Phylogenetic Stochastic Mapping

Jan Irvahn

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Vladimir Minin, Chair

Elizabeth Thompson

Tyler McCormick

Program Authorized to Offer Degree:
Statistics

University of Washington

Abstract

Phylogenetic Stochastic Mapping

Jan Irvahn

Chair of the Supervisory Committee:
Associate Professor Vladimir Minin
Departments of Statistics and Biology

Phylogenetic stochastic mapping is a method for reconstructing the history of trait changes on a phylogenetic tree relating species/organisms carrying the trait. State-of-the-art methods assume that the trait evolves according to a continuous-time Markov chain (CTMC) and work well for small state spaces. The computations slow down considerably for larger state spaces (e.g. space of codons), because current methodology relies on exponentiating CTMC infinitesimal rate matrices — an operation whose computational complexity grows as the size of the CTMC state space cubed. In this work, we introduce a new approach, based on a CTMC technique called uniformization, that does not use matrix exponentiation for phylogenetic stochastic mapping. Our method is based on a new Markov chain Monte Carlo (MCMC) algorithm that targets the distribution of trait histories conditional on the trait data observed at the tips of the tree. The computational complexity of our MCMC method grows as the size of the CTMC state space squared. Moreover, in contrast to competing matrix exponentiation methods, if the rate matrix is sparse, we can leverage this sparsity and increase the computational efficiency of our algorithm further. Using simulated data, we illustrate advantages of our MCMC algorithm and investigate how large the state space needs to be for our method to outperform matrix exponentiation approaches. We show that even on the moderately large state space of codons our MCMC method can be significantly faster than currently used matrix exponentiation methods.

We apply our new stochastic mapping technique to two data sets. The first concerns the reproductive parity mode of squamates, and the second concerns the evolution of bioluminescent bacterial photophores in cephalopods. In both cases there were concerns that the standard CTMC model of trait evolution for the binary morphological traits was insufficient due to rate matrix heterogeneity across the phylogeny. To address these concerns we developed a Markov modulated Markov process model of trait evolution and integrated this hidden rates model with our matrix exponentiation free stochastic mapping technique. We found that the evidence supporting multiple gains of bioluminescence in cephalopods was mildly attenuated by accounting for potential rate matrix heterogeneity. Conversely, we found that accounting for rate matrix heterogeneity on the squamate phylogeny dramatically changed conclusions about the reproductive parity mode of the most recent common ancestor of squamates. The standard two state CTMC model of trait evolution found insufficient evidence to distinguish between oviparity and viviparity at the root of Squamata while a variety of hidden rates models found strong evidence that the most recent common ancestor of squamates was oviparous.

TABLE OF CONTENTS

Page

List of Figures iii

Chapter 1: Introduction 1
1.1 Phylogenetic Trees 1
1.2 Phylogenetic Comparative Analysis 4
1.3 Phylogenetic Stochastic Mapping 5
1.4 Structure of the Thesis 10

Chapter 2: Review of Statistical Methods 12
2.1 CTMC Model of Evolution 12
2.2 Markov chain Monte Carlo 19

Chapter 3: Phylogenetic Stochastic Mapping without Matrix Exponentiation . . . 23
3.1 New Algorithm 23
3.2 Assessing Computational Efficiency 32
3.3 Numerical Experiments 35
3.4 Discussion 44

Chapter 4: Hidden Rates Model 46
4.1 Introduction 46
4.2 Markov Modulated Markov Process 47
4.3 Stochastic Mapping with Unknown Rate Matrix Parameters 51
4.4 Multiple Trees 60

Chapter 5: Squamates and Cephalopods 62
5.1 Squamata 62
5.2 Cephalopods 69

5.3 Discussion 74
5.4 Model Checking 75

Chapter 6: Discussion and Future Work 88

Bibliography 92

Appendix A: Fixed Rate Matrix Supplement 101
A.1 Posterior Distribution Checks 101
A.2 Tuning Parameter 105
A.3 Convergence 105
A.4 Sparsity 105

Appendix B: Posterior Predictive Plots 113

LIST OF FIGURES

Figure Number		Page
1.1	A primate phylogeny	2
1.2	Example trait history	8
2.1	Example substitution history	14
3.1	Example augmented substitution history	26
3.2	Application of our two Markov kernels to an augmented substitution history	32
3.3	Computation time plotted against state space size	37
3.4	Computation time plotted against tuning parameter	39
3.5	Computation time plotted against state space size for sparse rate matrices	40
3.6	Computation time plotted against tuning parameter for the GY94 codon rate matrix	42
4.1	Example Markov modulated Markov process	48
4.2	Hidden rates model of trait evolution	50
5.1	Squamate phylogeny	64
5.2	Histograms of the posterior distributions concerning the number of transitions between oviparity and viviparity in squamates	66
5.3	Results concerning the reproductive parity mode of the most recent common ancestor of Squamata	68
5.4	Cephalopod phylogeny	71
5.5	Cephalopod results concerning the number of times bacterial photophores separately arose using a 2-state model of evolution	72
5.6	Cephalopod results concerning the number of times bacterial photophores separately arose using a 4-state model of evolution	73
5.7	Simulation comparison of two different restricted prior set distributions for the total number of trait gains	79
5.8	Simulation comparison of two different diffuse prior set distributions for the total number of trait gains	80

5.9 Distribution of the total number of trait gains (n_{01}) for a specific 4-state rate matrix and 70 tip tree. 84
5.10 Posterior predictive plots 86

A.1 Univariate summaries of substitution histories, small state space 102
A.2 Boxplots of posterior dwell times, small state space 103
A.3 Histograms of posterior transtion counts, small state space 104
A.4 Univariate summaries of substitution histories, medium state space 106
A.5 Boxplots of posterior dwell times, medium state space 107
A.6 Histograms of posterior transtion counts, medium state space 108
A.7 Computation time versus Ω 109
A.8 Trace plots 110
A.9 Computation time versus state space size 112

B.1 Posterior predictive plot, 4-state model, simulation 1 113
B.2 Posterior predictive plot, 4-state model, simulation 2 114
B.3 Posterior predictive plot, 4-state model, simulation 3 115
B.4 Posterior predictive plot, 4-state model, simulation 4 116
B.5 Posterior predictive plot, 4-state model, simulation 5 117
B.6 Posterior predictive plot, 4-state model, simulation 6 118
B.7 Posterior predictive plot, 4-state model, simulation 7 119
B.8 Posterior predictive plot, 4-state model, simulation 8 120
B.9 Posterior predictive plot, 4-state model, simulation 9 121
B.10 Posterior predictive plot, 4-state model, simulation 10 122
B.11 Posterior predictive plot, 2-state model, simulation 1 123
B.12 Posterior predictive plot, 2-state model, simulation 2 124
B.13 Posterior predictive plot, 2-state model, simulation 3 125
B.14 Posterior predictive plot, 2-state model, simulation 4 126
B.15 Posterior predictive plot, 2-state model, simulation 5 127
B.16 Posterior predictive plot, 2-state model, simulation 6 128
B.17 Posterior predictive plot, 2-state model, simulation 7 129
B.18 Posterior predictive plot, 2-state model, simulation 8 130
B.19 Posterior predictive plot, 2-state model, simulation 9 131
B.20 Posterior predictive plot, 2-state model, simulation 10 132

B.21 Posterior predictive plot, 4-state model, fixed root, simulation 1 134
B.22 Posterior predictive plot, 4-state model, fixed root, simulation 2 135
B.23 Posterior predictive plot, 4-state model, fixed root, simulation 3 136
B.24 Posterior predictive plot, 4-state model, fixed root, simulation 4 137
B.25 Posterior predictive plot, 4-state model, fixed root, simulation 5 138
B.26 Posterior predictive plot, 4-state model, fixed root, simulation 6 139
B.27 Posterior predictive plot, 4-state model, fixed root, simulation 7 140
B.28 Posterior predictive plot, 4-state model, fixed root, simulation 8 141
B.29 Posterior predictive plot, 4-state model, fixed root, simulation 9 142
B.30 Posterior predictive plot, 4-state model, fixed root, simulation 10 143
B.31 Posterior predictive plot, 2-state model, fixed root, simulation 1 145
B.32 Posterior predictive plot, 2-state model, fixed root, simulation 2 146
B.33 Posterior predictive plot, 2-state model, fixed root, simulation 3 147
B.34 Posterior predictive plot, 2-state model, fixed root, simulation 4 148
B.35 Posterior predictive plot, 2-state model, fixed root, simulation 5 149
B.36 Posterior predictive plot, 2-state model, fixed root, simulation 6 150
B.37 Posterior predictive plot, 2-state model, fixed root, simulation 7 151
B.38 Posterior predictive plot, 2-state model, fixed root, simulation 8 152
B.39 Posterior predictive plot, 2-state model, fixed root, simulation 9 153
B.40 Posterior predictive plot, 2-state model, fixed root, simulation 10 154

ACKNOWLEDGMENTS

All of the work in this thesis was enabled and supported by many people. I would specifically like to thank my advisor, Vladimir Minin, for his patience, cheerfulness, and mentorship. I would also like to thank my committee members, Elizabeth Thompson, Tyler McCormick, and Adam Leaché, for their guidence, feedback, and help.

I am grateful for my classmates who shared in the adventure, notably Theresa Smith, Alex Volfovsky, and Aaron Zimmerman. My parents have always been astonishingly supportive, perhaps this should not surprise me but it does, thank you. And to my partner, Deborah, thank you for your calm presence and for the dances.

Chapter 1
INTRODUCTION

Phylogenetics is the study of evolutionary relationships among groups of organisms. These relationships are described by an object called a phylogenetic tree — a graphical representation of the hierarchical relationship among species or other entities. The shapes of evolutionary trees are informed by similarities and differences in the genotypic and phenotypic characteristics of the species under consideration. Intuitively, species with similar genomes should share common ancestors more recently than do species with more disparate genomes.

1.1 Phylogenetic Trees

New biological species are created in speciation events so that a single species exists before the event and two species exist after the event. The speciation process requires limited gene flow (or no gene flow) between two subgroups of the original species. There are multiple ways to restrict gene flow including physical barriers, different mating seasons, and different mating rituals. Physical barriers such as mountains, oceans, or just isolation by distance restrict genetic flow because individuals in one subgroup never/rarely encounter individuals in the other subgroup. When the sexual reproductive periods of the two subgroups fail to overlap we again encounter a situation with limited or no gene flow. Plants that flower in early summer do not reproduce with plants that flower in late summer. When the mating rituals of the two subgroups differ (such as the songs of birds or insects) we also encounter reproductive isolation because individuals in one subgroup do not respond to the mating rituals of individuals from the other subgroup. In all these settings the lack of gene flow allows different sets of mutations to accumulate in the two different subgroups. Different

selection pressures as well as genetic drift can accentuate the differences between the two gene pools and, with time, the two subgroups can evolve into two new biological species. Before the speciation event there was one species and after the event there were two species. Clearly, this is a simplification of reality but it is useful when modelling evolution. Speciation events are represented on a phylogeny by internal nodes where a single branch splits into two new branches. Speciation events and branches make up the bulk of a phylogenetic tree. The length of branches that connect nodes to each other is often related to time. There are two types of nodes, internal and tip. While the internal nodes represent speciation events, tip nodes are found at the end of branches that do not split. An example phylogeny of primates can be seen in Figure 1.1.

