; the assertion that $dN/dS > 1$ necessarily corresponds to positive, diversifying selection requires that synonymous changes are neutral, which clearly does not hold if there are fitness differences among synonymous codons. This result contributes to a growing body of literature which has found that purifying selection can yield $dN/dS > 1$ if model assumptions are not met. For instance, dN/dS can be greater than 1, even under purifying selection, if sequences contain segregating polymorphisms [63,75,95] or when GC-biased gene conversion is pervasive [92]. Thus, it is becoming increasingly clear that the $dN/dS = 1$ neutral threshold typically used to distinguish purifying and positive selection is highly sensitive to violations in model assumptions. We emphasize that it is crucial to verify that data adhere to model assumptions before conclusions from dN/dS are drawn.

We suggest several strategies to limit such false positive results under synonymous selection. For one, certain formulations of dN/dS-based methods consider dN and dS rate variation separately [60, 70, 77, 79] rather than using a single parameter to represent dN/dS. These kinds of methods, and indeed others which explicitly model nucleotide-level selection in conjunction with codon-level selection [101] or correct dS for synonymous selection [135], may be able to distinguish situations in which $dN/dS > 1$ because dN is unusually large (positive selection) or dS is unusually small (purifying selection). Further, our benchmarking approach, in which we simulate sequences according to MutSel models and infer dN/dS both from MutSel parameters directly and using ML, may be used to benchmark these kinds of models and may help to identify circumstances under which synonymous selection confounds dN/dS interpretations.

Finally, we emphasize the utility of establishing relationships among distinct modeling frameworks to probe model behavior and evaluate model performance. Such an approach is uniquely able to reveal unrecognized behaviors and/or limitations of different modeling frameworks and can precisely reveal the circumstances in which different models are best suited. We hope that further studies in this spirit will ensure robust model development in future studies.

3.4 Methods

3.4.1 Simulation of scaled selection coefficients

To examine the relationship between dN/dS and scaled selection coefficients, we simulated 200 distributions of amino-acid scaled fitness values, $F_a^{\text{aa}} = 2Nf_a^{\text{aa}}$, from a normal distribution $\mathcal{N}(0, \sigma^2)$, where a unique σ^2 for each fitness distribution was drawn from a uniform distribution $\mathcal{U}(0, 4)$. We converted these amino-acid fitnesses to codon fitnesses as follows. For 100 of the fitness distributions, we directly assigned all codons within a given amino acid family the fitness $F_i^{\text{codon}} = F_a^{\text{aa}}$, so that all synonymous codons had the same fitness. For the other 100 fitness distributions, we assigned synonymous codons different fitnesses by randomly drawing a preferred codon for each amino acid. This preferred codon was assigned the fitness of $F_i^{\text{codon}} = F_a^{\text{aa}} + \lambda$, and all non-preferred codons were given the fitness $F_j^{\text{codon}} = F_a^{\text{aa}} - \lambda$. We drew a unique λ for each fitness distribution from $\mathcal{U}(0, 2)$. We then computed stationary codon frequencies as

$$P_i = \frac{e^{F_i^{\text{codon}}}}{\sum_k e^{F_k^{\text{codon}}}}, \tag{3.10}$$

where the sum in the denominator runs over all 61 sense codons [106]. Equation (3.10) gives the analytically precise stationary frequencies for a MutSel model, under the assumption of symmetric mutation rates [106]. We used equations (3.6) – (3.9) to compute dN/dS for each resulting set of stationary codon frequencies. For these calculations, we assumed the HKY85 [47] nucleotide mutation model, and accordingly we set the transition mutation rate as $\mu\kappa$ and the transversion rate as μ. We used the value $\mu = 10^{-6}$ for all dN/dS calculations, and we drew a unique value for κ from $\mathcal{U}(1, 6)$ for each set of codon frequencies.

3.4.2 Alignment simulations

We simulated protein-coding sequences as a continuous-time Markov process using standard methods [126] according to the Halpern-Bruno Mut-Sel model [46]. In brief, this model's instantaneous rate matrix $Q = q_{ij}$ is populated by elements

$$q_{ij} = \begin{cases} \mu_{ij}\frac{S_{ij}}{1-\exp(S_{ji})} & \text{single nucleotide change} \\ \\ 0 & \text{multiple nucleotide changes} \end{cases}, \tag{3.11}$$

for a mutation from codon i to j, where m_{ij} is the mutation rate, and the scaled selection coefficient S_{ij} is defined in equation (3.5). All alignments presented here were simulated along a symmetric 4-taxon phylogeny with all branch lengths equal to 0.01, beginning with a root sequence generated in proportion to stationary codon frequencies [126]. Unless otherwise stated, all simulated alignments contained 500,000 codon positions. A single evolutionary model was applied to all positions in the simulated sequences. While this lack of site-wise heterogeneity is unrealistic for real sequence evolution, it allowed us to verify our derived relationship between scaled selection coefficients and dN/dS with a sufficiently sized data set.

3.4.3 Computation of stationary frequencies for experimental data sets

We used experimentally-determined site-specific amino-acid fitness parameters F_a^{aa} for influenza nucleoprotein (NP), from ref. [16], in combination with experimental nucleotide mutation rates for either NP [16], yeast [137], or polio virus [2], to derive realistic distributions of stationary codon frequencies. We combined each of the 498 site-wise amino-acid preference sets reported by ref. [16] with each of the three mutation-rate matrices to construct a total of $498 \times 3 = 1494$ unique experimental evolutionary Markov models, using the approach in refs. [16,17]. The instantaneous rate matrix Q for each experimental model is populated by elements

$$q_{ij} = \begin{cases} \max(1, F_j^{\text{codon}} \big/ F_i^{\text{codon}}) m_{ij} & \text{single nucleotide change} \\ 0 & \text{multiple nucleotide changes} \end{cases} \tag{3.12}$$

for a substitution from codon i to codon j, where F_i^{codon} is the fitness of codon i [16,17]. We calculated F_i^{codon} values by simply assigning a given amino acid's experimental fitness F_a^{aa} to each of its constituent codons; thus, all synonymous changes were neutral. We determined the stationary codon frequencies for each resulting experimental model from the matrix's eigenvector corresponding to the eigenvalue 0. Finally, we simulated alignments for each set of stationary frequencies and corresponding mutation rates according to the Halpern-Bruno model (equation (3.11)).

3.4.4 Maximum likelihood inference of *dN/dS*

For the 200 alignments simulated with symmetric mutation rates, we inferred dN/dS using the M0 model [131], as implemented in the HyPhy batch language [58]. The M0 model uses the GY94 instantaneous rate matrix, which is populated by elements

$$q_{ij} = \begin{cases} P_j & \text{synonymous transversion} \\ \kappa P_j & \text{synonymous transition} \\ \omega P_j & \text{nonsynonymous transversion} \\ \omega\kappa P_j & \text{nonsynonymous transition} \\ 0 & \text{multiple nucleotide changes} \end{cases}, \tag{3.13}$$

for a substitution from codon i to codon j, where κ is the transition-transversion bias, P_j is the equilibrium frequency of the target codon j, and ω represents dN/dS [42, 81]. The P_i parameters are intended to represent those codon frequencies which would exist in absence of selection pressure generated by mutation alone [42, 79, 126, 128]. Thus, when inferring ω on datasets which used symmetric mutation rates, we assigned the value $1/61$ to all parameters P_i, as all codons are equally probable under unbiased mutation.

Alternatively, when inferring ω for alignments simulated with experimental fitness and mutation rates, we used several different model parameterizations, including GY-style [42] (target codon frequency) and MG-style [79] (target nucleotide frequency) parameterizations. We considered the GY-style parameterizations F61 [42], F3x4 [42], CF3x4 [55], and F1x4 [79]. We implemented two varieties of MG-style models; the first, MG1, employs four parameters for nucleotide frequencies (one per nucleotide) [79], and the second, MG3, employs twelve nucleotide frequency parameters, with four nucleotide frequency parameters for each of the three codon positions [60]. All frequency parameters were estimated from the data. Note that we used the state frequencies of F1x4 for the MG1 framework and F3x4 for the MG3 framework.

In addition to frequency parameter, all models included the parameters κ and ω.

3.4.5 Availability

All code is freely available from `https://github.com/clauswilke/Omega_MutSel`. Simulated alignments are available from the Data Dryad repository at `http://doi.org/10.5061/dryad.51sq0`.

3.5 Appendix 1

We prove that $dN/dS \leq 1$ when calculated from scaled selection coefficients. We assume that mutation rates are symmetric ($m_{ij} = m_{ji}$) and that synonymous codons have the same fitness (synonymous changes are neutral). As described in the main text, these assumptions yield $dS = 1$, and hence we have to show that $dN = K_{\mathrm{N}} \big/ L_{\mathrm{N}} \leq 1$. To this end, we note that the sums in K_{N} and L_{N} can be reordered such that the substitution probability from codon i to j is always added to the substitution probability from codon j to i. We can then show that the sum of each of these pairs in the expression for K_{N} is smaller than the corresponding term in L_{N}, and hence $dN/dS \leq 1$.

Without loss of generality, we consider a pair of nonsynonymous codons i and j whose respective stationary frequencies P_i and P_j satisfy $P_i \leq P_j$ and $P_i, P_j \geq 0$. As follows from equations (3.2) and (3.5), the sum of the probability weights of evolving from codon i to j and from codon j to i is

$$N_e m_{ij} u_{ij} + N_e m_{ji} u_{ji} = \frac{2P_i P_j[\log(P_i) - \log(P_j)]}{P_i - P_j}\,. \tag{3.14}$$

This quantity represents K_N in the dN calculation. To prove $dN \leq 1$, we must show that this quantity is less than or equal to $P_i + P_j$, which represents L_N in the dN calculation. To this end, we introduce the function

$$F(x,y) = x + y - \frac{2xy[\log(x) - \log(y)]}{x - y}\,, \tag{3.15}$$

and we will now show that $F(x,y) \geq 0$ for $x \leq y$ and $y \geq 0$. Using l'Hôpital's rule, it is straightforward to show that $\lim_{|x-y|\to 0} F(x,y) = 0$. Thus, we can define $F(x,x) \equiv 0$. For $x < y$, we show that the first derivative of equation (3.15) is negative throughout $x \in (0,y)$, which proves that the function monotonically decreases, and thus $F(x,y) > 0$, in this interval. We calculate the first derivative as

$$\frac{\partial F(x,y)}{\partial x} = \frac{\big[(x-3y)(x-y) - 2y^2(\log x - \log y)\big]}{(x-y)^2}\,. \tag{3.16}$$

We now replace the expression $\log x - \log y$ by its Taylor series expansion, yielding

$$\frac{\partial F(x,y)}{\partial x} = \frac{\left[(x-3y)(x-y) - 2y^2\left(\sum\limits_{n=1}^{\infty} \frac{1}{n}(1 - x/y)^n\right)\right]}{(x-y)^2}\,. \tag{3.17}$$

We note that the first two terms of the Taylor series equal $(x-3y)(x-y)$, and thus expression (3.17) simplifies to

$$\frac{\partial F(x,y)}{\partial x} = \frac{-2y^2 \sum\limits_{n=3}^{\infty} \frac{1}{n}\left(1 - \frac{x}{y}\right)^n}{(x-y)^2}, \tag{3.18}$$

which is clearly negative. This concludes the proof.

3.6 Appendix 2

GY-style matrices may be expressed in the framework of the general-time reversible (GTR) model, in which the instantaneous matrix Q can be decomposed into a 61×61 symmetric substitution rate matrix and a 61-dimensional vector containing the equilibrium codon frequencies. The latter corresponds to the stationary distribution of the Markov chain. By contrast, MG-style rate matrices are written in terms of nucleotide frequencies rather than codon frequencies. Therefore, whether these models fit into the GTR framework is unclear *a priori*. We now describe how the MG-style matrix can be rewritten in terms of a symmetric matrix and a vector of equilibrium codon frequencies, thus demonstrating that these matrices also fit into the GTR framework.