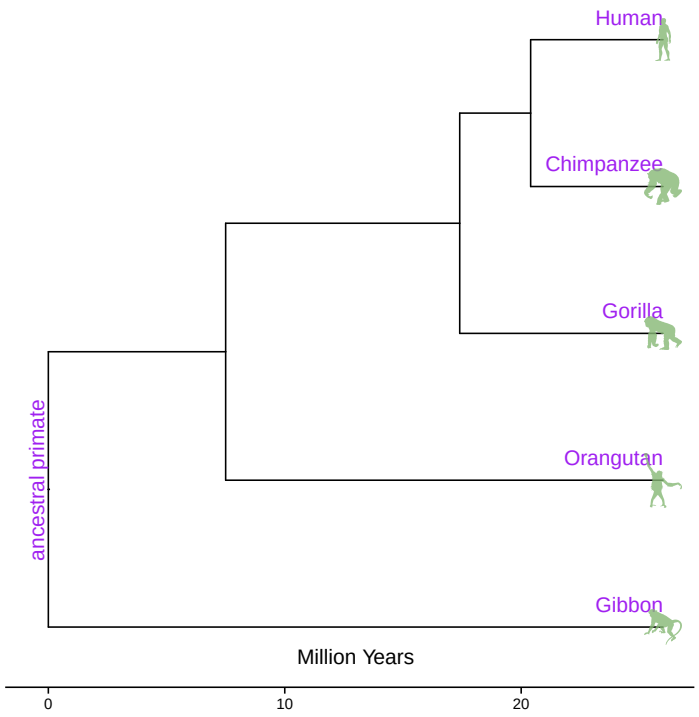

Figure 1.1: A rooted primate phylogeny with branch lengths measured in millions of years.

There are a variety of methods used to create phylogenies; the most common are parsi-

mony, maximum likelihood, and MCMC-based Bayesian methods [Farris, 1970, Felsenstein, 1981, Rodrigue et al., 2008a]. Parsimony creates trees that minimize the total number of necessary trait changes subject to the constraint that the trait history is consistent with observed data. Maximum likelihood and Bayesian methods use a data generating model and estimate parameters of the model including tree topology and branch lengths. Maximum likelihood methods provide parameter estimates and measures of uncertainty in those estimates. Bayesian methods require prior distributions for model parameters and create posterior distributions that also yield parameter estimates with measures of uncertainty.

Phylogenetic trees come in two flavors, rooted and unrooted. A rooted tree contains a root node that is the most recent common ancestor of all the species at the tips of the tree. The oldest part of the tree is found at this root node while the newest parts of the tree are found at the tip nodes. Tip nodes are not, generally, assumed to be contemporaneous, and serial sampling of tip nodes does appear in certain phylogenetic applications. Unrooted trees do not contain a root node and the direction that time flows is not specified on these trees. One can turn a rooted tree into an unrooted tree by removing the root node. One can turn an unrooted tree into a rooted tree by incorporating an outgroup species into the analysis. The outgroup needs to share enough genetic material to enable inference while also being sufficiently different so as to be clearly less related to all the remaining species in the tree than those species are to each other.

Branch lengths represent time, but not always in a straightforward way. Sometimes the length of a branch is directly proportional to time as measured by a clock. Trees that follow this model are callibrated with fossil records and the use of a molecular clock. The molecular clock assumes that the expected number of mutations found at DNA alignment sites increases linearly with time [Zuckerkandl and Pauling, 1962], and is unchanging across the tree. More advanced relaxed clock models allow the rates of molecular evolution to vary in a potentially autocorrelated manner between branches. These models can estimate phylogeny and divergence times jointly, without assuming a strict molecular clock [Drummond et al., 2006, Thorne et al., 1998]. Alternatively, branch lengths may be measured in the expected

number of mutations and no attempt is made to match these branch lengths to clock time. This approach does not allow us to date speciation events.

Rooted trees where branches do correspond to clock time and where all the tip nodes are contemporaneous are called ultrametric. This means the distance between the root node and every tip node is the same. These distances are the sum of the lengths of all the branches that connect the root to the tip node of interest. Generally, trees with branch lengths that represent an expected number of mutations are not ultrametric.

The creation of phylogenetic trees from DNA sequence data is a fascinating component of phylogentics and it is not the topic of this thesis.

1.2 Phylogenetic Comparative Analysis

The topic of this thesis is stochastic mapping, a family of methods for drawing random realizations of the full evolutionary history of trait changes on a phylogenetic tree while conditioning on the observed trait values at the tips of the tree. Stochastic mapping is a type of phylogenetic comparative analysis. This class of methods can investigate correlated evolutionary change between two or more traits. Comparative analysis can also test whether a trait contains a phylogenetic signal, found when closely related species display similar trait values. There are many different types of questions that phylogenetic comparative analysis is used to address [Pagel, 1999b,a, 1993, 1994, Pagel and Meade, 2006, Galtier et al., 1999, Martins and Garland Jr, 1991, Huelsenbeck et al., 2003].

For example, we might be interested in how brain mass varies in relation to body mass. Conventional statistical methods would typically address this question by implicitly assuming that each species is completely unrelated to each other species. Felsenstein [1985] notes that this is a bad assumption to make and shows how the independence assumption is excessively likely to find significant relationships that do not actually exist. To avoid problems that arise from the independence assumptions Felsenstein [1985] proposed the use of phylogenetically independent contrasts, a type of comparative analysis.

Felsenstein [1985]'s phylogenetically independent contrasts were built for continuously

varying traits so alternative methods were developed for discrete traits [Pagel, 1994]. Pagel and Meade [2006] described a Bayesian method intended to detect correlated evolution of discrete binary traits on a phylogeny. This method employed a reversible jump Markov chain Monte Carlo (MCMC) algorithm to switch between models that treat the traits as independently evolving and models that treat the traits as dependently evolving. Pagel and Meade [2006] used the model to investigate the potential coevolution of mating system and female advertisement of estrus in Old World monkeys and great apes. A Bayes factor calculation using the MCMC output found strong support for correlated evolution. Pagel and Meade [2006] concluded that it was unlikely for these primates to be monogamous while advertising estrus anywhere across the primate phylogeny.

Comparative analysis can be used to determine the ancestral state of a trait. For example, consider the trait that is the proportion of nucleotides that are either guanine or cytosine, the G+C content of ribosomal RNA. This proportion is an indicator of the environmental temperature that an organism lives in [Galtier et al., 1999] and can be used to make inference about ancient environmental conditions surrounding creatures early in the tree of life. Galtier et al. [1999] compared 40 eukaryotic, bacterial, and archaebacterial species to reconstruct a tree and estimate the G+C content of their most recent common ancestor. Galtier et al. [1999] estimated a G+C content of 54%, which is generally considered to be incompatible with high temperature survival. This result challenged the common hypothesis that early life was adapted to hot conditions.

Ancestral state reconstruction is actually a subset of phylogenetic stochastic mapping, a valuble tool in computational evolutionary biology. Stochastic mapping aims not just to reconstruct a trait value for a particular ancestor but to reconstruct trait values across an entire phylogeny.

1.3 Phylogenetic Stochastic Mapping

There are many types of traits that are amenable to stochastic mapping procedures. These include morphological traits, geographic locations, and the state of deoxyribonucleic acid

(DNA) at the nucleotide level.

Morphological traits measure some aspect of the physical characteristics of the species under consideration. Some binary morphological traits that have been studied are the presence of horned soldiers among aphids [Huelsenbeck et al., 2003], the parity reproductive mode among squamates [Pyron and Burbrink, 2014, King and Lee, 2015], and the presence of bioluminescent bacterial photophores among cephalopods [Pankey et al., 2014]. Polination systems among flowering plants have more than two distinct states as determined by the different types of animal vectors, such as bees, hummingbirds, and bats [Tripp and Manos, 2008]. Continuous morphological traits, such as brain size or body size can be binned to create traits that take on discrete values.

Phylogeography can be used to reconstruct the geographical movements of ancestral populations. In these situations the trait to be mapped onto a phylogeny is the physical location of a species. Lemey et al. [2009] found that the spatial diffusion of a rapidly evolving flu virus can be reconstructed using stochastic mapping procedures. Using a data set of sequenced Avian influenza A-H5N1 accumulated from 20 major Asian cities Lemey et al. [2009] found that this particular flu virus most likely started spreading from Hong Hong or Guangdong. Otálora et al. [2010] estimated the historical biogeography of *Leptogium furfuraceum*, a species complex of lichen, using stochastic mapping techniques. Otálora et al. [2010] found evidence suggesting transoceanic dispersal was responsible for species divergence and the geographic distribution of the species we see today.

A third application of stochastic mapping involves molecular evolution. The trait of interest that we map onto a phylogeny can be the state of the DNA itself. This may be confusing because many phylogenies are created using DNA data. Mapping DNA state transitions onto a phylogeny is separate and distinct from the creation of phylogenies using DNA data. There are four DNA nucleotide molecules, cytosine (C), guanine (G), adenine (A), and thymine (T). These are the four states that comprise the trait of interest in molecular evolution. In a single column of aligned DNA segments from related species we may observe different trait values for different species. Species A may show an adenine molecule where

species B shows a guanine molecule. These differences arise from genetic mutation events that occurred on the phylogeny relating the two species. Microbiologists are interested in mapping these mutation events to learn about convergent evolution and positive selection. The state space of four unique nucleotides is not the only state space under the purview of molecular evolution. Triplets of nucleotides, codons, code for the twenty standard amino acids. Just as a mutation event can change an adenine molecule into a guanine molecule the same event can change an isoleucine amino acid into a valine amino acid. The size of the state space is twenty, significantly larger than the state space size of nucleotides (four). The amino acid state space allows modelers to explicitly differentiate between synonymous and nonsynonymous mutations; a powerful tool when investigating selection effects. Synonymous mutations are changes on the nucleotide level that do not result in changes on the amino acid level. The actual codons themselves make for an even larger state space whose size is 61 ($4^3 = 64$ but the three stop codons are generally not included in most models). Nucleotides, amino acids, and codons are all state spaces of molecular evolution that can be mapped onto a phylogeny using phylogenetic stochastic mapping.