MG-style matrix elements, for a the substitution from codon i to j, are generally given by

$$q_{ij} = \begin{cases} \theta_{o_i t_j}\pi_{t_j} & \text{synonymous change} \\ \omega\theta_{o_i t_j}\pi_{t_j} & \text{nonsynonymous change} \\ 0 & \text{multiple nucleotide changes} \end{cases}, \tag{3.19}$$

where ω is the ratio of nonsynonymous to synonymous substitution rates and the product $\theta_{o_i t_j}\pi_{t_j}$ corresponds to a nucleotide-level mutation rate $\mu_{o_i t_j}$, where o_i is the origin nucleotide in codon i, and t_j is the target nucleotide in codon j. Note that the matrix $\theta_{o_i t_j}$ is symmetric in o_i and t_j.

For a given codon i, the matrix of Eq. (3.19) yields the stationary frequency $P_i = \pi_{i_1}\pi_{i_2}\pi_{i_3}C$, where $C = 1 - \Pi_{\text{stop}}$ and $\Pi_{\text{stop}} = \pi_{\text{T}}\pi_{\text{A}}\pi_{\text{G}} + \pi_{\text{T}}\pi_{\text{G}}\pi_{\text{A}} +$

$\pi_{\mathrm{T}}\pi_{\mathrm{A}}\pi_{\mathrm{A}}$ [79]. Therefore, we can rewrite the term $\theta_{o_i t_j}\pi_{t_j}$ as $\theta_{o_i t_j}P_j C/(\pi_m \pi_n)$, where m and n are the nucleotides which do not change in a given instantaneous codon substitution. This allows us to rewrite the rate instantaneous matrix as

$$q_{ij} = \begin{cases} \frac{C\theta_{o_i t_j}}{\pi_m \pi_n} P_j & \text{synonymous change from } i \text{ to } j \\ \omega \frac{C\theta_{o_i t_j}}{\pi_m \pi_n} P_j & \text{nonsynonymous change from } i \text{ to } j \\ 0 & \text{multiple nucleotide changes} \end{cases} \tag{3.20}$$

for a substitution from codon i to codon j, and this matrix clearly conforms to the GTR framework.

Chapter 4

Extensively-parameterized mutation–selection models reliably capture site-specific selective constraint

4.1 Introduction

Proteins are subject to a variety of structural, functional, and physiochemical constraints that influence their evolutionary trajectories. A growing body of research has demonstrated that these constraints lead individual protein sites to have distinct tolerances to different amino acids [1,11,16,17,34,39, 86,87,91,93]. Recent experimental studies have further demonstrated that, for essential house-keeping proteins, site-wise amino-acid preferences are broadly conserved over evolutionary time [11, 34, 93].

To achieve a complete picture of protein evolutionary dynamics, it is critical that we employ robust sequence evolution frameworks which explicitly incorporate site-specific amino acid propensities. One such evolutionary model that achieves this goal, known as the mutation–selection model, is based on fundamental population-genetics principles [26] and models the joint forces of selection and mutation in protein-coding sequences along a phylogeny. This model considers site-specific amino-acid and/or codon propensities, or fitness values, as its focal parameters [46, 118, 119, 130]. Specifically, the mutation–selection model estimates the scaled fitness, $F = 2N_ef$ (or $F = N_ef$ for haploid organisms), where N_e is the effective population size and f is the Malthusian fitness, of each amino acid at a given position in a protein-coding sequence. These fitnesses are often used to infer the distribution of scaled selection coefficients $S_{ij} = F_j - F_i$, where F_i and F_j are the scaled fitnesses of amino acids i and j. The distribution of S values indicates the range of selective responses to new mutations across a given protein sequence.

Recently, two alternative implementations of site-specific mutation–selection models have been released. The first implementation, known as swMutSel, estimates site-specific fitness parameters as fixed-effect variables through a maximum penalized-likelihood (MPL) approach [115, 116]. The second implementation, available in the PhyloBayes software package, instead employs a Dirichlet Process (DP) Bayesian framework and models site-specific fitness parameters as random effects [98, 100]. For simplicity, we will refer to the latter implementation as "pbMutSel" throughout this paper. Both platforms are based on the mutation–selection models introduced by ref.
[46] and ref. [130], and they make nearly identical assumptions about the evolutionary process. For instance, both swMutSel and pbMutSel assume that sites evolve independently, that there is no selection on synonymous codons (i.e. all synonymous codons have the same fitness), and that nucleotide mutation rates are shared across all sites.

Although the mutation–selection model provides a promising framework for modeling protein sequence evolution in a mechanistic context, it is not yet clear how one might use its estimates to gain insight into the evolutionary process. Whether the amino-acid fitnesses estimated by either swMutSel or pbMutSel truly reflect evolutionary constraint remains an open question, in particular because these two implementations produce seemingly-incompatible results: swMutSel infers S distributions with two peaks representing nearly-neutral (centered at $S = 0$) and highly deleterious changes, commonly defined as $S < -10$ in the context of mutation–selection models [96, 115, 116]. In contrast, pbMutSel infers unimodal distributions centered at $S = 0$, without a peak of highly deleterious changes.

The relative accuracy between these two distinct approaches has sparked a lively debate in the literature [96,98,103,116]. Specifically, ref. [96] critiqued early swMutSel implementations as suffering from overparameterization, as swMutSel's fixed-effects framework requires estimating 19 parameters per site. He argued that the characteristic peak at $S < -10$ in swMutSel-inferred scaled selection-coefficient distributions is an erroneous artifact of model overparameterization. ref. [96] additionally contended that, by modeling fitnesses as random effects, pbMutSel avoids overfitting and certain statistical inconsistencies that extensive parameterization might introduce. In response, ref. [116] argued that experimental evidence from population genetics literature supports swMutSel's recovery of a prominent peak of highly deleterious $S < -10$ changes. To ameliorate potential overfitting artifacts, swMutSel has been updated with several likelihood penalty functions that regularize extreme amino-acid fitness estimates [116].

Importantly, quantitative comparisons of swMutSel and pbMutSel inferences have focused nearly exclusively on asking how well they recapitulate the gene-wide distribution of S, or similarly the gene-wide proportions of deleterious and beneficial substitutions [96, 98, 100, 115, 116]. In spite of these efforts, however, there remains no conclusive evidence supporting either swMutSel or pbMutSel as the more reliable inference approach. Indeed, support for either approach currently rests on theoretical arguments regarding either pbMutSel's more desirable statistical properties, or swMutSel's general agreement with population-genetics literature. However, statistical consistency does not necessarily correspond to empirical accuracy, and phylogenetic data may not be directly comparable to population data. As such, neither argument presents strong evidence in favor of either pbMutSel or swMutSel.

We posit that no consensus regarding mutation–selection implementation accuracy has emerged specifically because performance has been assessed using whole-gene S distributions. Pooling all site-specific S values into a single distribution makes it impossible to conduct a systematic analysis of differences between inference methods, especially given that these methods were implemented to estimate amino-acid fitness values at individual sites. As a consequence of this approach, it remains unknown how well inferred parameters capture site-specific evolutionary processes.

Therefore, in this study, we have investigated the relative performance of mutation–selection model implementations by directly comparing how well each infers evolutionary constraints across individual sites, rather than focusing primarily on S distributions. We have found that swMutSel, specifically run as either unpenalized or with a weak likelihood penalty function, consistently estimates the most accurate site-specific fitness values. By contrast, pbMutSel and strongly-penalized swMutSel parameterizations systematically underestimates the strength of natural selection across sites.

4.2 Results

4.2.1 Simulation and Inference Approach

We simulated eleven coding-sequence alignments wherein each position evolved according to a distinct mutation–selection model parameterization. We derived site-specific codon fitness parameters from a set of structurally-curated yeast amino-acid alignments from ref. [91], as described in *Methods and Materials.* Each simulation was performed using parameters derived from a specific yeast alignment, and thus the number of codon positions across simulated alignments ranged from 115–291. This approach ensured that the evolutionary heterogeneity across each simulated alignment was directly comparable to that seen in real protein alignments. We assumed that all codons for a given amino acid had the same fitness, and we assumed globally equal mutation rates. All simulations were performed along a balanced 512-taxon tree with branch lengths of 0.5 so that that each dataset contained sufficient information to discern the underlying stationary amino-acid fitnesses (Spielman et al. 2015).

We processed each simulated alignment with both swMutSel and pbMutSel. For swMutSel, we processed each alignment both without a penalty and under six penalty functions [116]. Penalty functions examined included the multivariate normal penalty function with the σ parameter equal to either 1, 10, or 100 (referred to as mvn1, mvn10, and mvn100, respectively), as well as the Dirichlet-based penalty function with the α parameter equal to either 1.0, 0.1, or 0.01 (referred to as d1.0, d0.1, and d0.01, respectively). Each set of penalty-function parameterizations represents stronger to weaker penalties, i.e. mvn1 is a strong penalty, mvn10 is a moderate penalty, and mvn100 is a weak penalty. Similarly, d1.0 is a strong penalty, d0.1 is a moderate penalty, and d0.01 is a weak penalty. Unless otherwise stated, we refer to swMutSel inferences using their penalty specification and to swMutSel run without a penalty function as "unpenalized."

4.2.2 No method can infer the true distribution of selection coefficients

We began our analysis by assessing how well each inference approach estimated the true, simulated distribution of scaled selection coefficients, S. Qualitatively, either unpenalized or weakly-penalized swMutSel (specifically mvn100 or mvn10) best captured the shape of the underlying S distribution (Figures 4.1). However, a more rigorous comparison of S distributions using the Kolomogorov-Smirnov (KS) test revealed that no inference method could precisely infer the S distribution (all $P < 10^{-15}$), and therefore S distributions were unable to unequivocally determine the relative merits of swMutSel and pbMutSel.