The act of stochastic mapping can be thought of as an exercise in coloring. Each distinct state is represented by its own unique color. At the tips of a phylogeny we can observe the state of the trait or equivalently, the color of the tree. Phylogenetic stochastic mapping aims to color the rest of the tree, all of the branches and the internal nodes. Our model of evolution posits that there was one true coloring that led to the state of the world as we see it today. A cartoon example involving a small tree and a trait with three distinct states (corresponding to the colors red, blue, and brown) can be found in Figure 1.2. Of course, we do not believe we can exactly reconstruct the one true history of trait transitions. There are an infinite number of histories that are consistent with any set of observed data at the tips of the tree. Instead, we aim to create a distribution of trait histories. Plausible histories, given our model of trait evolution and data, should have relatively high density in our distribution while implausible histories should have relatively low densities.

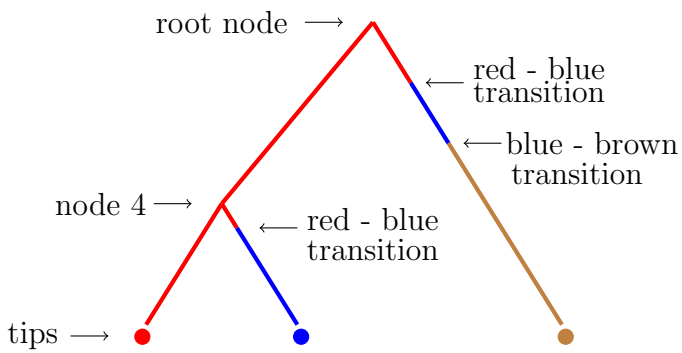

Figure 1.2: Example of a phylogeny where the three colors represent three different trait values.

1.3.1 Statistical Applications of Stochastic Mapping

Biologically, there are many applications for stochastic mapping. From a more statistical point of view stochastic mapping can provide benefits even when the target of inferential interest does not lie in the stochastic history, for example when estimating mutation rates or evolutionary distances between molecular sequences. Three benefits we discuss in this section are computational improvements (better MCMC mixing), a framework for assessing model fit, and robust conclusions in the face of model misspecification. It turns out that stochastic mapping has a useful role to play in the creation of phylogenetic trees. While the creation of phylogenies is a different part of phylogenetics from stochastic mapping the two topics do have some features in common. Stochastic mapping and phylogeny creation both make inference about the rates at which DNA mutates and these rates can not always be assumed to remain constant across nucleotide sites. Stochastic mapping has an important role to play in the development of methods used to create phylogenies from DNA data when the models employed are interested in accounting for different rates of evolution.

Yang [1993] extended Felsenstein [1981]'s maximum likelihood method for estimating phylogenies, allowing mutation rates to vary across nucleotide alignment sites. Huelsenbeck and Nielsen [1999] developed a likelihood ratio test showing that allowing for variation in the

transition/transversion rate bias across sites led to significant improvement in model fit for most protein coding genes. Nielsen and Yang [1998] found strong support of positive selection at specific amino acid sites in the HIV-1 envelope gene after allowing for different rate categories across DNA alignment sites. Lartillot and Philippe [2004] introduced a Bayesian mixture model of amino acid substitution rates, developing a Bayesian version of the site heterogenous maximum likelihood models. Lartillot [2006] further advanced the topic using a data augmentation technique that allowed for a conjugate Gibb's MCMC approach to implementing Bayesian phylogenetic models. The augmented data used in Lartillot [2006] was a full substitution history of DNA mutation events. For purely computational reasons Lartillot [2006] found that stochastic mapping was an important component of MCMC methods with dramatically better mixing than had previously been observed.

Stochastic mapping, initally developed by Nielsen [2002] and subsequently refined by others ([Lartillot, 2006, Hobolth, 2008]), provided a solid statistical foundation for mapping mutations that parsimony was unable to provide. Nielsen [2002] noted the problems with parsimony after using stochastic mapping to infer the number of mutations that occurred on a phylogeny of the hemagglutinin gene of human influenza virus A. He found that parsimony significantly underestimated the number of inferred mutations. Nielsen [2002] used stochastic mapping to address two model fit questions. The first question concerned mutation rate variation across sites and the second model fit question concerned the ratio of the rate of synonymous to nonsynonymous mutations. In addition to computational benefits we see that stochastic mapping is an important tool for assesing model fit in phylogenetic contexts.

Stochastic mapping has been shown to provide results that are robust to some model misspecification. Phylogenetic models of nucleotide evolution usually assume a specific parametric Markovian model that is a simplification of the actual data generating process. This is a sensible approach to take for tractability reasons but also raises concerns about the robustness of the model to such purposeful misspecification. O'Brien et al. [2009], Minin et al. [2011], Lemey et al. [2012] all investigate situations involving phylogenetic model misspecification. Lemey et al. [2012] investigates the role of positive selection while employing

computationally tractable codon partition models. O'Brien et al. [2009] and Minin et al. [2011] both estimate evolutionary distances using simplified (purposely misspecified) models of codon evolution. All of these papers find that with the help of stochastic mapping robust conclusions can be obtained despite the model misspecification.

1.3.2 Hypothesis Testing for Trait Histories

In this thesis, after developing a new stochastic mapping procedure, we investigate two different binary traits. The first trait is the reproductive mode of squamates (snakes and lizards). Some squamates lay eggs (oviparous) and some give live birth (viviparous). Using our new mapping technique we find that the reproductive mode of the most recent ancestor of Squamata was egg laying. The second trait we investigate is the presence of bioluminescent bacterial photophores among cephalopods (tentacled marine animals including octopuses and squid). The evidence shows that this type of bioluminesce likely evolved more than once on the cephalopod phylogeny.

1.4 Structure of the Thesis

In chapter 2 we introduce some of the mathematical tools required for modern phylogenetic stochastic mapping, and describe the standard techniques used to map traits onto a phylogeny. There is a brief description of continuous time Markov chains (CTMC), Nielsen [2002]'s stochastic mapping technique which uses Felsenstein [1981]'s partial likelihood machinery, and MCMC methods.

In chapter 3 we develop a new stochastic mapping technique. State-of-the-art methods work well for small state spaces but the computations slow down considerably for larger state spaces (e.g. space of codons), because current methodology relies on exponentiating CTMC infinitesimal rate matrices. The computational complexity grows as the size of the CTMC state space cubed. We introduce a new approach, based on a CTMC technique called uniformization, that does not use matrix exponentiation for phylogenetic stochastic mapping. Our method is based on a new MCMC algorithm that targets the distribution of trait histories

conditional on the trait data observed at the tips of the tree. The computational complexity of our MCMC method grows as the size of the CTMC state space squared. Moreover, in contrast to competing matrix exponentiation methods, if the rate matrix is sparse, we can leverage this sparsity and increase the computational efficiency of our algorithm further. Using simulated data, we illustrate advantages of our MCMC algorithm and investigate how large the state space needs to be for our method to outperform matrix exponentiation approaches. We show that even on the moderately large state space of codons our MCMC method can be significantly faster than currently used matrix exponentiation methods.

In chapter 4 we extend our new stochastic mapping technique to incorporate uncertainty both in rate matrix parameters and in phylogenetic tree reconstruction. We incorporate rate matrix uncertainty by proposing new rate matrix parameters at each iteration of our MCMC and applying the standard Metropolis-Hastings accept/reject procedure. We incorporate uncertainty in tree reconstruction by averaging over a predetermined set of trees. In chapter 4 we also develop a new rate matrix parameterization of a hidden rates model for binary traits. Hidden rates models are intended to be used when the rates at which traits evolve are slow on some parts of a phylogeny and fast on other parts.

In chapter 5 we apply our new stochastic mapping technique to two data sets, one involving squamates and the other involving cephalopods. We find that using a hidden rates model with the squamate data set yields conclusions that are different from what we find with a simple two state model. For the cephalopod data set we do not find large differences between the hidden rates model and the simple two state model.

Chapter 2

REVIEW OF STATISTICAL METHODS

Our review begins with our model of trait evolution. The trait evolves along a single branch of a phylogenetic tree according to a CTMC. A CTMC is a mathemetical model that describes a stochastic process with the Markov property. At each time, t, the Markov chain, X_t, is in one of s distinct states. The Markov property is satisfied when the future state of a Markov chain depends only on the current state and not on the history of the chain. Mathematically this can be written as, $\Pr(X_{t+\Delta t} = k|X_{[0,t]}) = \Pr(X_{t+\Delta t} = k|X_t)$.

The time spent in a state before the chain jumps to a new state is governed by an infinitesimal $s \times s$ rate matrix, $\mathbf{Q} = \{q_{kh}\}$. The rate matrix is constructed so that the sum of the elements in each row is zero, the k^{th} diagonal element of $\mathbf{Q}$ is negative and its absolute value is equal to the sum of the other elements from row k. The k^{th} diagonal element, $q_{kk} \equiv q_k$, governs the rate at which the CTMC leaves state k. The time spent in state k before the chain transitions to a new state is distributed exponentially with rate $-q_k$. When the Markov chain leaves state k the chain transitions to a new state h that is randomly selected with probability, $q_{kh}/|q_k|$. The next section extends our CTMC model of evolution from a single branch to an entire phylogeny.

2.1 CTMC Model of Evolution

We start with a trait of interest, $X(t)$, and a rooted phylogenetic tree with n tips and $2n-2$ branches. We assume that the phylogeny and its branch lengths, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{2n-2})$, are known to us. The trait can be in one of s distinct states, $\{1, \ldots, s\}$, at any particular place on the tree. We follow standard phylogenetic practice and assume that the trait evolves along the phylogenetic tree by the following stochastic process. First, a trait value is drawn at the

root of the tree from an initial distribution $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_s)$. Next, starting with the root state, we use the CTMC rate matrix $\mathbf{Q}$ to produce two *independent* CTMC trajectories along the branches leading to the two immediate descendants of the root node. After generating a CTMC trajectory along a branch we necessarily have generated a state for the child node of the branch. We proceed recursively by conditioning on a parent node and evolving the same CTMC independently along the two branches leading to the parent's children nodes. The random process stops when we reach the tips — nodes that have no descendants. Trait states at the tips of the tree are observed, while the trait values everywhere else on the tree are considered missing data. We collect the observed data into a vector $\mathbf{y}$.

A substitution history for a phylogenetic tree is the complete list of transition events (CTMC jumps), including the time of each event (location on the tree) and the type of the transition event (e.g., $2 \to 1$ transition). This state history can be encoded in a set of vectors, two vectors for each branch. Suppose branch i has n_i transitions so the full state history for branch i can be described by a vector of state labels, $\mathbf{s}_i = (s_{i0}, ..., s_{in_i})$, and a vector of intertransition times, $\mathbf{t}_i = (t_{i0}, ..., t_{in_i})$. Let $\mathcal{S}$ be the collection of all the $\mathbf{s}_i$ vectors and let $\mathcal{T}$ be the collection of all the $\mathbf{t}_i$ vectors, forming the full substitution history, $(\mathcal{S}, \mathcal{T})$. See Figure 2.1 for a substitution history example. The tree in Figure 2.1 has four branches so the collection of state labels is $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4\}$, where $\mathbf{s}_1 = (1)$, $\mathbf{s}_2 = (1, 2)$, $\mathbf{s}_3 = (3, 1)$, and $\mathbf{s}_4 = (3)$. The collection of intertransition times is $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4\}$, where $\mathbf{t}_1 = (3.2)$, $\mathbf{t}_2 = (0.64, 2.56)$, $\mathbf{t}_3 = (2.4, 2.4)$, and $\mathbf{t}_4 = (8)$.

The likelihood of a realization of the CTMC for a phylogenetic tree is,

$$\begin{aligned} P(\mathcal{S}, \mathcal{T}) &= \pi_{\text{root}} \prod_{i=1}^{2n-2} \left(\left(\prod_{d=1}^{n_i} |q_{s_{i,d-1}}| e^{q_{s_{i,d-1}} t_{i,d-1}} \frac{q_{s_{i,d-1} s_{i,d}}}{|q_{s_{i,d-1}}|} \right) e^{q_{s_{i,n}} t_{i,n}} \right), \\ &= \pi_{\text{root}} \prod_{i=1}^{2n-2} \left(\prod_{d=1}^{n_i} \left(q_{s_{i,d-1} s_{i,d}} \right) \exp\left(\int_0^{\beta_i} Q_{s_i(t)} dt \right) \right). \end{aligned}$$

This likelihood begins with the probability that the substitution history starts with the trait value found at the root, π_{root}. The product indexed by i combines the likelihoods of the substitution histories for each branch (there are $2n - 2$ branches). The product indexed

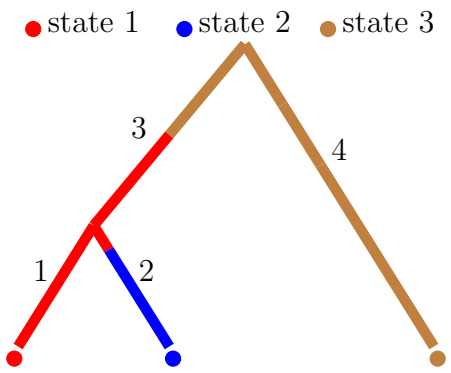

Figure 2.1: Example substitution history, $(\mathcal{S}, \mathcal{T})$, with a state space size of 3.

by d computes the likelihood of a single branch with n_i transitions. The likelihood of a single branch is a combination of multiple exponential distributions and multiple multinomial distributions. The exponential distributions account for how long the trait remains in a particular state before a transition event and the multinomial distributions account for the transition probabilities between states. All together we have the likelihood of a substitution history on a phylogenetic tree.

The goal of stochastic mapping is to be able to compute properties of the distribution of the substitution history of a phylogenetic tree conditional on the observed states at the tips of the tree, $\mathrm{p}(\mathcal{S}, \mathcal{T}|\mathbf{y})$. Actually making inference about substitution histories can be difficult partially because the state space of substitution histories is an infinite dimensional space. To address this difficulty Nielsen [2002] developed a Monte Carlo approach that allows us to sample independent realisations from $\mathrm{p}(\mathcal{S}, \mathcal{T}|\mathbf{y})$. Given a large enough number of independent samples we can describe any aspect of the conditional density of substitution histories with a high degree of accuracy. For example, we could describe a distribution concerning the number of transitions between two specific trait values. We could produce the probability that an ancestral node was in a particular state. We could estimate the amount of time spent in a particular state across the tree. All of these properties can be described for subsets of the phylogeny if, for example, there were a particular clade of interest.

Minin and Suchard [2008] used the conditional density, $p(\mathcal{S}, \mathcal{T}|\mathbf{y})$, to estimate the number of times a binary trait changed states in its evolutionary history. Minin and Suchard [2008] used the same techniques to obtain the average number of mutations at a specific alignment site in nucleotide data. Multivariate summaries of $p(\mathcal{S}, \mathcal{T}|\mathbf{y})$ include Siepel et al. [2006]'s use of stochastic mapping to obtain a joint distribution for the number of transitions on a subtree and the complement of the subtree.

2.1.1 Nielsen's Monte Carlo Sampler

Nielsen [2002] proposed the basic framework that state-of-the-art phylogenetic stochastic mapping currently uses. His approach samples directly from the conditional distribution, $p(\mathcal{S}, \mathcal{T}|\mathbf{y})$, in three steps. First, one calculates partial likelihoods using Felsenstein's algorithm [Felsenstein, 1981]. The partial likelihood matrix records the likelihood of the observed data at the tips of the tree beneath each node after conditioning on the state of said node. This requires calculating transition probabilities for each branch via matrix exponentiation. Second, one recursively samples internal node states, starting from the root of the tree. Third, one draws realizations of CTMC trajectories on each branch conditional on the sampled states at the branch's parent and child nodes. The last step can be accomplished by multiple algorithms reviewed in Hobolth and Stone [2009].

Nielsen [2002]'s sampler starts by using Felsenstein's algorithm to compute a partial likelihood matrix $\mathbf{L} = \{l_{jk}\}$, where l_{jk} is the probability of the observed tip states below node j given that node j is in state k [Felsenstein, 1981]. The matrix $\mathbf{L}$ has $(2n-1)$ rows and s columns because there are $(2n-1)$ nodes (including the tips) and there are s states. Starting at the tips, we work our way up the tree calculating partial likelihoods at internal nodes as we go. We need to calculate the partial likelihood at both child nodes before calculating the partial likelihood at a parent node as described in Felsenstein [1981]. The algorithm is initialized by setting the elements of each row corresponding to a tip node to zero everywhere except for the column corresponding to the observed state of that tip. The matrix value at this entry is set to 1. Next, we calculate the partial likelihoods for all

the internal nodes. Suppose branch i connecting parent node p to child node c has length β_i so the probability transition matrix for branch i is $\mathbf{E}(\beta_i) \equiv e^{\mathbf{Q}\beta_i}$. The probability of transitioning from state h to state k along branch i is $\mathrm{e}(i)_{hk}$, the $(h,k)^{\text{th}}$ element of $\mathbf{E}(\beta_i)$. We refer to the state of node j as y_j. The probability of observing the tip states below node c conditional on node p being in state h is

$$g_{pch} = \sum_{k=1}^{s} \Pr(y_c = k | y_p = h) l_{ck} = (\mathbf{e}(\beta_i)_{h-}) \, (\mathbf{l}_{c-})^T,$$

where $\mathbf{l}_{c-} = (l_{c1}, \ldots, l_{cs})$. If node c is a tip then conditioning on the tip states below c is the same as conditioning on the state of tip c. We combine the probabilities, g_{pch}, for each state h into a single vector, $\mathbf{g}_{pc-}$, and then create the same type of vector for the second branch below node p, $\mathbf{g}_{pd-}$. Element wise multiplication of the two vectors yields the vector of partial likelihoods for node p:

$$(\mathbf{l}_{p-})^T = \mathbf{g}_{pc-} * \mathbf{g}_{pd-}.$$

After working our way up the tree we have the matrix of partial likelihoods, $\mathbf{L}$.

Sampling internal node states Starting at the root we work our way down the tree sampling the states of internal nodes conditional on tip states, the length of each branch, and previously sampled internal node states. The prior probability that the root is in state k is the k^{th} element of the probability vector $\boldsymbol{\pi}$. The probability that the root is in state k given the states of all the tip nodes is,

$$\Pr(y_{\text{root}} = k | \mathbf{y}) = \frac{\Pr(y_{\text{root}} = k \;\&\; \mathbf{y})}{\Pr(\mathbf{y})} = \frac{\Pr(y_{\text{root}} = k) \Pr(\mathbf{y} | y_{\text{root}} = k)}{\sum_{h=1}^{s} \Pr(y_{\text{root}} = h) \Pr(\mathbf{y} | y_{\text{root}} = h)} = \frac{\pi_k l_{(\text{root } k)}}{\mathbf{l}_{(\text{root } -)} \boldsymbol{\pi}}.$$

Once we calculate the probability of the root being in each possible state we sample the state of the root from the multinomial distribution with probabilities we just computed. Next, we sample all non-root, non-tip nodes. Without loss of generality, let us consider node c connected to its parent node, node p, by branch i. Suppose node p's previously sampled state is h and the length of branch i is β_i. The vector containing observed tip states at the

eventual descendants of node c is $\mathbf{d}_c$. The probability that node c is in state k given node p is in state h and given the state of the tips below node c is

$$\Pr(y_c = k | y_p = h \;\&\; \mathbf{d}_c) = \frac{\Pr(y_c = k | y_p = h)\Pr(\mathbf{d}_c | y_c = k)}{\sum_{k=1}^{s} \Pr(y_c = k | y_p = h)\Pr(\mathbf{d}_c | y_c = k)} = \frac{\mathrm{e}(\beta_i)_{hk} \mathrm{l}_{ck}}{\mathbf{e}(\beta_i)_{h-}(\mathbf{l}_{c-})^T}. \quad (2.1)$$

Starting with the root we can now work our way down the tree sampling the states of each node from the multinomial distributions with probabilities we just described.

End Point Conditioned Branch State Sampling The third step in Nielsen [2002]'s stochastic mapping procedure requires sampling CTMC trajectories along each branch. These trajectories need to be sampled independently conditional on the states at both ends of the branch because the states of all the internal nodes were sampled in the second step. To this end, Nielsen [2002] proposed a modified form of rejection sampling.

First, consider an unmodified version of rejection sampling for a branch with length t_i. We need to sample the state of a branch, X_t for all times $t \in [0, t_i]$, conditional on the parent node being in state y_p, $X_0 = y_p$, and the child node being in state y_c, $X_{t_i} = y_c$. Starting at the parent node we move along the branch sampling states as we go. We initialize the current state of the branch, X_t, at time $t = 0$, with the state of the parent node, y_p. We initialize the length of the remaining unsampled portion of the branch, T, with the total length of the branch, t_i.

First, we draw a time τ from an exponential distribution with rate $|q_{y_p}|$. If $\tau \geq T$ we are done, the branch remains in the current state X_t without any new transition events. If $\tau < T$ we sample a new state, k, from a multinomial distribution with probability $q_{y_p k}/|q_{y_p}|$, update the current state of the branch, X_t, to be k, and update the length of the unsampled portion of the branch by replacing T with $T - \tau$. We now repeat the procedure starting with a new state and a smaller branch segment. When the entire branch is sampled we check the ending state, X_{t_i}. If X_{t_i} is y_c we accept the proposed trajectory. If X_{t_i} is not y_c we reject the proposed trajectory and propose a new trajectory in the same way.