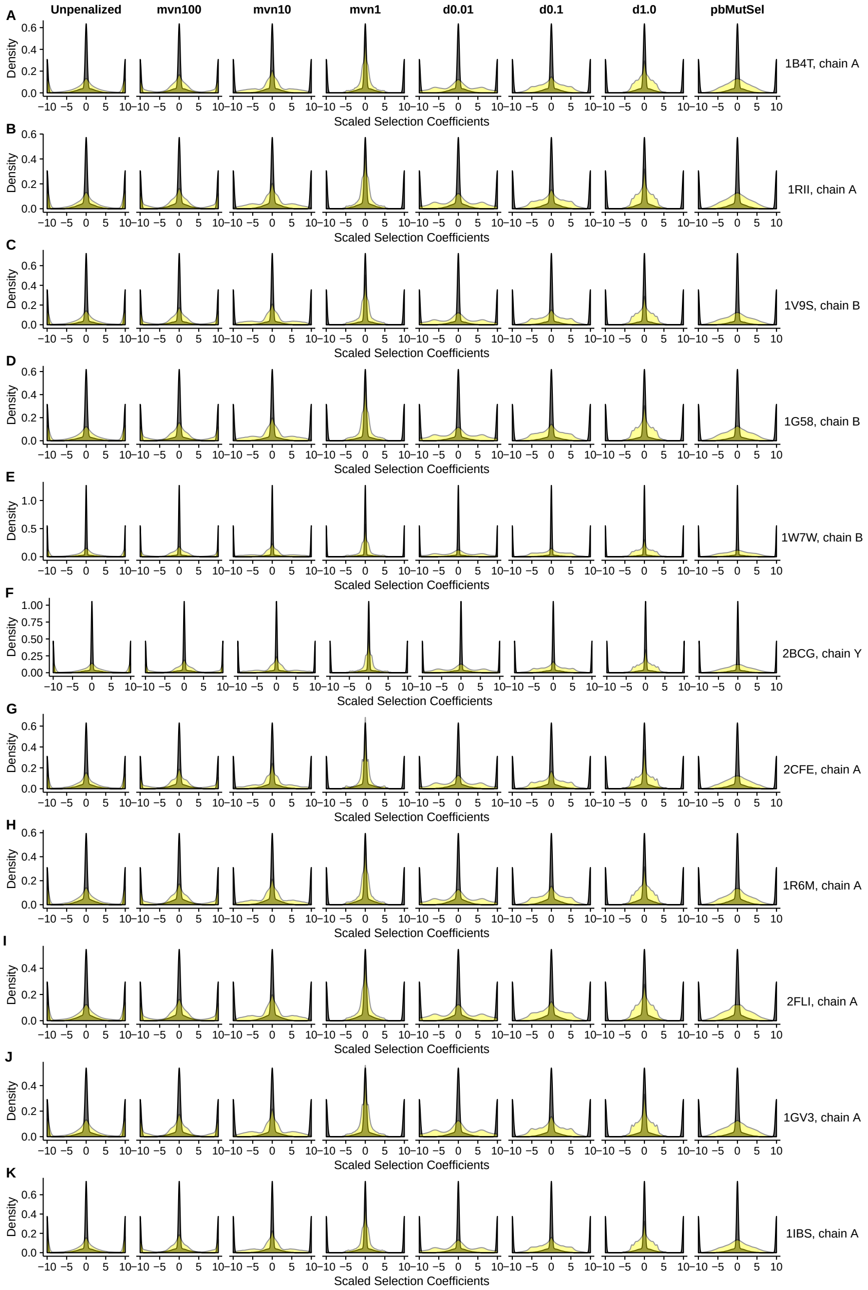


Figure 4.1: True and inferred distributions of scaled selection coefficients across inference methods, for a representative dataset. Scaled selection coefficients have been binned at $S \geq 10$ and $S \leq -10$ for visualization. The text to the right of each row indicates the yeast alignment in ref. [91] from which simulation parameters were derived.

4.2.3 Extensively-parameterized models show smallest distance between true and inferred parameters

We next assessed how the inferred site-specific fitness values compared to the true fitness values. We derived, for each site-specific set of inferred fitnesses, the corresponding equilibrium amino-acid frequencies [106, 109]. We calculated the Jensen-Shannon distance (JSD) between the inferred and true equilibrium frequency distributions. JSD is defined as

$$\mathrm{JSD}(P, Q) = \sqrt{\frac{D(P, M) + D(Q, M)}{2}}, \tag{4.1}$$

where $P = (p_1, \ldots, p_{20})$ and $Q = (q_1, \ldots, q_{20})$ are the amino-acid frequency distributions to be compared, $M = (P + Q)/2$ is the element-wise average between P and Q, and $D(A, B) = \sum_i a_i \ln(a_i/b_i)$ is the Kullback-Leibler divergence between distributions $A = (a_1, \ldots, a_{20})$ and $B = (b_1, \ldots, b_{20})$. JSD values range from 0 for completely identical distributions to 1 for completely dissimilar distributions.

Across all datasets, unpenalized swMutSel, mvn10, and mvn100 yielded the lowest mean JSD value of roughly 0.09 (Figure 4.2). The JSD distributions inferred by these three approaches were statistically indistinguishable ($P > 0.95$, mixed-effects linear model). The stringent mvn1 penalty, as well as all swMutSel Dirichlet penalties, showed increasingly larger distances between the inferred and true amino-acid frequencies. In general, swMutSel's Dirichlet penalty function appeared to influence JSD more strongly than did its multivariate normal penalty function. The JSD distributions from pbMutSel were comparable to an intermediate Dirichlet penalty (here, d0.1), consistent with the mathematical equivalence between its use of a Dirichlet prior with the Dirichlet-penalized maximum likelihood [116].

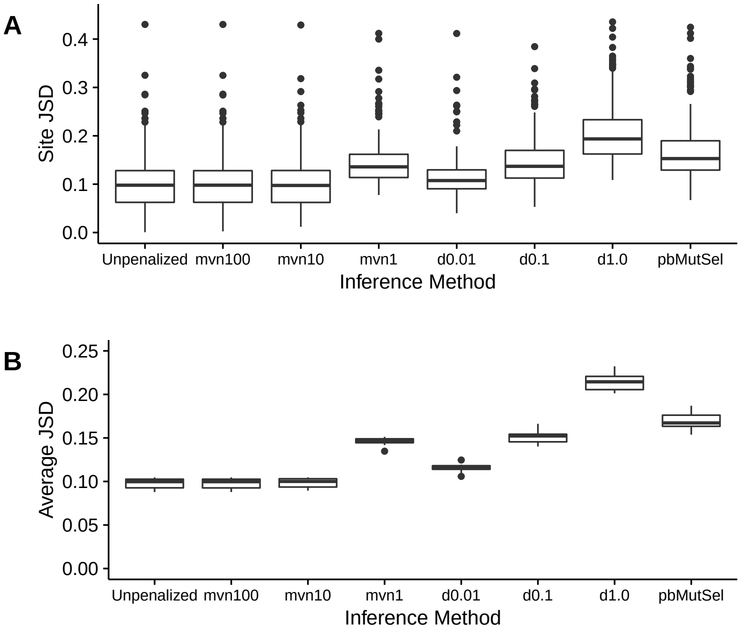


Figure 4.2: Jensen-Shannon distance between true and inferred amino-acid frequency distributions. (A) JSD for individual sites from a representative simulation dataset. The simulation dataset shown was derived from the yeast protein alignment in ref. [91] corresponding to PDB ID 1R6M, chain A. (B) Average JSD for all eleven simulated datasets, where each point represents the mean JSD across sites for a given simulation.

4.2.4 Extensively-parameterized models best infer evolutionary constraint

While JSD provided a useful metric for determining the distance between inferred and true frequency distributions, it is not an explicit evolutionary measure. For instance, while a large JSD indicates high dissimilarity, it is neither possible to tell how this dissimilarity relates to selection pressure, nor whether high JSD corresponds to systematically-biased or randomly-distributed error in estimates.

Therefore, we next asked whether site-specific inferences from swMutSel and pbMutSel corresponded to the true selective constraint at each site. We measured selective constraint by predicting a dN/dS ratio for each site's set of mutation–selection parameters [108, 109]. In this context, dN/dS provides an evolutionarily-aware summary statistic for the selection pressure acting at a given site, incorporating both amino-acid fitness values and nucleotide mutation rates. Moreover, dN/dS has a clear, widely-accepted interpretation: Lower ratios indicate stronger selective constraint, and higher ratios indicate progressively weaker constraint. Importantly, because our simulations assumed symmetric nucleotide mutation rates and no codon bias, all dN/dS ratios are constrained to $dN/dS \in [0, 1]$, as we have previously shown [109].

We derived a site-specific dN/dS ratio for each true and inferred distribution of site-specific amino-acid fitnesses and nucleotide mutation rates [109], and we compared the resulting true and predicted dN/dS ratios across inference methods. Results recovered from this analysis followed similar trends to those seen in the JSD analysis (Figures 4.3 and 4.4). swMutSel, run either as unpenalized or with the mvn100, mvn10, or d0.01 penalties, yielded the strongest Pearson correlations ($r \sim 0.95$) between true and predicted dN/dS (Figure 4.4A). Furthermore, these four approaches tended to slightly underestimate dN/dS across sites, indicating that the inferred selection constraint was stronger than was the true level of constraint (Figure 4.4B). However, the estimator bias observed for d0.01 was not statistically significant for any of the eleven simulated datasets (all $P > 0.01$, test for intercept in linear model).

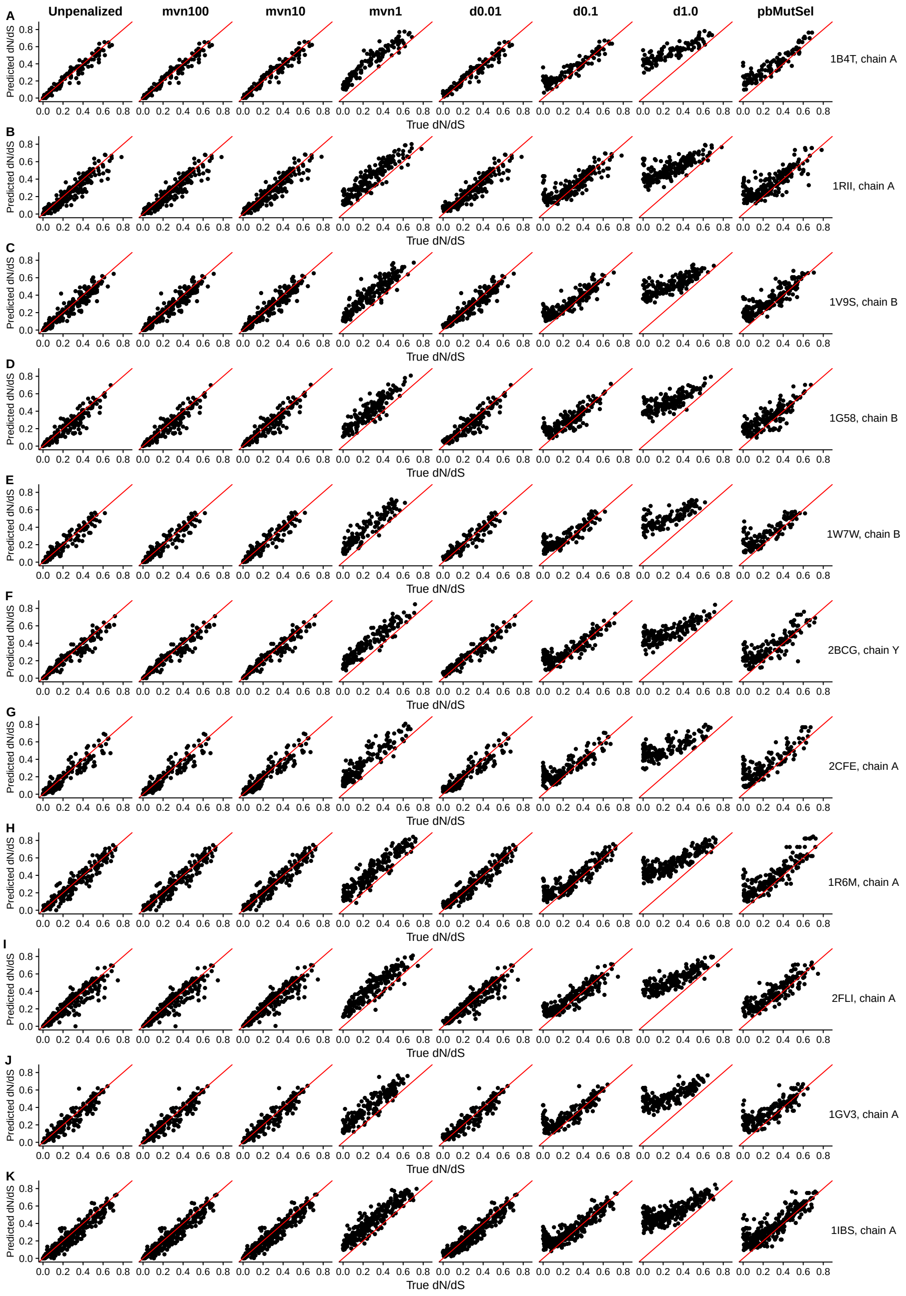


Figure 4.3: Scatterplots of predicted vs. true dN/dS ratios, for all inference methods, across simulated datasets, for simulations with strongly deleterious changes. Each red line indicates the $y = x$ line, and the text to the right of each row indicates the yeast alignment in ref. [91] from which simulation parameters were derived.