The main difficulty with this procedure is that frequently the simulated X_{t_i} will not be

y_c. When the sampled path does not result in the desired end point we must reject the path and try again. Nielsen [2002] modified the procedure to improve acceptance rates. The modified procedure is unchanged whenever the states of the two endpoints of the unsampled portion of the branch are the same. When the two endpoints are different we know that at least one transition event occurred and we can sample the time to the first transition event exactly. If the unsampled segment of the branch starts in state k the density of the time to the next transition event is,

$$f(\tau|\tau < T) = \frac{-q_k e^{q_k T}}{1 - e^{q_k T}} \quad \text{for} \quad 0 \leq \tau \leq T.$$

As noted in Hobolth and Stone [2009], sampling τ in this context can be accomplished by sampling u from a Uniform(0,1) distribution and transforming u via the inverse of the cumulative distribution function,

$$F^{-1}(u) = -\log\left(1 - u(1 - e^{q_k T})/|q_k|\right).$$

Even with modified rejection sampling there are times when the transition from y_p to y_c is so unlikely almost all sample paths will be rejected. In these settings we can turn to uniformization, another endpoint conditioned sampling scheme reviewed in Hobolth and Stone [2009] that does not require the user to reject any sampled paths. The uniformization technique is a critical component of the new stochastic mapping procedure explained in chapter 3. Uniformization allows us to sample X_t by creating an auxilliary stochastic process, Y_t. While X_t is a continuous time Markov process Y_t is a discrete time Markov process with transition probability matrix, $\mathbf{B} = \mathbf{I} + \mathbf{Q}/\Omega$, where $\Omega \geq \max_k q_k$. Unlike X_t, our auxilliary Markov process, Y_t, allows virtual transitions (from state k to state k). The locations of all the transitions in Y_t are determined by an independent Poisson process with rate Ω. The probability transition matrix for our auxilliary stochastic process after time t is,

$$P_{Y_t} = \sum_{m=0}^{\infty} \frac{e^{-\Omega t}(\Omega t)^m}{m!}\mathbf{B}^m = e^{-\Omega t}\sum_{m=0}^{\infty}\frac{(\Omega t\mathbf{B})^m}{m!} = e^{-\Omega t}e^{\Omega \mathbf{B} t} = e^{\Omega(\mathbf{B}-\mathbf{I})t} = e^{\mathbf{Q}t} = P_{X_t}.$$

Our auxillary stochastic process Y_t subordinated to a Poisson process produces identical transition probability matrices as the original Markov process, X_t.

Uniformization allows us to sample endpoint conditioned paths by sampling the number and locations of transitions events first and sampling the actual states visited second. We use Bayes rule to sample from the distribution of the number of transitions, N, conditional on endpoint states,

$$\Pr(N = m|X_0 = k, X_t = h) = \frac{\Pr(X_0 = k, X_t = h|N = m)\Pr(N = m)}{\Pr(X_0 = k, X_t = h)} = \frac{(\mathbf{B}^m)_{kh}\frac{e^{-\Omega t}(\Omega t)^m}{m!}}{(e^{\mathbf{Q}t})_{kh}}. \tag{2.2}$$

Folowing Hobolth and Stone [2009], we can sample from the distribution defined by Equation 2.2 by first sampling u from a Uniform(0,1) and letting m be the first time the cumulative sum of 2.2 exceeds u. After sampling the number of transitions, m, we distribute them uniformly across the interval. Finally, we sample states sequentially according a discrete time Markov chain with transition probability matrix, $\mathbf{B}$, conditional on the starting and ending states. This amounts to sampling from a multinomial distribution with probabilities determined by,

$$\Pr(Y_i = s|Y_{i-1} = k, Y_m = h, m) = \frac{\mathbf{B}_{ks}(\mathbf{B}^{m-i})_{sh}}{(\mathbf{B}^m)_{kh}}.$$

We revisit uniformization in chapter 3 to make use of the work of Rao and Teh [2011] on sampling hidden trajectories for continuous time hidden Markov models.

2.2 Markov chain Monte Carlo

The Monte Carlo approach to stochastic mapping of Nielsen [2002] works well for small CTMC state spaces but the approach becomes time consuming for larger state space sizes because of the reliance on matrix exponentiation. In chapter 3 we develop a stochastic mapping algorithm that avoids matrix exponentiation and instead relies on the power of MCMC algorithms which we review in this section.

Bayesian statistics provides the framework that a researcher can use to update her beliefs about unknowns that describe some aspect of the state of nature. We often employ mathematical models to describe observed data and to predict future data. These models rely on

probability distributions to describe uncertainty in our predicitions and in our knowledge. Our data, $\mathbf{y}$, are usually considered to be a sample from a probability distribution indexed by $\boldsymbol{\theta}$, $p(\mathbf{y}|\boldsymbol{\theta})$. Often, our target of inference is $\boldsymbol{\theta}$ and we are interested in creating a posterior distribution, $p(\boldsymbol{\theta}|\mathbf{y})$ that describes our beliefs about $\boldsymbol{\theta}$ after having observed $\mathbf{y}$. Before observing $\mathbf{y}$ our beliefs about $\boldsymbol{\theta}$ are encapsulated in a prior distribution, $p(\boldsymbol{\theta})$. The posterior distribution is obtained by combining the prior distribution, $p(\boldsymbol{\theta})$, with the sampling model, $p(\mathbf{y}|\boldsymbol{\theta})$ (also called the likelihood) via Bayes rule,

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\boldsymbol{\theta}, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{2.3}$$

Calculating posterior quantities of interest may be difficult or impossible even when we have the posterior distribution from Equation 2.3. When these calculations prove obstinate we turn to Monte Carlo approximation. If we can generate random samples from the posterior distribution then any posterior quantity of interest can be calculated to an arbitrary degree of precision by examining a large number of samples from the posterior. Nielsen's stochastic mapping procedure is a Monte Carlo approach for creating posterior distributions of substitution histories, which are not parameters, but auxiliary variables/missing data [Nielsen, 2002].

There are times when generating independent samples from the full posterior distribution is too difficult to be useful, especially in high dimensional spaces. In these situations we can turn to MCMC simulation to help. Our goal in this setting is to create an ergodic Markov chain whose stationary distribution is the same as the posterior distribution of interest. This approach is useful because the ergodic theorem tells us that if $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_N)$ is a realization from an irreducible and positive recurrent Markov chain with stationary distribution $\boldsymbol{\pi}$, then for an integrable function, $g(\boldsymbol{\theta})$,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} g(\boldsymbol{\theta}_i) \to_p E_{\boldsymbol{\pi}} g(\boldsymbol{\theta}).$$

For a Markov chain to be irreducible it must be able to transition between any two parts of the state space in a finite number of transitions. A Markov chain is positive recurrent if it

returns to any initial state in a finite number of transitions (applicable when the state space of $\boldsymbol{\theta}$ is countable). For general state spaces the positive recurrent requirement is replaced with Harris recurrence. This leaves us with the problem of constructing a Markov chain with a specific target distribution, $\boldsymbol{\pi}$.

The Metropolis-Hastings algorithm is a powerful method that allows us to contruct a Markov chain whose stationary distribution, $\boldsymbol{\pi}$, is the posterior distribution, $\boldsymbol{\pi}(\boldsymbol{\theta}|\mathbf{y})$. The algorithm was published by Metropolis et al. [1953] and later expanded by Hastings [1970]. Our chain, $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N)$, starts in the state $\boldsymbol{\theta}_0$ and iteratively produces the next element of the chain as follows,

1. Sample new elements, $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}'|\boldsymbol{\theta}_t)$,

2. Compute an acceptance ratio, R, where
$$R = \min\left(1, \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_t|\mathbf{y})q(\boldsymbol{\theta}'|\boldsymbol{\theta}_t)}{\boldsymbol{\pi}(\boldsymbol{\theta}'|\mathbf{y})q(\boldsymbol{\theta}_t|\boldsymbol{\theta}')}\right) = \min\left(1, \frac{\boldsymbol{\pi}(\mathbf{y}|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t)q(\boldsymbol{\theta}'|\boldsymbol{\theta}_t)}{\boldsymbol{\pi}(\mathbf{y}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')q(\boldsymbol{\theta}_t|\boldsymbol{\theta}')}\right),$$

3. Set $\boldsymbol{\theta}_{t+1}$ to be $\boldsymbol{\theta}'$ with probability R (and set $\boldsymbol{\theta}_{t+1}$ to be $\boldsymbol{\theta}_t$ with probability $1 - R$).

We shold note that while we can not compute $\boldsymbol{\pi}(\boldsymbol{\theta}|\mathbf{y})$, we can compute the ratio, $\frac{\boldsymbol{\pi}(\boldsymbol{\theta}_t|\mathbf{y})}{\boldsymbol{\pi}(\boldsymbol{\theta}'|\mathbf{y})}$, because the normalizing constant, $\pi(\mathbf{y})$, cancels out. The proposal density, q, can be quite general though it should be chosen with care to attain good mixing of the Markov chain. If the proposal suggests values with very low likelihoods most of our proposed updates will be rejected and the chain will mix very slowly. If the proposal does not create values that are sufficiently different from current values our proposals will likely be accepted but again, it will take the chain a long time to properly explore the parameter space.

One version of the Metropolis-Hastings algorithm that mixes very well is called a Gibbs sampler. In this setting we can sample from full conditional distributions, $f(\theta_i|\boldsymbol{\theta}_{-i})$, where $\boldsymbol{\theta}_{-i}$ is composed of all the elements of $\boldsymbol{\theta}$ except θ_i, the i^{th} component. A Gibbs sampler updates each element of $\boldsymbol{\theta}$ in turn, proposing a new value for $\boldsymbol{\theta}_i$ by sampling from $f(\theta_i|\boldsymbol{\theta}_{-i})$. The Metropolis-Hastings acceptance ratios for these proposals are always one because the proposal density is a full conditional distribution. There is no need for an accept/reject step.

In chapter 3 we develop a new method that makes use of full conditional distributions to approximate the posterior distribution of substitution histories conditional on observed data at the tips of a phylogenetic tree. In chapter 4 we expand the state space of the Markov chain, introducing Metropolis-Hastings accept/reject steps for our rate matrix parameter proposals.