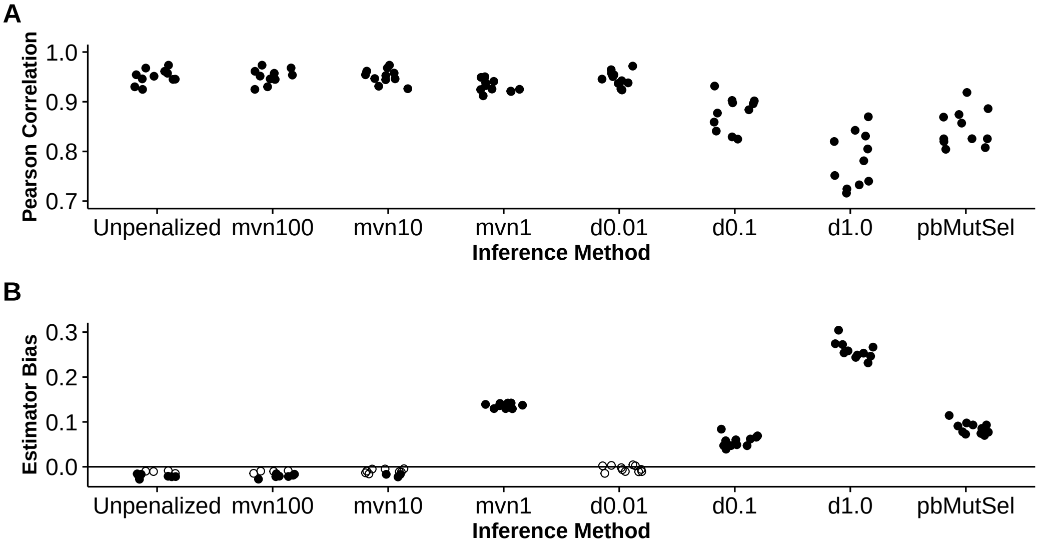


Figure 4.4: Performance of mutation–selection model inference platforms. (A) Pearson correlation coefficients between true and inferred dN/dS across inference methods, for all simulated datasets. (B) Estimator bias of inference methods relative to true dN/dS values, for all simulated datasets. Open points indicate estimator biases that were not significantly different from 0 ($P > 0.01$, test for intercept in linear model), and solid points indicate biases that were significantly different from 0 ($P \leq 0.01$, test for intercept in linear model). The $y = 0$ line shown indicates a bias of 0, which would reflect an unbiased estimator.

The four remaining approaches (mvn1, d0.1, d1.0, and pbMutSel) additionally showed moderate-to-high correlations between true and predicted dN/dS, but they all systematically overestimated dN/dS. In other words, these approaches (particularly d1.0) consistently inferred much weaker selection pressure than was truly present. This trend was pronounced for highly-constrained, i.e. low dN/dS, sites. Therefore, inferences from these approaches did not capture underlying site-specific selection constraint with the same accuracy as did unpenalized or weakly-penalized swMutSel.

4.2.5 Direction of estimation error depends on parameterization

We next asked whether a given site's underlying selective constraint, as represented by dN/dS, influenced error in the inferred fitness values, as represented by site-specific JSD. For the unpenalized and multivariate normal swMutSel penalities, JSD increased with decreasing selective constraint (i.e. increasing dN/dS), indicating that these approaches estimated fitness values most precisely for highly conserved residues (Figure 4.5A). By contrast, the swMutSel Dirichlet parameterizations and pbMutSel displayed the opposite trend: These approaches estimated fitnesses most precisely for sites with weak selective constraint (high dN/dS), and consequently JSD was highest for sites with low dN/dS. Even so, the overall JSD remained lowest for unpenalized, mvn100, and mvn10 (Figure 4.2).

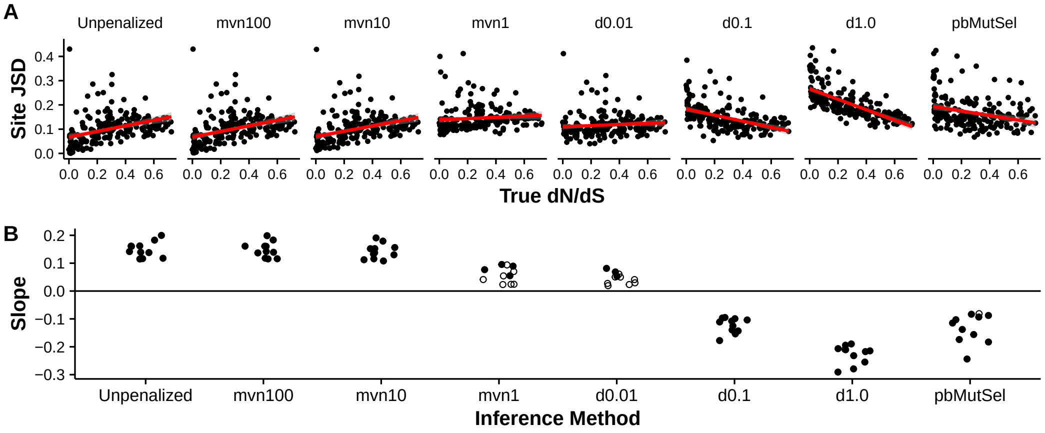


Figure 4.5: The site-specific Jensen-Shannon distance between true and inferred amino-acid frequencies depends both on selective constraint and inference method. (A) Site-specific JSD plotted against true dN/dS for a representative simulation dataset. The line in each panel indicates the linear regression line. The simulation dataset shown was derived from the yeast protein alignment in ref. [91] corresponding to PDB ID 1R6M, chain A. (B) Slope of relationship shown in panel (A) for all eleven simulated datasets. The straight line indicates the $y = 0$ line, meaning no linear relationship between JSD and dN/dS. Open points indicate slopes that were not significantly different from 0 ($P > 0.01$), and solid points indicate slopes that were significantly different from 0 (all significant $P < 10^{-3}$).

Interestingly, across all datasets (Figure 4.5B), mvn1 and d0.01, representing the swMutSel's strongest multivariate normal and weakest Dirichlet penalty, respectively, displayed the weakest relationship between JSD and true dN/dS. Of the eleven datasets analyzed, only four analyzed with mvn1 and three analyzed with d0.01 showed a significant relationship ($P < 0.01$) between dN/dS and JSD. These results, coupled with the strong agreement between true and predicted dN/dS ratios (Figures 4.3 and 4.4), support the use of swMutSel with a weak Dirichlet prior (d0.01) for the most reliable mutation–

selection model inference. This swMutSel parameterization displayed the highest correlation for site-specific constraint without any significant estimator bias, and the error in site-specific fitness estimation was the least influenced by underlying selection pressure.

4.2.6 Results are robust to an absence of strongly deleterious substitutions

Taken together, our results pointed to the swMutSel implementation, either unpenalized or weakly-penalized, as the most reliable mutation–selection model inference platform. However, as seen in Figure 4.1, all true S distributions contained relatively high proportions of strongly deleterious changes ($S < -10$). Across datasets, an average $40.6 \pm 1.3\%$ of the possible substitutions were considered strongly deleterious. Given that swMutSel is known to estimate large proportions of strongly deleterious changes, our simulations may have been biased towards favoring the extensively-parameterized swMutSel platform over pbMutSel.

To address this potentially confounding factor, we simulated a second set of eleven alignments, whose site-specific selective constraints were virtually identical to those described earlier (average Pearson correlation $r = 0.95 \pm 0.01$), except that we re-assigned the scaled fitness values of all strongly deleterious codons to a weakly deleterious fitness drawn from a uniform distribution $F = \mathcal{U}(-6, -4.5)$. Note that the maximally-fit codon, for all sites, had a fitness of $F = 0$, and hence all resulting $|S| \leq 6$. These new selective pressures removed all strongly deleterious changes from the simulations, leaving only weakly deleterious changes. Indeed, the resulting true S distributions for these updated parameters were distinctly unimodal (Figure 4.6). We will refer to these new simulations as "weakly deleterious" and to the original simulations as "strongly deleterious." We processed each new alignment with swMutSel and pbMutSel, and we assessed the correspondence between true and inferred S distributions as well as the true and predicted site-specific dN/dS ratios.

As with the strongly deleterious simulations, we found that no inferred S distribution precisely corresponded to the true S distribution (all $P < 10^{-15}$, KS test). However, we found that which inferred S distribution provided the best qualitative approximation of the true distribution differed from the strongly deleterious simulations. For weakly deleterious simulations, S distributions inferred by strongly-penalized swMutSel (in particular mvn1) and pbMutSel best captured the shape of the true S distribution (Figures 4.7 and 4.6). Further, in spite of the lack of strongly deleterious substitutions, unpenalized swMutSel, and to a lesser degree mvn100, inferred a distinct mode of $S < -10$ coefficients (Figures 4.7 and 4.6), which suggested possible overparameterization.

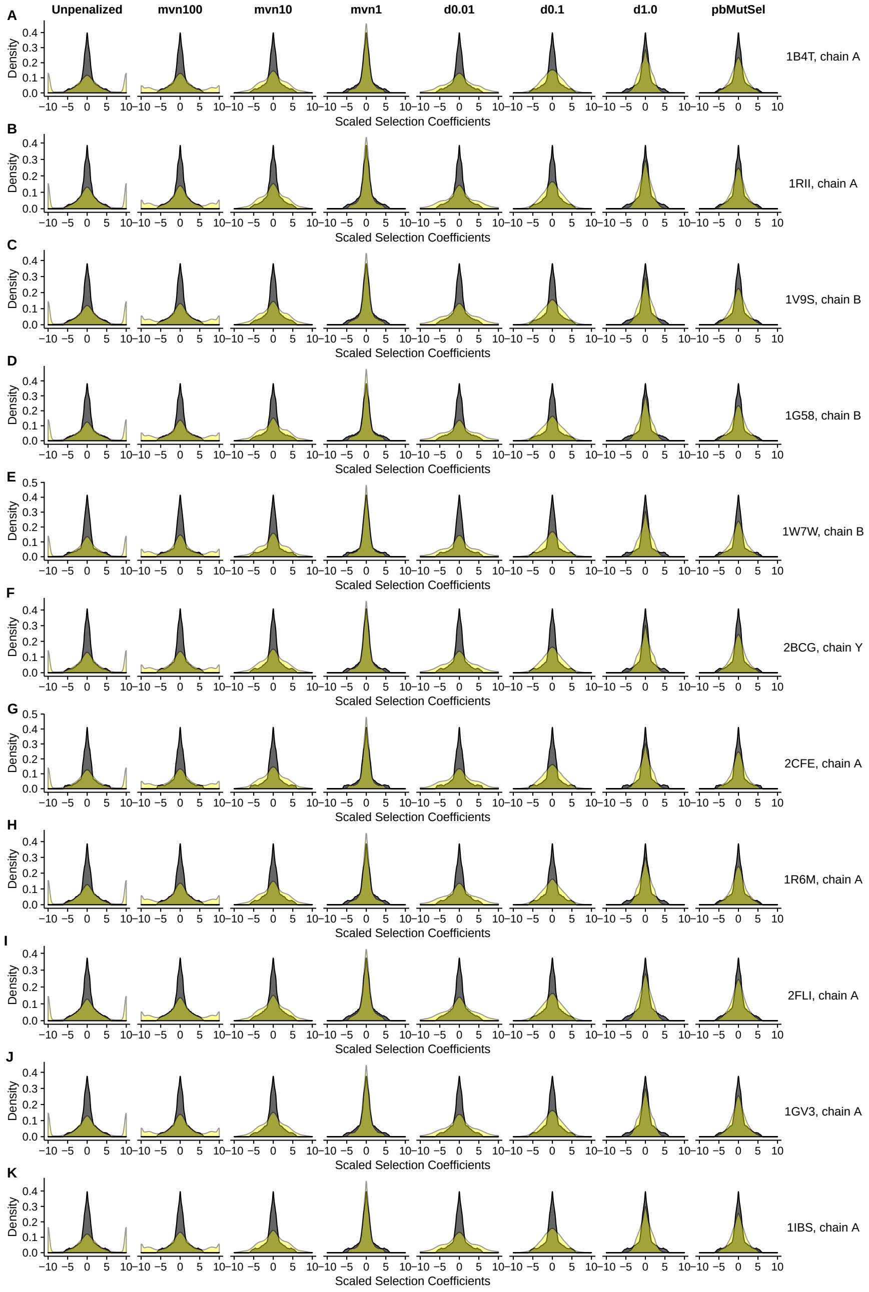


Figure 4.6: Distributions of scaled selection coefficients across all inference methods, for weakly deleterious simulations. For visualization, distributions have been binned at $S \leq -10$ and $S \geq 10$. The text to the right of each row indicates the yeast alignment in ref. [91] from which simulation parameters were derived.