Chapter 3

PHYLOGENETIC STOCHASTIC MAPPING WITHOUT MATRIX EXPONENTIATION

3.1 New Algorithm

Stochastic mapping, initially developed by Nielsen [2002] and subsequently refined by others [Lartillot, 2006, Hobolth, 2008], assumes that discrete traits of interest evolve according to a CTMC. Random sampling of evolutionary histories, conditional on the observed data, is accomplished by an algorithm akin to the forward filtering-backward sampling algorithm for hidden Markov models (HMMs) [Scott, 2002]. However, since stochastic mapping operates in continuous-time, all current stochastic mapping algorithms require computing CTMC transition probabilities via matrix exponentiation — a time consuming and potentially numerically unstable operation, when the CTMC state space grows large. de Koning et al. [2010] recognized the computational burden of the existing techniques and developed a faster, but approximate, stochastic mapping method based on time-discretization. We propose an alternative, *exact* stochastic mapping algorithm that relies on recent developments in the continuous-time HMM literature.

Rao and Teh [2011] used a CTMC technique called uniformization to develop a method for sampling hidden trajectories in continuous time HMMs. The use of uniformization in this context is not new, but all previous methods produced independent samples of hidden trajectories with the help of matrix exponentiation — an operation with algorithmic complexity $\mathcal{O}(s^3)$, where s is the size of the CTMC state space [Fearnhead and Sherlock, 2006]. Rao and Teh [2011] constructed a Markov chain Monte Carlo (MCMC) algorithm targeting the posterior distribution of hidden trajectories. Their new method eliminates the need for matrix exponentiation and results in an algorithm with complexity $\mathcal{O}(s^2)$. Moreover, the

method of Rao and Teh [2011] can further increase its computational efficiency by taking advantage of sparsity of the CTMC rate matrix. Here, we take the method of Rao and Teh [2011] and extend it to phylogenetic stochastic mapping.

As in the original method of Rao and Teh [2011], our new stochastic mapping method must pay a price for bypassing the matrix exponentiation step. The cost of the improved algorithmic complexity is the replacement of Monte Carlo in the state-of-the-art stochastic mapping with MCMC. Since Monte Carlo, if practical, is generally preferable to MCMC, it is not immediately clear that our new algorithm should be an improvement on the original method in all situations. We perform an extensive simulation study, comparing performance of our new MCMC method with a matrix exponentiation method for different sizes of the state space. We conclude that, after accounting for dependence of trait history samples, our new MCMC algorithm can outperform existing approaches even on only moderately large state spaces ($s \sim 30$). We demonstrate additional computational efficiency of our algorithm when taking advantage of sparsity of the CTMC rate matrix. Since we suspect that our new method can speed up computations during studies of protein evolution, we examine in detail a standard GY94 codon substitution model ($s = 61$) [Goldman and Yang, 1994]. We show that our new method can reduce computing times of state-of-the-art stochastic mapping by at least of factor of ten when working with this model. The last finding is important, because state-of-the-art statistical methods based on codon models often grind to a halt when applied to large datasets [Valle et al., 2014].

3.1.1 CTMC Uniformization

An alternative way to describe the CTMC model of evolution on a phylogenetic tree uses a homogenous Poisson process coupled with a discrete time Markov chain (DTMC) that is independent from the Poisson process. The intensity of the homogenous Poisson process, Ω, must be greater than the largest rate of leaving a state, $\max_k |q_{kk}|$. The generative process that produces a substitution history on a phylogenetic tree first samples the total number of DTMC transitions over the tree, N, drawn from a Poisson distribution with mean equal

to $\Omega \sum_{i=1}^{2n-2} \beta_i$ — the product of the Poisson intensity and the sum of all the branch lengths. The locations/times of the N transitions are then distributed uniformly at random across all the branches of the tree. These transition time points separate each branch into segments. The intertransition times (the length of each segment) for branch i compose the vector, $\mathbf{w}_i$, where the sum of elements of this vector equal the branch length β_i. The state of each segment evolves according to a DTMC with transition probability matrix $\mathbf{B} = \{b_{kh}\}$ satisfying $\mathbf{B} = \mathbf{I} + \mathbf{Q}/\Omega$.

Again, the uniformized generative process samples a state at the root of the tree and works down the tree sampling the state of each branch segment sequentially. Conditional on the previous/ancestral segment being in state k, we sample the current segment's state from a multinomial distribution with probabilities $(b_{k1}, \ldots, b_{ks})$. The states of each segment of branch i compose the vector, $\mathbf{v}_i$. It is important to note that the stochastic transition matrix $\mathbf{B}$ allows the DTMC to transition from state k to state k, i.e., self transitions are allowed. Intuitively, the dominating homogenous Poisson process produces more transition events (on average) than we would expect under the CTMC model of evolution. The DTMC allows some of the transitions generated by the Poisson process to be self transitions so that the remaining "real" transitions and times between them yield the exact CTMC trajectories we desire [Jensen, 1953].

An augmented substitution history of a phylogenetic tree encodes all the information in $(\mathcal{S}, \mathcal{T})$ and adds virtual jump times as seen in Figure 3.1. The notation describing an augmented substitution history is similar to the notation used to describe a substitution history. One branch is fully described by two vectors. Let branch i have m_i jumps (real and virtual) and again, $\mathbf{v}_i = (v_{i0}, ..., v_{im_i})$ is a vector of state labels, $\mathbf{w}_i = (w_{i0}, ..., w_{im_i})$ is a vector of intertransition times, $\mathcal{V}$ is the collection of all the $\mathbf{v}_i$ vectors, and $\mathcal{W}$ is the collection of all the $\mathbf{w}_i$ vectors. The augmented state history is $(\mathcal{V}, \mathcal{W})$. The tree in Figure 3.1 has four branches so the collection of state labels is $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$, where $\mathbf{v}_1 = (1, 1)$, $\mathbf{v}_2 = (1, 2)$, $\mathbf{v}_3 = (3, 1)$, and $\mathbf{v}_4 = (3, 3)$. The collection of intertransition times is $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4\}$, where $\mathbf{w}_1 = (1.6, 1.6)$, $\mathbf{w}_2 = (0.64, 2.56)$, $\mathbf{w}_3 = (2.4, 2.4)$, and $\mathbf{w}_4 =$

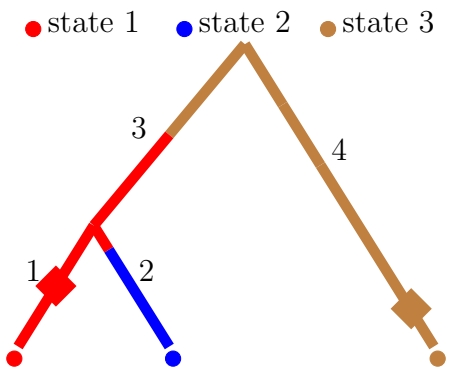

Figure 3.1: Example augmented substitution history, $(\mathcal{V}, \mathcal{W})$, with a state space size of 3.

$(7, 1)$. The locations/times of each virtual jump on branch i is represented by a vector $\mathbf{u}_i = (u_{i1}, .., u_{i(m_i - n_i)})$. For example, the distance from the parent node of branch i to the d^{th} virtual jump is u_{id}. The collection of the $\mathbf{u}_i$ vectors, fully determined by $(\mathcal{V}, \mathcal{W})$, is denoted by $\mathcal{U}$. In Figure 3.1 the collection of virtual jump times is $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\}$ where $\mathbf{u}_1 = (1.6)$, $\mathbf{u}_2 = ()$, $\mathbf{u}_3 = ()$, and $\mathbf{u}_4 = (7)$.

3.1.2 New MCMC Sampler

Equipped with notation describing the CTMC model of evolution and a companion uniformization process, we now turn our attention to making inference about a phylogenetic tree state history conditional on observed data. We investigate the situation where the tree topology is fixed, branch lengths are fixed, and the rate matrix parameters are all known and fixed. The goal is to construct an ergodic Markov chain on the state space of augmented substitution histories with the stationary distribution $\mathrm{p}(\mathcal{V}, \mathcal{W}|\mathbf{y})$.

Our MCMC sampler uses two Markov kernels to create a Markov chain whose stationary distribution is $\mathrm{p}(\mathcal{V}, \mathcal{W}|\mathbf{y})$. The first kernel samples from $\mathrm{p}(\mathcal{V}|\mathcal{W}, \mathbf{y})$ — the distribution of states on the tree conditional on tip states and the jump locations on each branch. A Markov chain that sequentially draws from this full conditional has $\mathrm{p}(\mathcal{V}, \mathcal{W}|\mathbf{y})$ as its stationary distribution. The first kernel, $\mathrm{p}(\mathcal{V}, \mathcal{W}|\mathbf{y})$, is not ergodic by itself because the set of transition

times, $\mathcal{W}$, is not updated. To create an ergodic Markov chain we introduce a second Markov kernel to sample from $\mathrm{p}(\mathcal{U}|\mathcal{S},\mathcal{T},\mathbf{y})$ — the distribution of virtual transitions conditional on the substitution history. Again, drawing from the full conditional of $\mathcal{U}$ ensures that $\mathrm{p}(\mathcal{V},\mathcal{W}|\mathbf{y})$ is a stationary distribution of this kernel. This kernel alone is not ergodic either but when the two kernels are combined we create an ergodic Markov chain with the desired stationary distribution. In general, it takes two sequential applications of the above kernels before the probability density of transitioning between two arbitrary augmented substitution histories becomes nonzero; see Besag et al. [1995] for further discussion of Markov chain visiting schedules.

Sampling States from $p(\mathcal{V}|\mathcal{W},\mathbf{y})$

Our strategy for sampling states $\mathcal{V}$ is to make a draw from the full conditional of internal node states and then to sample the states along each branch conditional on the branch's parent and child nodes. It is useful to remember that when conditioning on the number of virtual and real jumps, and locations of these jumps on the tree, our data generating process becomes a DTMC with transition probability matrix $\mathbf{B}$ and a known number of transitions on each branch. Alternatively, we can think of the trait evolving along each branch i according a branch-specific DTMC with transition probability matrix $\mathbf{B}^{m_i}$, where m_i is the number of transitions on branch i. This is similar to representing a regular, non-uniformized, CTMC model as a collection of branch-specific DTMCs with transition probability matrices $\mathbf{E}(\beta_1),\ldots,\mathbf{E}(\beta_{2n-2})$. This analogy allows us to use standard algorithms for sampling internal node states on a phylogenetic tree by replacing in these algorithms $\mathbf{E}(\beta_i)$ with $\mathbf{B}^{m_i}$ for all $i = 1,\ldots,2n-2$. For completeness, we make this substitution explicit below.