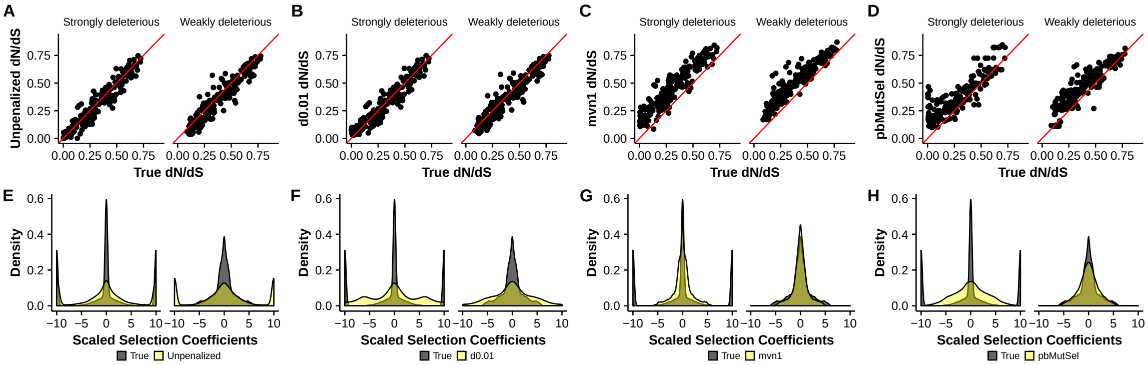


Figure 4.7: Accuracy of predicted dN/dS, but not of selection coefficient distributions, was robust to the proportion of highly deleterious amino acids. (A-D) Predicted dN/dS from vs. true dN/dS for a strongly and weakly deleterious simulated alignment, for inference approaches unpenalized, d0.01, mvn1, and pbMutSel. The line in each panel represents the $y = x$ line. (E-H) True and inferred S distributions for a strongly (left) and weakly (right) deleterious simulated alignment,for inference approaches unpenalized, d0.01, mvn1, and pbMutSel. All results in this figure correspond to the alignment simulated using parameters derived from the yeast protein alignment in ref. [91] corresponding to PDB ID 1R6M, chain A.

Between simulation sets, the true vs. predicted dN/dS correlations were relatively larger under the weakly deleterious simulations when analyzed by strongly-penalized swMutSel and/or pbMutSel (Figure 4.9A). By contrast, when analyzed with unpenalized and weakly-penalized swMutSel, correlations were larger under the strongly deleterious regime compared to the weakly deleterious regime. Even so, under both weakly and strongly deleterious conditions, unpenalized and weakly-penalized swMutSel consistently showed the largest correlations. In addition, estimator bias was generally lower for weakly deleterious compared to strongly deleterious simulations under all swMut-

Sel parameterizations (Figure 4.9B). For pbMutSel, the extent of estimator bias was consistent between simulation sets. Again, however, unpenalized and weakly-penalized swMutSel remained the least biased inference methods. Therefore, the proportion of strongly deleterious changes did not dramatically influence relative performance accuracy.

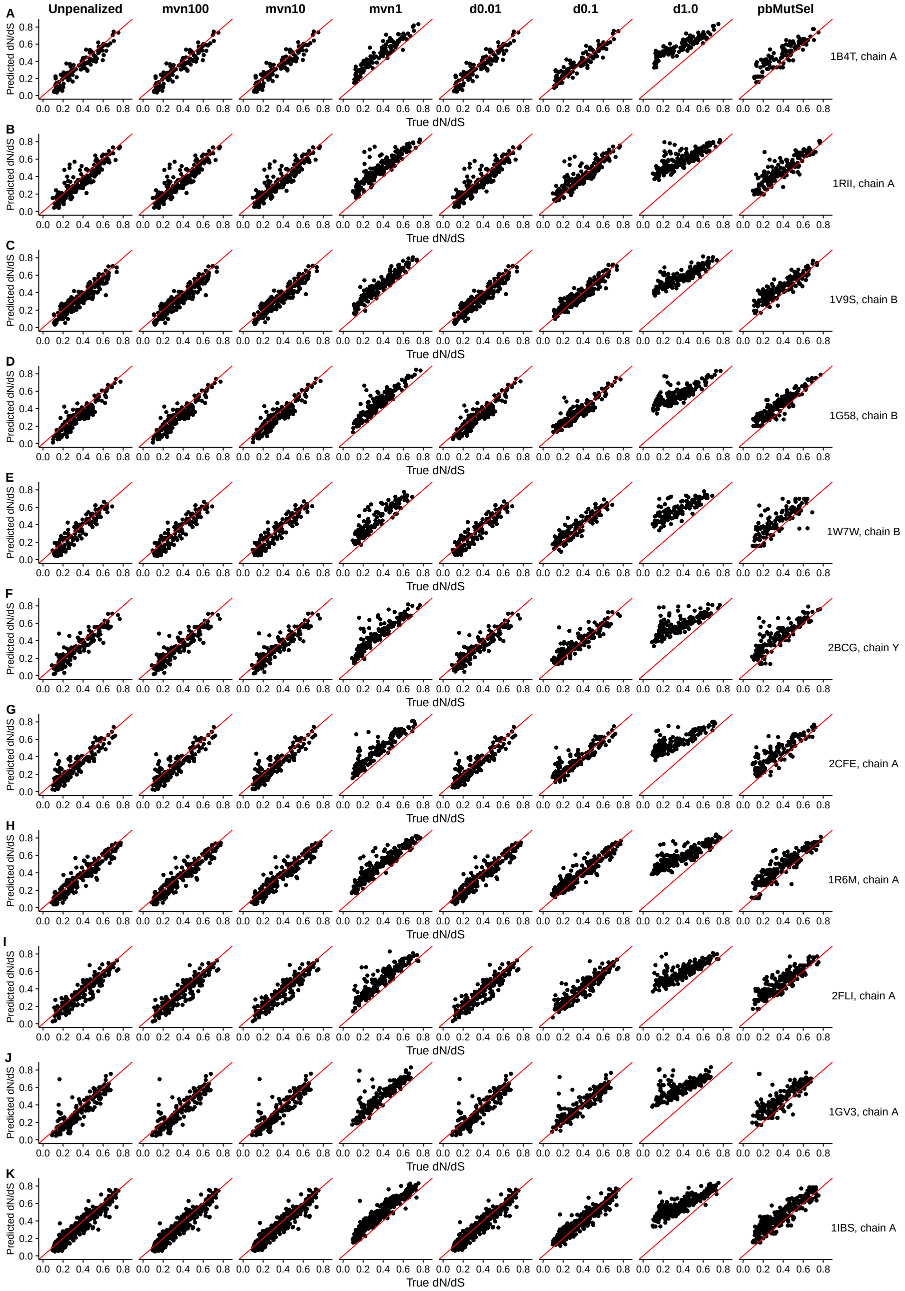


Figure 4.8: Scatterplots of predicted vs. true dN/dS ratios, for all inference methods, across weakly deleterious simulations. Each red line indicates the $y = x$ line, and the text to the right of each row indicates the yeast alignment in ref. [91] from which simulation parameters were derived.

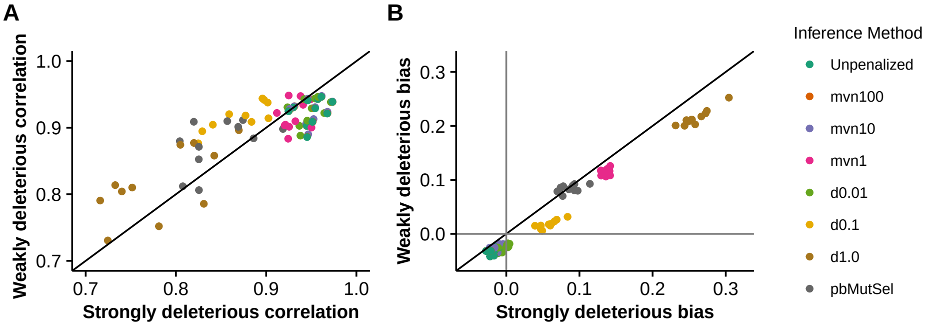


Figure 4.9: Correlation and estimator bias compared between strongly and weakly deleterious simulations. (A) Pearson correlations for simulation pairs across methods. The straight line is the $y = x$ line. (B) Estimator bias for simulation pairs across methods. The diagonal line indicates the $y = x$ line, and the vertical and horizontal lines indicate the $x = 0$ and $y = 0$ lines, respectively. Estimator bias is smallest for points closest to (0, 0). Note that all mvn100 points fall directly beneath all unpenalized points.

Taken together, these results revealed a strong disconnect between S distributions and site-specific evolutionary constraint: The mutation–selection implementation that provided the best S estimates did not necessarily provide the best estimates of site-specific selection pressure (Figure 4.7). The method which yielded the best S estimate depended on the underlying selection pressure (weakly or strongly deleterious), whereas unpenalized and/or weakly-penalized swMutSel (particularly d0.01) consistently yielded the highest dN/dS correlations under both weakly and strongly deleterious simulation sets. In other words, while accuracy of S inference depended on the underlying selective landscape, accuracy in estimating the strength of natural selection at individual sites was robust to this change.

4.3 Discussion

We have investigated the utility of mutation–selection model inference platforms for inferring site-specific selective constraint from coding sequences. We did not recover unequivocal evidence that any inference method could precisely infer the gene-wide distribution of scaled selection coefficients. However, swMutSel, run either unpenalized or with a weak penalty function, consistently inferred site-specific fitness values that reliably captured each site's evolutionary constraint (Figures 4.3, 4.4, and 4.8). pbMutSel and swMutSel run with a strong penalty function systematically underestimated the strength of natural selection across sites. Importantly, these results were robust to the proportion of deleterious changes in the data: Even when all $|S| \leq 6$, indicating only weakly deleterious substitutions, pbMutSel and strongly-penalized swMutSel still substantially underestimated selective constraint across sites.

We identified a striking discordance between gene-wide S distributions and site-specific evolutionary constraint, as measured by the dN/dS ratio. While unpenalized and weakly-penalized swMutSel consistently inferred site-specific amino-acid fitness values that most precisely corresponded to selection pressure, the mutation–selection implementation that inferred the most qualitatively correct S distribution depended heavily on the gene-wide selective constraint (i.e. presence or absence of strongly deleterious substitutions). Strongly-penalized swMutSel and pbMutSel best estimated the S distribution for weakly deleterious simulations, but unpenalized or weakly-penalized swMutSel best estimated S for strongly deleterious simulations. As such, it would be difficult to discern from S inferences alone which inference platform produced amino-acid fitness estimates that best reflected evolutionary constraint, ultimately revealing that S distributions may be a poor and misleading quantity for evaluating methodological performance. Taking the purpose of these models into consideration, this observation makes perfect sense. Both swMutSel and pbMutSel were implemented for the specific purpose of mechanistically modeling site-specific selection pressure in protein-coding sequences. Focusing on whole-gene metrics over site-specific inferences stands at odds with the very motivation behind the site-wise mutation–selection model.

From a purely statistical standpoint, it may seem unsettling that unpenalized swMutSel inferred a peak, albeit a relatively small one, of strongly deleterious substitutions for the weakly deleterious simulations (Figures 4.7 and 4.6). These estimates, as previously noted by ref. [96], were likely made because swMutSel's "extensive parameterization approach considers unobserved amino acids as highly deleterious." ref. [96] further suggested that pbMutSel's "less conclusive" inferences regarding unseen amino acids represent a more desirable behavior, reasoning that just because an amino acid has not been observed does not necessarily mean that it was highly deleterious or lethal.

While this argument may seem appealing, we contend that swMutSel's treatment of unseen amino acids is preferred. Inferring anything other than a highly deleterious fitness value for unseen amino acids directly contradicts the logic of the underlying reversible Markov model. The mutation–selection model implemented in both swMutSel and pbMutSel assumes that sequences evolve under an equilibrium process. As such, the observed data is directly interpreted as representative sample of the model's steady-state distribution [46, 109, 130]. Under the assumption of equilibrium, the only logical way to model unseen amino acids is to assume that they have an exceptionally small steady-state frequency. In the mutation–selection model framework, such small frequencies are a direct result of extremely low fitnesses which, by definition, mostly preclude their fixation. Whether scaled selection coefficients associated with such highly deleterious amino acids are $S = -20$ or $S = -50$ is largely irrelevant: As long as the fixation probability for that amino acid is sufficiently low, then selective constraint will be well-estimated.

We additionally emphasize that, while unpenalized and/or weakly-penalized swMutSel emerged here as the more reliable mutation–selection inference platform, dN/dS ratios predicted from all inferences correlated strongly with the true dN/dS ratios (Figure 4.4), and indeed with one another. For example, the Pearson correlation between dN/dS predicted from unpenalized swMutSel and pbMutSel was, on average, $r = 0.89$ across all simulations. This high correlation contrasts strongly with conclusions drawn from previous studies that swMutSel and pbMutSel make fundamentally distinct inferences. Importantly, such assertions have been made entirely by comparing true and inferred S distributions, and not based on any rigorous quantitative comparison of true vs. inferred site-specific parameters. Our results, therefore, demonstrated that performance differences between swMutSel and pbMutSel, while clearly present, were smaller than one would assume based on S distributions alone.

We suggest that some modifications to pbMutSel's default settings, such as changing the fixed dispersion parameter for its Dirichlet prior, may produce more reliable inferences. Although such efforts may be helpful, there remained salient differences in runtime between swMutSel and pbMutSel. For example, each swMutSel inference required between six and 24 hours to converge (with unpenalized swMutSel inferences taking the most time), whereas each pbMutSel inference required between one to two weeks. In other words, each swMutSel inference converged seven to 50 times more quickly than did each pbMutSel inference. From a practical standpoint, swMutSel's relatively short runtime and reliable inferences make it the preferred inference platform.

In sum, although mutation–selection models may not be well-suited for inferring the precise distribution of S from any dataset, they can readily capture selection pressures acting at individual sites. We recommend the use of swMutSel with a weak Dirichlet prior (e.g. with $\alpha = 0.01$, as investigated here), as this parameterization provided the most accurate and least biased estimates of site-specific evolutionary constraint.

4.4 Materials and Methods

4.4.1 Generation of simulated data

Sequences were simulated according to the mutation–selection model in ref. [46], which assumes a reversible Markov model of sequence evolution.

For each site k, this model's rate matrix is given by

$$q_{ij}^{(k)} = \begin{cases} \mu_{ij} f_{ij}^{(k)} & \text{single nucleotide change} \\ 0 & \text{multiple nucleotide changes} \end{cases}, \tag{4.2}$$

where μ_{ij} is the site-invariant mutation rate between codons i and j, and $f_{ij}^{(k)}$, the site-specific fixation probability from codon i to j, is defined as

$$f_{ij}^{(k)} = \frac{S_{ij}^{(k)}}{1 - e^{-S_{ji}^{(k)}}}, \tag{4.3}$$

where $S_{ij}^{(k)}$ is the scaled selection coefficient from codon i to j at site k [46]. Note that $f_{ij}^{(k)}$ can also be expressed as

$$f_{ij}^{(k)} = \ln\left(\frac{\pi_j^{(k)}\mu_{ij}}{\pi_i^{(k)}\mu_{ji}}\right) \Big/ \left(1 - \frac{\pi_i^{(k)}\mu_{ji}}{\pi_j^{(k)}\mu_{ij}}\right), \tag{4.4}$$

where π_i^{k} is the equilibrium frequency of codon i at site k [46, 109].

We determined each alignment's site-specific codon frequencies directly from an empirical dataset of yeast amino-acid alignments, each homologous to a given PDB struture, compiled by ref. [91]. For each yeast alignment which contained at least 150 taxa, we calculated each site's amino acid frequencies, which we converted to codon frequencies under the assumption that all synonymous codons for a given amino acid had the same frequency. In addition, sites which contained fewer than 150 amino acids (e.g. a column in an alignment with 200 taxa but half of whose characters are gaps) were discarded. A total of eleven yeast alignments, with a number of codon positions ranging from 115–291, were obtained from this procedure. We additionally set the equilibrium frequency of all unobserved amino acids to 10^{-9}. For alignments simulated with only weakly deleterious changes, we re-assigned the equilibrium frequency of the most deleterious amino acids by drawing a fitness value from a uniform distribution $F = \mathcal{U}(-4.5, -6)$, where each set of site-specific fitnesses were scaled to give the maximally-fit codon a fitness of $F = 0$.

We then simulated an alignment corresponding to each of these eleven proteins using the Python library Pyvolve [108]. All simulations were conducted along a 512-taxon balanced tree with branch lengths equal to 0.5. Finally, we inferred a true dN/dS for each alignment's column as described in ref. [109].

4.4.1.1 Mutation–selection model inference

We processed all alignments, both simulated and empirical, with swMutSel and pbMutSel. swMutSel inference was carried out under seven specifications, including without the use of a penalty function, and three parameterizations each for both the multivariate normal and the Dirichlet penalty functions. For the multivariate normal penalty, we set μ to either 1, 10 or 100, and for the Dirichlet penalty, we set α to either 1.0, 0.1, or 0.01.

For inference with pbMutSel, we followed the inference approach given in ref. [96]. We ran each chain for 5500 iterations, saving every 5 cycles until a total sample size of 1100 was obtained. The first 100 samples were discarded as burnin, and hence the final posterior distribution from which fitnesses were calculated contained 1000 draws. Convergence was assessed visually using Tracer [90].

For each mutation–selection inference, we calculated a site-specific dN/dS value directly from inferred parameters [109].

4.4.2 Statistical Analysis and Data Availability

All statistical analyses were conducted in the R programming language [88]. All reported P-values were corrected for multiple testing using the Bonferroni correction. Simulated data, statistical analyses, and all code used are freely available from the github repository `https://github.com/sjspielman/mutsel_benchmark`.

Chapter 5

Conclusion

The work described in this dissertation will provide a concrete basis on which new statistical models of sequence evolution can be evaluated, compared, and developed. The Pyvolve library described in Chapter 2 has already provided a flexible and easily-extensible tool for sequence simulation, which can in turn be used to develop and test models [37, 109]. In addition, the formal, mathematical relationship derived in Chapter 3 between dN/dS and mutation–selection model parameters allowed for a more rigorous assessment of both dN/dS-based and mutation–selection models. Past attempts to discern the relative merits among dN/dS-based model formulations have relied either on model fit metrics (e.g. AIC or BIC) or simply through intuitive assumptions, which are not strictly scientific. Our results demonstrated that the use of model fit metrics can, in fact, be highly misleading (Tables 3.3 and 3.4) and allow for spurious conclusions. Importantly, we were only able to recover this critical finding by testing dN/dS-based models with mutation–selection model simulation, revealing the broad utility of understanding the relationships among sequence evolution modeling frameworks.

Because the mutation–selection model assumes that selection pressure is constant, on a per-site basis, over time, it effectively provides a null model for asking whether a given site is evolving under an equilibrium evolutionary process. Using a combined modeling approach, one can envision a novel hypothesis-testing framework wherein we can infer both dN/dS and mutation–selection model parameters from a dataset, both using standard maximum-likelihood approaches. Next, we can predict a null expectation for dN/dS based on the inferred mutation–selection parameterization at each site. We can then formally test the hypothesis that a given site is evolving under the mutation–selection model by comparing the ML-inferred dN/dS to the dN/dS ratio predicted from mutation–selection parameters. This framework would necessitate estimating a confidence interval for the null dN/dS expectation, which can be accomplished by bootstrapping.

An analogous approach to the one described above has recently been proposed, through which experimentally-derived and computationally-predicted estimates of site-specific fitness are compared to reveal sites under selection [15]. This framework has already shown promising results. It has successfully been applied to four proteins and has identified sites known to be under selection that would not normally be identified as such using a dN/dS-based model. However, the proposed procedure requires extensive, and costly, experimental assays. The computational approach described above would therefore provide a more accessible and reproducible test for non-equilibrium evolution. Such a hypothesis test would have a wide-ranging influence across evolutionary biology, and potentially establish a new paradigm, in addition to the standard

dN/dS-based tests for positive selection, for identifying cases of shifting selective constraint over time.

Bibliography

[1] L. A. Abriata, T. Palzkill, and M. Dal Peraro. How structural and physicochemical determinants shape sequence constraints in a functional enzyme. *PLOS ONE*, 10:e0118684–15, 2015.

[2] A. Acevedo, L. Brodsky, and R. Andino. Mutational and fitness landscapes of an RNA virus revealed through population sequencing. *Nature*, 505:686–690, 2014.

[3] J. Adachi and M. Hasegawa. MOLPHY version 2.3: programs for molecular phylogenetics based on maximum likelihood. *Comput. Sci. Monogr.*, 28:1–150, 1996.

[4] D. Agashe, N. C. Martinez-Gomez, D. A. Drummond, and C. J. Marx. Good codons, bad transcript: Large reductions in gene expression and fitness arising from synonymous mutations in a key enzyme. *Mol. Biol. Evol.*, 30:549–560, 2013.

[5] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:6:716–723, 1974.

[6] M. Anisimova and C. Kosiol. Investigating protein-coding sequence evolution with probabilistic codon substitution models. *Mol. Biol. Evol.*, 26:255–271, 2009.

[7] M. Arenas. Simulation of molecular data under diverse evolutionary scenarios. *PLoS Comp. Biol.*, 8:e1002495, 2012.

[8] M. Arenas. Advances in computer simulation of genome evolution: Toward more realistic evolutionary genomics analysis by approximate bayesian computation. *J. Mol. Evol.*, 8:189–192, 2015.

[9] M. Arenas, H. G. Dos Santos, D. Posada, and U. Bastolla. Protein evolution along phylogenetic histories under structurally constrained substitution models. *Bioinformatics*, 29:3020–3028, 2013.

[10] M. Arenas and D. Posada. Simulation of genome-wide evolution under heterogeneous substitution models and complex multispecies coalescent histories. *Mol. Biol. Evol.*, 31:1295–1301, 2014.

[11] O. Ashenberg, L. I. Gong, and J. D. Bloom. Mutational effects on stability are largely conserved during protein evolution. *Proc. Natl. Acad. Sci. U.S.A.*, 110:21071–21076, 2013.

[12] R. G. Beiko and R. L. Charlebois. A simulation test bed for hypotheses of genome evolution. *Bioinformatics*, 23:825–831, 2007.

[13] S. Bhatt, E. C. Holmes, and O. G. Pybus. The genomic rate of molecular adaptation of the human influenza A virus. *Mol. Biol. Evol.*, 28:2443–2451, 2011.

[14] F. Bielejec, P. Lemey, L. M. Carvalho, G. Baele, A. Rambaut, and M. A. Suchard. piBUSS: A parallel BEAST/BEAGLE utility for sequence sim-

ulation under complex evolutionary scenarios. *BMC Bioinform*, 15:133, 2014.

[15] J. Bloom. Identification of positive selection in genes is greatly improved by using experimentally informed site-specific models. *bioRxiv*, 2016.