As explained in Chapter 2 we start by using Felsenstein's algorithm to compute a partial likelihood matrix $\mathbf{L}$. When working with augmented substitution histories we change the construction of the branch specific transition probability matrices. Before, our transition probability matrix depended on the length of the branch, $\mathbf{E}(\beta_i) \equiv e^{\mathbf{Q}\beta_i}$. With augmented

substitution histories our transition probability matrix depends on the number of transitions on the branch, $\mathbf{E}(m_i) \equiv \mathbf{B}^{m_i}$. With our new transition probability matrices we calculate the matrix parital likelihoods as described in Chapter 2. The probability of observing tip states below node c conditional on node p being in state h is

$$g_{pch} = \sum_{k=1}^{s} \Pr(y_c = k | y_p = h) l_{ck} = (\mathbf{e}(m_i)_{h-}) \, (\mathbf{l}_{c-})^T,$$

where $\mathbf{l}_{c-} = (l_{c1}, \ldots, l_{cs})$. Element wise multiplication of two vectors yields the vector of partial likelihoods for node p, $(\mathbf{l}_{p-})^T = \mathbf{g}_{pc-} * \mathbf{g}_{pd-}$.

Sampling internal node states We follow the procedure for sampling internal node states as described in Chapter 2. Again, we substitute $\mathbf{E}(m_i)$ for $\mathbf{E}(\beta_i)$. This does not change how we sample the state of the root node. The probability that node c is in state k given node p is in state h and given the state of the tips below node c is
$\Pr(y_c = k | y_p = h \;\&\; \mathbf{d}_c) = \frac{\mathrm{e}(m_i)_{hk} \mathrm{l}_{ck}}{\mathbf{e}(m_i)_{h-} (\mathbf{l}_{c-})^T}$.

Sampling branch states We sample the states on each branch separately, conditioning both on previously sampled internal nodes states and on the number of transitions on each branch. Conditioning on the internal node states means the starting and ending state of each branch are set so we only sample internal segments of the branches. Conditioning on the number of transitions on a branch means we are sampling states of the discrete time Markov chain with transition matrix $\mathbf{B}$. Suppose branch i starts in state v_{i0} and ends in state v_{im_i} (or y_c). We sample each segment of the branch in turn, starting with the second segment because the first segment has to be in the same state as the parent node of the branch. The state of each segment is sampled conditional on the state of the previous segment, the number of transitions until the end of the branch, and the ending state of the branch, $v_{im_i} = y_c$. The state of the d^{th} segement is sampled from a multinomial distribution with probabilities

calculated according to the following formula:

$$\Pr(v_{id} = k | v_{i(d-1)} = h, v_{im_i} = y_c) = \frac{\Pr(v_{i(d-1)} = h, v_{id} = k, v_{im_i} = y_c)}{\Pr(v_{i(d-1)} = h, v_{im_i} = y_c)}$$
$$= \frac{\Pr(v_{i(d-1)} = h)\Pr(v_{id} = k | v_{i(d-1)} = h)\Pr(v_{im_i} = y_c | v_{id} = k)}{\Pr(v_{i(d-1)} = h)\Pr(v_{im_i} = y_c | v_{i(d-1)} = h)} = \frac{b_{hk}\mathrm{e}(m_i - d)_{ky_c}}{\mathrm{e}(m_i - d + 1)_{hy_c}}.$$

After sampling the states along each branch we have completed one cycle through the first Markov kernel by sampling from $\mathrm{p}(\mathcal{V}|\mathcal{W}, \mathbf{y})$. The second Markov kernel requires us to sample virtual transitions conditional on the current substitution history (not augmented by virtual jumps).

Sampling Virtual Jumps from $p(\mathcal{U}|\mathcal{S}, \mathcal{T}, \mathbf{y})$

After sampling the states on each branch, $\mathcal{V}$, we resample virtual jumps, $\mathcal{U}$, on each branch separately. Without loss of generality consider a branch with a newly sampled substitution history, $(\mathbf{s}, \mathbf{t})$, which is the augmented substitution history with all the virtual jumps removed. Suppose the branch contains n real transitions. Resampling virtual jumps for the branch involves resampling virtual jumps for each of the $n + 1$ segments of the branch separately. To sample the a^{th} segment of the branch we need to sample the number of virtual jumps, μ_a, and the locations of these virtual jumps. After sampling virtual jumps for each of the $n + 1$ branch segments we have m transitions total, both real and virtual, so that $m = n + \sum_{a=0}^{n} \mu_a$. Examination of the likelihood of the dominating homogenous Poisson process for a single branch of the tree allows us to derive the distribution of virtual jumps conditional on the substitution history of the branch.

Suppose there are m jumps along a branch including real transitions and virtual transitions. Let v_d be the state of the chain after the d^{th} transition and let π'_{v_0} be the probability that the branch starts in state v_0. The density of the augmented substitution history is

$$\mathrm{p}(\mathbf{v}, \mathbf{w}) = \pi'_{v_0} \frac{e^{-\Omega t}(\Omega t)^m}{m!} \frac{m!}{t^m} \prod_{d=1}^{m} B_{v_{d-1}, v_d}. \tag{3.1}$$

The density as written above has four parts, the probability of starting in state $v_0 = s_0$, the probability of m transition points, the density of the locations of m unordered points

conditional on there being m points, and the probability of each transition in a discrete time Markov chain with transition matrix $\mathbf{B}$.

The density of the augmented substitution history of one branch, $\mathrm{p}(\mathbf{v}, \mathbf{w})$, can be rewritten as $\mathrm{p}(\mathbf{u}, \mathbf{s}, \mathbf{t})$, because the substitution history, $(\mathbf{s}, \mathbf{t})$ combined with the virtual jump locations, $\mathbf{u}$, form the augmented substitution history. To derive the full conditional for $\mathbf{u}$, we follow Rao and Teh [2011] and rewrite density (3.1) as follows:

$$
\begin{aligned}
\mathrm{p}(\mathbf{u}, \mathbf{s}, \mathbf{t}) &= \mathrm{p}(\mathbf{v}, \mathbf{w}) \pi'_{v_0} \frac{e^{-\Omega t}(\Omega t)^m}{m!} \frac{m!}{t^m} \prod_{d=1}^{m} b_{v_{d-1}, v_d} \\
&= \pi'_{s_0} e^{-\Omega t} \Omega^m \prod_{d=1}^{m} b_{v_{d-1} v_d}, \\
&= \pi'_{s_0} e^{-\Omega t} \Omega^{n+\sum_{a=0}^{n} \mu_a} \prod_{a=0}^{n} \left(1 + \frac{q_{s_a s_a}}{\Omega}\right)^{\mu_a} \prod_{z=1}^{n} \frac{q_{s_{z-1} s_z}}{\Omega}, \\
&= \pi'_{s_0} e^{-\Omega t} \prod_{a=0}^{n} \left(\Omega^{\mu_a} \left(1 + \frac{q_{s_a s_a}}{\Omega}\right)^{\mu_a}\right) \prod_{z=1}^{n} \left(q_{s_{z-1} s_z}\right), \\
&= \prod_{a=0}^{n} \left((\Omega + q_{s_a})^{\mu_a} e^{-(\Omega + q_{s_a}) t_a}\right) \pi'_{s_0} \prod_{z=1}^{n} \left(q_{s_{z-1} s_z}\right) \exp\left(\int_0^t q_{X(t)} dt\right), \\
&= \prod_{a=0}^{n} \left(r_a^{\mu_a} e^{-r_a t_a}\right) \pi'_{s_0} \prod_{z=1}^{n} \left(q_{s_{z-1} s_z}\right) \exp\left(\int_0^t q_{X(t)} dt\right), \\
&= \prod_{a=0}^{n} \left(r_a^{\mu_a} e^{-r_a t_a}\right) \pi'_{s_0} \left(\prod_{z=1}^{n} |q_{s_{z-1}}| e^{q_{s_{z-1}} t_{z-1}} \frac{q_{s_{z-1} s_z}}{|q_{s_{z-1}}|}\right) e^{q_{s_n} t_n} \\
&= \prod_{a=0}^{n} \left(r_a^{\mu_a} e^{-r_a t_a}\right) \mathrm{p}(\mathbf{s}, \mathbf{t}),
\end{aligned}
$$

where $q_{s_a} \equiv q_{s_a s_a}$ and $r_a = \Omega + q_{s_a}$. Therefore,

$$
\mathrm{p}(\mathbf{u} | \mathbf{s}, \mathbf{t}, \mathbf{y}) = \mathrm{p}(\mathbf{u} | \mathbf{s}, \mathbf{t}) = \frac{\mathrm{p}(\mathbf{u}, \mathbf{s}, \mathbf{t})}{\mathrm{p}(\mathbf{s}, \mathbf{t})} = \prod_{a=0}^{n} \left(r_a^{\mu_a} e^{-r_a t_a}\right) = \prod_{a=0}^{n} \frac{e^{-r_a t_a} (r_a t_a)^{\mu_a}}{\mu_a!} \frac{\mu_a!}{t_a^{\mu_a}}. \tag{3.2}
$$

The full conditional density (3.2) is a density of an inhomogenous Poisson process with intensity $r(t) = \Omega + q_{X(t)}$. This intensity is piecewise constant so we can add self transition locations/times to a branch segment in state s_a by drawing a realization of a homogenous

Poisson process with rate $r_a = \Omega + q_{s_a}$. More specifically, we sample the number of self transitions, μ_a, on this segment by sampling from a Poisson distribution with mean $r_a t_a$ and then distributing the locations/times of the μ_a self transitions uniformly at random across the segment. This procedure is repeated independently for all segments on all branches of the phylogenetic tree, concluding our MCMC development, summarized in Algorithm 1 and illustrated in Figure 3.2. The numbering in the algorithm is a little strange because inside the procedure we sample self transitions conditional on a substitution history that does not correspond to the substitution history at the beginning or the end of a single iteration. These 'internal' substitution histories are indexed with odd subscripts.