[16] J. D. Bloom. An experimentally determined evolutionary model dramatically improves phylogenetic fit. *Mol. Biol. Evol.*, 31:1956–1978, 2014.

[17] J. D. Bloom. An experimentally informed evolutionary model improves phylogenetic fit to divergent lactamase homologs. *Mol. Biol. Evol.*, 31:1956–1978, 2014.

[18] J. D. Bloom, L. I. Gong, and D. Baltimore. Permissive secondary mutations enable the evolution of influenza oseltamivir resistance. *Science*, 328:1272–1275, 2010.

[19] A. C. Brault, C. Y.-H. Huang, S. A. Langevin, R. M. Kinney, R. A. Bowen, W. N. Ramey, N. A. Panella, E. C. Holmes, A. M. Powers, and B. R. Miller. A single positively selected West Nile viral mutation confers increased virogenesis in American crows. *Nature Genetics*, 39(9):1162–6, 2007.

[20] K. P. Burnham and D. R. Anderson. Multimodel inference: understanding AIC and BIC in model selection. *Sociol. Method Res.*, 33:261–304, 2004.

[21] R. M. Bush, C. A. Bender, K. Subbarao, N. J. Cox, and W. M. Fitch. Predicting the evolution of human influenza A. *Science*, 286:1921–1925, 1999.

[22] R. A. Cartwright. DNA assembly with gaps Dawg): simulating sequence evolution. *Bioinformatics*, 21:iii31–iii38, 2005.

[23] J. V. Chamary and L. D. Hurst. Evidence for selection on synonymous mutations affecting stability of mRNA secondary structure in mammals. *Genome Biol.*, 6:R75, 2005.

[24] J. V. Chamary, J. L. Parmley, and L. D. Hurst. Hearing silence: non-neutral evolution at synonymous sites in mammals. *Nature Rev. Genet.*, 7:98–108, 2006.

[25] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25:1422–1423, 2009.

[26] J. F. Crow and M. Kimura. *An Introduction to Population Genetics Theory*. Burgess Pub. Co., California, 1970.

[27] J. M. Cuevas, P. Domingo-Calap, and R. Sanjuan. The fitness effects of synonymous mutations in DNA and RNA viruses. *Mol. Biol. Evol.*, 29:17–20, 2011.

[28] D. A. Dalquen, M. Anisimova, G. H. Gonnet, and C. Dessimoz. ALF–A simulation framework for genome evolution. *Mol. Biol. Evol.*, 29:1115–1123, 2012.

[29] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352, 1978.

[30] W. Delport, A. F. Y. Poon, S. D. W. Frost, and S. L. Kosakovsky Pond. Datamonkey 2010: a suite of phylogenetic analysis tools for evolutionary biology. *Bioinformatics*, 26:2455–2457, 2010.

[31] W. Delport, K. Scheffler, and C. Seoighe. Frequent toggling between alternative amino acids is driven by selection in HIV-1. *PLoS Pathog.*, 4(12):e1000242, 2008.

[32] S. Dimitrieva and M. Anisimova. Unraveling patterns of site-to-site synonymous rates variation and associated gene properties of protein domains and families. *PLOS ONE*, 9:e95034, 2014.

[33] M. dos Reis. How to calculate the non-synonymous to synonymous rate ratio of protein-coding genes under the fisher-wright mutation-selection framework. *Biol. Lett.*, 11:20141031, 2015.

[34] M. B. Doud, O. Ashenberg, and J. D. Bloom. Site-specific amino acid preferences are mostly conserved in two closely related protein homologs. *Mol. Biol. Evol.*, 32:2944–2960, 2015.

[35] A. J. Drummond and A. Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol. Biol.*, 7:214, 2007.

[36] D. A. Drummond and C. O. Wilke. Mistranslation-Induced Protein Misfolding as a Dominant Constraint on Coding-Sequence Evolution. *Cell*, 134:341–352, 2008.

[37] S. Duchene, F. Di Giallonardo, and E. C. Holmes. Substitution model adequacy and assessing the reliability of estimates of virus evolutionary rates and time scales. *Mol. Biol. Evol.*, 33:255–267, 2016.

[38] L. Duret. Evolution of synonymous codon usage in metazoans. *Curr. Opin. Genet. Dev.*, 12:640–649, 2002.

[39] J. Echave, S. J. Spielman, and C. O. Wilke. Causes of evolutionary rate variation among protein sites. *Nat. Rev. Genet.*, 17:109–121, 2016.

[40] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

[41] W. Fletcher and Z. Yang. INDELible: A flexible simulator of biological sequence evolution. *Mol. Biol. Evol.*, 26:1879–1888, 2009.

[42] N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol. Biol. Evol.*, 11:725–736, 1994.

[43] J. A. Grahnen and D. A. Liberles. CASS: Protein sequence simulation with explicit genotype-phenotype mapping. *Trends in Evolutionary Biology*, 4:e9, 2012.

[44] W. Gu, X. Wang, C. Zhai, X. Xie, and T. Zhou. Selection on synonymous sites for increased accessibility around miRNA binding sites in plants. *Mol. Biol. Evol.*, 29:3037–3044, 2012.

[45] W. Gu, T. Zhou, and C. O. Wilke. A universal trend of reduced mRNA stability near the translation-initiation site in prokaryotes and eukaryotes. *PLoS Comput. Biol.*, 6:e1000664, 2010.

[46] A. L. Halpern and W. J. Bruno. Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies. *Mol. Biol. Evol.*, 15:910–917, 1998.

[47] M. Hasegawa, H. Kishino, and T Yano. Dating the humanape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, 22:160–174, 1985.

[48] R. Hershberg and D.A. Petrov. Selection on codon bias. *Annu. Rev. Genet.*, 42, 2008.

[49] M. T. Holder, D. J. Zwickl, and C. Dessimoz. Evaluating the robustness of phylogenetic methods to among-site variability in substitution processes. *Phil. Trans. R. Soc. B*, 363:4013–4021, 2008.

[50] J. P. Huelsenbeck, S. Jain, S. W. D. Frost, and S. L. Kosakovsky Pond. A Dirichlet process model for detecting positive selection in protein-coding DNA sequences. *Proc. Natl. Acad. Sci. U.S.A.*, 103:6263–6268, 2006.

[51] D. T. Jones, W. R. Taylor, and J. M. Thornton. The rapid generation of mutation data matrices from protein sequences. *CABIOS*, 8:275–282, 1992.

[52] T. H. Jukes and C. R. Cantor. Evolution of protein molecules. In H. N. Munro, editor, *Mammalian protein metabolism*. Academic Press, New York, 1969.

[53] M. Kimura. On the probability of fixation of mutant genes in a population. *Genetics*, 4:713–719, 1962.

[54] T. Koestler, A. von Haeseler, and I. Ebersberger. REvolver: modeling sequence evolution under domain constraints. *Mol. Biol. Evol.*, 29:2133–2145, 2012.

[55] S. L. Kosakovsky Pond, W. Delport, S. V. Muse, and K. Scheffler. Correcting the bias of empirical frequency parameter estimators in codon models. *PLOS ONE*, 5:e11230, 2010.

[56] S. L. Kosakovsky Pond and S. D. W. Frost. A genetic algorithm approach to detecting lineage-specific variation in selection pressure. *Mol. Biol. Evol.*, 22:478–485, 2005.

[57] S. L. Kosakovsky Pond and S. D. W. Frost. Not so different after all: A comparison of methods for detecting amino acid sites under selection. *Mol. Biol. Evol.*, 22:1208–1222, 2005.

[58] S. L. Kosakovsky Pond, S. D. W. Frost, and S. V. Muse. HyPhy: hypothesis testing using phylogenetics. *Bioinformatics*, 21:676–679, 2005.

[59] S. L. Kosakovsky Pond, B. Murrell, M. Fourment, S. D. W. Frost, W. Delport, and K. Scheffler. A random effects branch-site model for detecting episodic diversifying selection. *Mol. Biol. Evol.*, 28:3033–3043, 2011.

[60] S. L. Kosakovsky Pond and S. V. Muse. Site-to-site variation of synonymous substitution rates. *Mol. Biol. Evol.*, 22:2375–2385, 2005.

[61] S. L. Kosakovsky Pond, A. F. Y. Poon, A. J. L. Brown, S. D. W. Frost, and S. V. Muse. A maximum likelihood method for detecting directional evolution in protein sequences and its application to influenza A virus. *Mol. Biol. Evol.*, 25:1809–1824, 2008.

[62] C. Kosiol, I. Holmes, and N. Goldman. An empirical codon model for protein sequence evolution. *Mol. Biol. Evol.*, 24:1464–1479, 2007.

[63] S. Kryazhimskiy and J. B. Plotkin. The population genetics of *dN/dS*. *PLOS Genet.*, 4:e1000304, 2008.

[64] N. Lartillot, N. Rodrigue, D. Stubbs, and J. Richer. PhyloBayes MPI: Phylogenetic reconstruction with infinite mixtures of profiles in a parallel environment. *Syst. Biol.*, 62:611–615, 2013.

[65] D. S. Lawrie, P. W. Messer, R. Hershberg, and D. A. Petrov. Strong purifying selection at synonymous sites in *D. melanogaster*. *PLoS Genet.*, 9:e1003527, 2013.

[66] S. Q. Le and O. Gascuel. An improved general amino acid replacement matrix. *Mol. Biol. Evol.*, 25:1307–1320, 2008.

[67] P. O. Lewis. A likelihood approach to estimating phylogeny from discrete morphological character data. *Syst. Biol.*, 50:913–925, 2001.

[68] D. A. Liberles, A.I. Teufel, L. Liu, and T. Stadler. On the need for mechanistic models in computational genomics and metagenomics. *Genome Biol Evol*, 5:2008–2018, 2013.

[69] M. Luksza and M. Lässig. A predictive fitness model for influenza. *Nature*, 507:57–61, 2014.

[70] I. Mayrose, A. Doron-Faigenboim, E. Bacharach, and T. Pupko. Towards realistic codon models: among site variability and dependency of synonymous and non-synonymous rates. *Bioinformatics*, 23:i319–i327, 2007.

[71] A. G. Meyer, E. T. Dawson, and C. O. Wilke. Cross-species comparison of site-specific evolutionary-rate variation in influenza haemagglutinin. *Phil. Trans. R. Soc. B.*, 368:20120334, 2013.

[72] A. G. Meyer and C. O. Wilke. Integrating sequence variation and protein structure to identify sites under selection. *Mol. Biol. Evol.*, 30:36–44, 2013.

[73] V. Minin, Z. Abdo, P. Joyce, and J. Sullivan. Performance-based selection of likelihood models for phylogeny estimation. *Syst. Biol.*, 52:674–683, 2003.

[74] A. Mirsky, L. Kazandjian, and M. Anisimova. Antibody-specific model of amino acid substitution for immunological inferences from alignments of antibody sequences. *Mol. Biol. Evol.*, 32:806–819, 2015.

[75] C. F. Mugal, J. B. W. Wolf, and I. Kaj. Why time matters: Codon evolution and the temporal dynamics of *dN/dS*. *Mol. Biol. Evol.*, 31:212–231, 2014.

[76] B. Murrell, T. de Oliveira, C. Seebregts, S. L. Kosakovsky Pond, K. Scheffler, and on behalf of the Southern African Treatment and Resistance Network (SATuRN) Consortium. Modeling HIV-1 drug resistance as episodic directional selection. *PLoS Comput. Biol.*, 8:e1002507, 2012.

[77] B. Murrell, S. Moola, A. Mabona, T. Weighill, D. Scheward, S. L. Kosakovsky Pond, and K. Scheffler. FUBAR: A Fast, Unconstrained

Bayesian AppRoximation for inferring selection. *Mol. Biol. Evol.*, 30:1196–1205, 2013.

[78] B. Murrell, J. O. Wertheim, S. Moola, T. Weighill, K. Scheffler, and S. L. Kosakovsky Pond. Detecting individual sites subject to episodic diversifying selection. *PLOS Genet*, 8:e1002764, 2012.

[79] S. V. Muse and B. S. Gaut. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Mol. Biol. Evol.*, 11:715–724, 1994.

[80] V. Mustonen and M. Lässig. From fitness landscapes to seascapes: non-equilibrium dynamics of selection and adaptation. *Trends Genet.*, 25:111–119, 2009.

[81] R. Nielsen and Z. Yang. Likelihood models for detecting positive selected amino acid sites and applications to the HIV-1 envelope gene. *Genetics*, 148:929–936, 1998.

[82] T. E. Oliphant. Python for scientific computing. *IEEE Comput. Sci. Eng.*, 9:10–20, 2007.

[83] J. L. Parmley, J. V. Chamary, and L. D. Hurst. Evidence for purifying selection against synonymous mutations in mammalian exonic splicing enhancers. *Mol. Biol. Evol.*, 23:301–309, 2006.

[84] J. L. Parmley and L. D. Hurst. Exonic splicing regulatory elements skew synonymous codon usage near intron-exon boundaries in mammals. *Mol. Biol. Evol.*, 24:1600–1603, 2007.

[85] J. B. Plotkin and G. Kudla. Synonymous but not the same: the causes and consequences of codon bias. *Nature Rev. Genet.*, 12:32–42, 2011.

[86] D. D. Pollack, G. Thiltgen, and R. A. Goldstein. Amino acid coevolution induces an evolutionary Stokes shift. *Proc. Natl. Acad. Sci. U.S.A.*, 109:E1352–E1359, 2012.

[87] M. Porto, H. E. Roman, M. Vendruscolo, and U. Bastolla. Prediction of site-specific amino acid distributions and limits of divergent evolutionary changes in protein sequences. *Mol. Biol. Evol.*, 22:630–638, 2004.

[88] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[89] A. Rambaut and N. C. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput Appl Biosci*, 13:235–238, 1997.

[90] A. Rambaut, M. A. Suchard, D. Xie, and A. J. Drummond. *Tracer v1.6*, 2014.

[91] D. C. Ramsey, M. P. Scherrer, T. Zhou, and C. O. Wilke. The relationship between relative solvent accessibility and evolutionary rate in protein evolution. *Genetics*, 188:479–488, 2011.

[92] A Ratnakumar, S Mousset, S Glémin, J Berglund, N Galtier, L Duret, and MT Webster. Detecting positive selection within genomes: the problem of biased gene conversion. *Phil Trans R Soc B*, 365:2571–2580, 2010.

[93] V. A. Risso, F. Manssour-Triedo, A. Delgado-Delgado, R. Arco, A. Barroso-delJesus, A. Ingles-Prieto, R. Godoy-Ruiz, J. A. Gavira, E. A. Gaucher, B. Ibarra-Molero, and J. M. Sanchez-Ruiz. Mutational studies on resurrected ancestral proteins reveal conservation of site-specific amino acid preferences throughout evolutionary history. *Mol. Biol. Evol.*, 32:440–455, 2014.

[94] D. M. Robinson, D. T. Jones, H. Kishino, N. Goldman, and J. L. Thorne. Protein evolution with dependence among codons due to tertiary structure. *Mol. Biol. Evol.*, 20:1692–1704, 2003.

[95] E. P. C. Rocha, J. Maynard Smith, L. D. Hurst, M. T. G. Holden, J. E. Cooper, N. H. Smith, and E. J. Feil. Comparisons of *dN/dS* are time dependent for closely related bacterial genomes. *J. Theor. Biol.*, 239:226–235, 2006.

[96] N. Rodrigue. On the statistical interpretation of site-specific variables in phylogeny-based substitution models. *Genetics*, 193:557–564, 2013.

[97] N. Rodrigue, C. L. Kleinman, H. Phillipe, and N. Lartillot. Computational methods for evaluating phylogenetic models of codong sequence

evolution with dependence between codons. *Mol. Biol. Evol.*, 26:1663–1676, 2000.

[98] N. Rodrigue and N. Lartillot. Site-heterogeneous mutation-selection models within the PhyloBayes-MPI package. *Bioinformatics*, 30:1020–1021, 2014.

[99] N. Rodrigue, N. Lartillot, and H. Phillipe. Bayesian comparisons of codon substitution models. *Genetics*, 180:1579–1591, 2008.

[100] N. Rodrigue, H. Philippe, and N. Lartillot. Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 107:4629–4634, 2010.

[101] N. D. Rubinstein, A. Faigenboim-Doron, I. Mayrose, and T. Pupko. Evolutionary models accounting for layers of selection in protein-coding genes and their impact on the inference of positive selection. *Mol. Biol. Evol.*, 28:3297–3308, 2011.

[102] P. Schattner and M. Diekhans. Regions of extreme synonymous codon selection in mammalian genes. *Nucleic Acids Res.*, 34:1700–1710, 2006.

[103] K. Scheffler, B. Murrell, and S. L. Kosakovsky Pond. On the validity of evolutionary models with site-specific parameters. *PLOS ONE*, 9:e94534, 2014.

[104] M. P. Scherrer, A. G. Meyer, and C. O. Wilke. Modeling coding-sequence evolution within the context of residue solvent accessibility. *BMC Evol. Biol.*, 12:179, 2012.

[105] M. Schoniger and A. von Haeseler. Performance of the maximum likelihood, neighbor joining, and maximum parsimony methods when sequence sites are not independent. *Syst. Biol.*, 44:533–547, 1995.

[106] G. Sella and A. E. Hirsh. The application of statistical physics to evolutionary biology. *Proc. Natl. Acad. Sci. U.S.A.*, 102:9541–9546, 2005.

[107] B. Sipos, T. Massingham, G. E. Jordan, and N. Goldman. PhyloSim - Monte Carlo simulation of sequence evolution in the R statistical computing environment. *BMC Bioinform.*, 12, 2011.

[108] S. J. Spielman and C. O. Wilke. Pyvolve: A flexible python module for simulating sequences along phylogenies. *PLOS ONE*, 10:e0139047, 2015.

[109] S. J. Spielman and C. O. Wilke. The relationship between dN/dS and scaled selection coefficients. *Mol. Biol. Evol.*, 32:1097–1108, 2015.

[110] C. L. Strope, S. D. Scott, and E. N. Moriyama. indel-Seq-Gen: a new protein family simulator incorporating domains, motifs, and indels. *Mol. Biol. Evol.*, 24:640–649, 2007.

[111] J. Sukumaran and Mark T. Holder. DendroPy: A Python library for phylogenetic computing. *Bioinformatics*, 26:1569–1571, 2010.

[112] Y. Suzuki. Natural selection on the influenza virus genome. *Mol. Biol. Evol.*, 23:1902–1911, 2006.

[113] Y. Suzuki and T. Gojobori. A method for detecting positive selection at single amino acid sites. *Mol. Biol. Evol.*, 16:1315–1328, 1999.

[114] K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.*, 10:512–526, 1993.

[115] A. U. Tamuri, M. dos Reis, and R. A. Goldstein. Estimating the distribution of selection coefficients from phylogenetic data using sitewise mutation-selection models. *Genetics*, 190:1101–1115, 2012.

[116] A. U. Tamuri, N. Goldman, and M. dos Reis. A penalized-likelihood method to estimate the distribution of selection coefficients from phylogenetic data. *Genetics*, 197:257–271, 2014.

[117] S. Tavare. Lines of descent and genealogical processes, and their applications in population genetics models. *Theor. Popul. Biol.*, 26:119–164, 1984.

[118] J. L. Thorne, S. C. Choi, J. Yu, P. G. Higgs, and H. Kishino. Population genetics without intraspecific data. *Mol. Biol. Evol.*, 24:1667–1677, 2007.

[119] J. L. Thorne, N. Lartillot, N. Rodrigue, and S. C. Choi. Codon models as vehicles for reconciling population genetics with inter-specific data. In G. Cannarozzi and A. Schneider, editors, *Codon evolution: mechanisms and models*. Oxford University Press, New York, 2012.

[120] Y. E. Wang, B. Li, J. M. Carlson, H. Streeck, A. D. Gladden, R. Goodman, A. Schneidewind, K. A. Power, I. Toth, N. Frahm, G. Alter, C. Brander, M. Carrington, B. D. Walker, M. Altfeld, D. Heckerman, and T. M. Allen. Protective HLA class I alleles that restrict acute-phase CD8+ T-cell responses are associated with viral escape mutations located in highly conserved regions of human immunodeficiency virus type 1. *J. Virol.*, 83(4):1845–55, 2009.

[121] S. Whelan and N. Goldman. A general empirical model of protein evolution derived from multiple protein families using a maximum likelihood approach. *Mol. Biol. Evol.*, 18:691–699, 2001.

[122] Simon Whelan. Spatial and temporal heterogeneity in nucleotide sequence evolution. *Mol. Biol. Evol.*, 25:1683–1694, 2008.

[123] A. Williford and J. P. Demuth. Gene expression levels are correlated with synonymous codon usage, amino acid composition, and gene architecture in the red flour beetle, Tribolium castaneum. *Mol. Biol. Evol.*, 29:3755–3766, 2012.

[124] N. Yang, R. Nielsen, and M. Hasegawa. Models of amino acid substitution and applications to mitochondrial protein evolution. *Mol. Biol.*

Evol., 15:1600–1611, 1998.

[125] Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *J. Mol. Evol.*, 39:306–314, 1994.

[126] Z. Yang. *Computational Molecular Evolution.* Oxford University Press, 2006.

[127] Z. Yang. PAML 4: Phylogenetic analysis by maximum likelihood. *Mol. Biol. Evol.*, 24:1586–1591, 2007.

[128] Z. Yang and R. Nielsen. Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. *Mol. Biol. Evol.*, 17:32–42, 2000.

[129] Z. Yang and R. Nielsen. Codon-substitution models for detecting molecular adaptation at individual sites along specific lineages. *Mol. Biol. Evol.*, 19:908–917, 2002.

[130] Z. Yang and R. Nielsen. Mutation-selection models of codon substitution and their use to estimate selective strengths on codon usage. *Mol. Biol. Evol.*, 25:568–579, 2008.

[131] Z. H. Yang, R. Nielsen, N. Goldman, and A. M. K. Pedersen. Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics*, 155:431–449, 2000.

[132] V. B. Yap, H. Lindsay, S. Easteal, and G. Huttley. Estimates of the effect of natural selection on protein-coding content. *Mol. Biol. Evol.*, 27:726–734, 2010.

[133] F. Zanini and R. A. Neher. Quantifying selection against synonymous mutations in HIV-1 env evolution. *J. Virol.*, 87:11843–11850, 2013.

[134] J. Zhang, R. Nielsen, and Z. Yang. Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Mol. Biol. Evol.*, 22:2472–2479, 2005.

[135] T. Zhou, W. Gu, and C. O. Wilke. Detecting positive and purifying selection at synonymous sites in yeast and worm. *Mol. Biol. Evol.*, 27:1912–1922, 2010.

[136] T. Zhou and C. O. Wilke. Reduced stability of mRNA secondary structure near the translation-initiation site in dsDNA viruses. *BMC Evol. Biol.*, 11:59, 2011.

[137] Y. O. Zhu, M. L. Siegal, D. W. Hall, and D. A. Petrov. Precise estimates of mutation rate and spectrum in yeast. *Proc. Natl. Acad. Sci. U.S.A.*, 111:E2310–E2318, 2014.

Vita

Stephanie Jill Spielman received her Bachelor of Science in Biology, with Honors, from Brown University in May 2010. She then spent the 2010–2011 year volunteering with coexistence-based education initiatives in Israel. In the fall of 2011, she joined Dr. Wilke's lab through the Ecology, Evolution, and Behavior graduate program at The University of Texas at Austin. After completing her Ph.D., she will be joining the Institute for Genomics and Evolutionary Medicine at Temple University as a Postdoctoral Researcher.

Permanent address: stephanie.spielman@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.