Algorithm 1 MCMC for phylogenetic stochastic mapping

1: Start with an augmented substitution history, $(\mathcal{V}_0, \mathcal{W}_0)$
2: **for** $\gamma \in \{1, 3, 5, \ldots, 2N-1\}$ **do**
3: sample from $\mathrm{p}(\mathcal{V}_\gamma | \mathcal{W}_{\gamma-1}, \mathbf{y})$ producing a new substitution history $(\mathcal{S}_\gamma, \mathcal{T}_\gamma)$

(i) sample internal node states conditional on $\mathbf{y}$ and the number of jumps on each branch

(a) starting at the tips work up the tree calculating partial likelihoods

(b) starting at the root work down the tree sampling internal node states

(ii) sample segmental states conditional on end states and number of jumps

4: sample from $\mathrm{p}(\mathcal{U}_{\gamma+1} | \mathcal{S}_\gamma, \mathcal{T}_\gamma)$, producing $(\mathcal{V}_{\gamma+1}, \mathcal{W}_{\gamma+1})$

(i) remove virtual jumps

(ii) sample virtual jumps conditional on substitution history

5: **end for**
6: **return** $(\mathcal{V}_0, \mathcal{W}_0), (\mathcal{V}_2, \mathcal{W}_2), \ldots, (\mathcal{V}_{2N}, \mathcal{W}_{2N})$

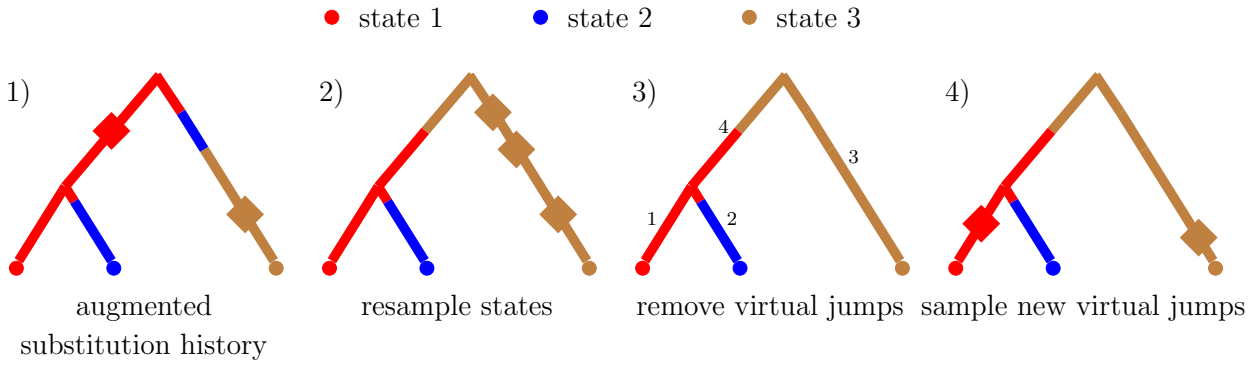

Figure 3.2: An example of applying the two Markov kernels of our MCMC sampler to an augmented substitution history. The diamonds represent virtual transitions. 1) shows an initial augmented substitution history; 2) shows the substitution history after resampling states on the phylogeny conditional on tip node states and the transition points (both real and virtual); 3) shows the substitution history seen in 2) with no virtual jumps; 4) shows the augmented substitution history after resampling virtual jumps conditional on the substitution history seen in 3). The transition from 1) to 2) shows the effect of the first Markov kernel, sampling from $\mathrm{p}(\mathcal{V}|\mathcal{W}, \mathbf{y})$. The transition from 3) to 4) shows the effect of the second Markov kernel, sampling from $\mathrm{p}(\mathcal{U}|\mathcal{S}, \mathcal{T})$.

3.2 Assessing Computational Efficiency

3.2.1 Algorithm Complexity

State-of-the-art stochastic mapping approaches rely on exponentiating CTMC rate matrices, requiring $\mathcal{O}(s^3)$ operations. Our MCMC algorithm uses only matrix-by-vector multiplications, allowing us to accomplish the same task in $\mathcal{O}(s^2)$ operations. Moreover, if the CTMC rate matrix is sparse, the algorithmic complexity of our method can go down further. For example, if $\mathbf{Q}$ is a tri-diagonal matrix, as in the birth-death CTMCs used to model evolution of gene family sizes [Spencer et al., 2006], then our MCMC achieves an algorithmic complexity of $\mathcal{O}(s)$. In contrast, even after disregarding the cost of matrix exponentiation, approaches relying on this operation require at least $\mathcal{O}(s^2)$ operations, because $e^{\mathbf{Q}t}$ is a dense matrix regardless of the sparsity of $\mathbf{Q}$. However, since the number of matrix-by-vector multiplications is a random variable in our algorithm, the algorithmic complexity with respect to the state

space size does not tell the whole story, prompting us to perform an empirical comparison of the two approaches in a set of simulation studies. In these simulation studies, we need to compare state-of-the-art Monte Carlo algorithms and our MCMC in a principled way, which we describe in the next subsection.

3.2.2 Effective Sample Size

When comparing timing results of our MCMC approach and a matrix exponentiation approach, we need to account for the fact that our MCMC algorithm produces correlated substitution histories. One standard way to compare computational efficiency of MCMC algorithms is by reporting CPU time divided by effective sample size (ESS), where ESS is a measure of the autocorrelation in a stationary time series [Holmes and Held, 2006, Girolami and Calderhead, 2011]. More formally, the ESS of a stationary time series of size N with stationary distribution ν is an integer N_{eff} such that N_{eff} independent realizations from ν have the same sample variance as the sample variance of the time series. The ESS of a stationary time series of size N is generally less than N and is equal to N if the time series consists of independent draws from ν.

In MCMC literature, ESSs are usually calculated for model parameters, latent variables, and the log-likelihood. Since we are fixing model parameters in this section, we monitor ESSs for our latent variables — augmented substitution history summaries — and $\log \mathrm{p}(\mathcal{S}, \mathcal{T})$ — the log-density of the substitution history. Although the amount of time spent in each state over the entire tree and the numbers of transitions between each possible pair of states are sufficient statistics of a fully observed CTMC [Guttorp, 1995], it is impractical to use all of these summaries for ESS calculations. This stems from the fact that we are interested in the parameter regimes under which we expect a small number of CTMC transitions over the entire tree. In such regimes, some of the states are never visited so the amount of time spent in these states is zero, which creates an impression that the MCMC is mixing poorly. To avoid this problem, we restrict our attention to the amount of time spent (over the entire tree) in each of the states that are *observed* at the tips. Similarly, we restrict our

attention to transition counts between *observed* tip states. Each of the univariate statistics, including the log-density of the substitution history, yields a potentially different ESS, which we calculate with the help of the R package `coda` [Plummer et al., 2006]. We follow Girolami and Calderhead [2011] and conservatively use the minimum of these univariate ESSs to normalize the CPU time of running our MCMC sampler. More specifically, in all our numerical experiments, we generate 10,000 substitution histories via both MCMC and matrix exponentiation methods and then multiply the CPU time of our MCMC sampler by $10,000/\min(\text{univariate ESSs})$.

3.2.3 Matrix Exponentiation

In all our simulations we compare timing results of our new MCMC approach with another CTMC uniformization approach that relies on matrix exponentiation [Lartillot, 2006]. For the matrix exponentiation approach we recalculate the partial likelihood matrix at each iteration, which involves re-exponentiating the rate matrix. We do so in order to learn how our MCMC method will compare to the matrix exponentiation method in situations where the parameters of the rate matrix are updated during a MCMC that targets the joint posterior of substitution histories and CTMC parameters [Lartillot, 2006, Rodrigue et al., 2008b]. Since matrix exponentiation is a potentially unstable operation [Moler and Van Loan, 1978], we do not repeat it at each iteration in our simulations. Instead, we pre-compute an eigen decomposition of the CTMC rate matrix once, cache this decomposition and then use it to exponentiate $\mathbf{Q}$ at each iteration, a process that only works when $\mathbf{Q}$ is fixed. Even though exponentiating $\mathbf{Q}$ using its pre-computed eigen decomposition is an $\mathcal{O}(s^3)$ operation, our simulations do not fully mimic a more realistic procedure that repeatedly re-exponentiates the rate matrix. Skipping the eigen decomposition operation at each iteration of stochastic mapping increases computational efficiency of the matrix exponentiation method, making our timing comparisons conservative.

In one of our simulation studies, when we consider the effect of sparsity in the rate matrix, we depart from this matrix exponentiation regime. Instead of exponentiating the

rate matrix at each iteration we exponentiate the rate matrix $\mathbf{Q}$ and compute the partial likelihood matrix once, sampling substitution histories at each iteration without recalculating branch-specific transition probabilities or partial likelihoods. We refer to this method as "EXP once." We do not believe that our MCMC method is the most appropriate in this regime, but we are interested in how our new method compares to state-of-the-art methods when the calculations requiring $\mathcal{O}(s^3)$ operations were not involved.

3.2.4 Implementation

We have implemented our new MCMC approach in an R package `phylomap`, available at `https://github.com/vnminin/phylomap`. The package also contains our implementation of the matrix exponentiation-based uniformization method of Lartillot [2006]. We reused as much code as possible between these two stochastic mapping methods in order to minimize the impact of implementation on our time comparison results. We coded all computationally intensive parts in C++ with the help of the `Rcpp` package [Eddelbuettel and François, 2011]. We used the `RcppArmadillo` package to perform sparse matrix calculations [Eddelbuettel and Sanderson, 2014].

3.3 Numerical Experiments

3.3.1 General Set Up

We started all of our simulations by creating a random tree with 50 or 100 tips using the `diversitree` R package [FitzJohn, 2012]. For each simulation that required the construction of a rate matrix, we set the transition rates between all state pairs to be identical. We then scaled the rate matrix for each tree so that the number of expected CTMC transitions per tree was either 2 or 6. These two values were intended to mimic slow and fast rates of evolution. Six expected transitions in molecular evolution settings is usually considered unreasonably high but six transitions (or more) is reasonable in other settings like phylogeography. For example, investigations of Lemey et al. [2009] into the geographical spread of human influenza

H5N1 found on the order of 40 CTMC transitions on their phylogenies. To obtain each set of trait data we simulated one full state history after creating a tree and a rate matrix. We used this full state history as the starting augmented substitution history for our MCMC algorithm.

To ensure our implementation of the matrix exponentiation approach properly sampled from $p(\mathcal{S}, \mathcal{T}|\mathbf{y})$ and to ensure the stationary distribution of our MCMC approach was $p(\mathcal{V}, \mathcal{W}|\mathbf{y})$, we compared distributions of univariate statistics produced by our implementations to the same distributions obtained by using `diversitree`'s implementation of phylogenetic stochastic mapping. We found that all implementations, including `diversitree`'s, appeared to produce the same distributions. Boxplots and histograms showing the results of our investigations can be found in Appendix A.1.

3.3.2 Effect of State Space Size

Our MCMC method scales more efficiently with the size of the state space than matrix exponentiation methods so we were first interested in comparing running times of the two approaches as the size of the CTMC state space increased. In Figure 3.3, we show the amount of time it took the matrix exponentiation method to obtain 10,000 samples for different state space sizes and we show the amount of time it took our MCMC method to obtain an ESS of 10,000 for different state space sizes. The size of the state space varied between 4 states and 60 states. The tuning parameter, Ω, was set to 0.2, ranging between 15 and 103 times larger than the largest rate of leaving a state.

Figure 3.3 contains timing results for four different scenarios. We considered two different rates of evolution corresponding to 2 expected transitions per tree and 6 expected transitions per tree and we considered two different tree tip counts, 50 and 100. The MCMC approach started to run faster than the matrix exponentiation approach when the size of the state space entered the 25 to 35 state range. At 60 states the MCMC approach was clearly faster in all four scenarios. For the senario involving 100 tips, 2 expected transtions, and 60 states the MCMC method was almost 3 times faster than the matrix exponentiation approach. For

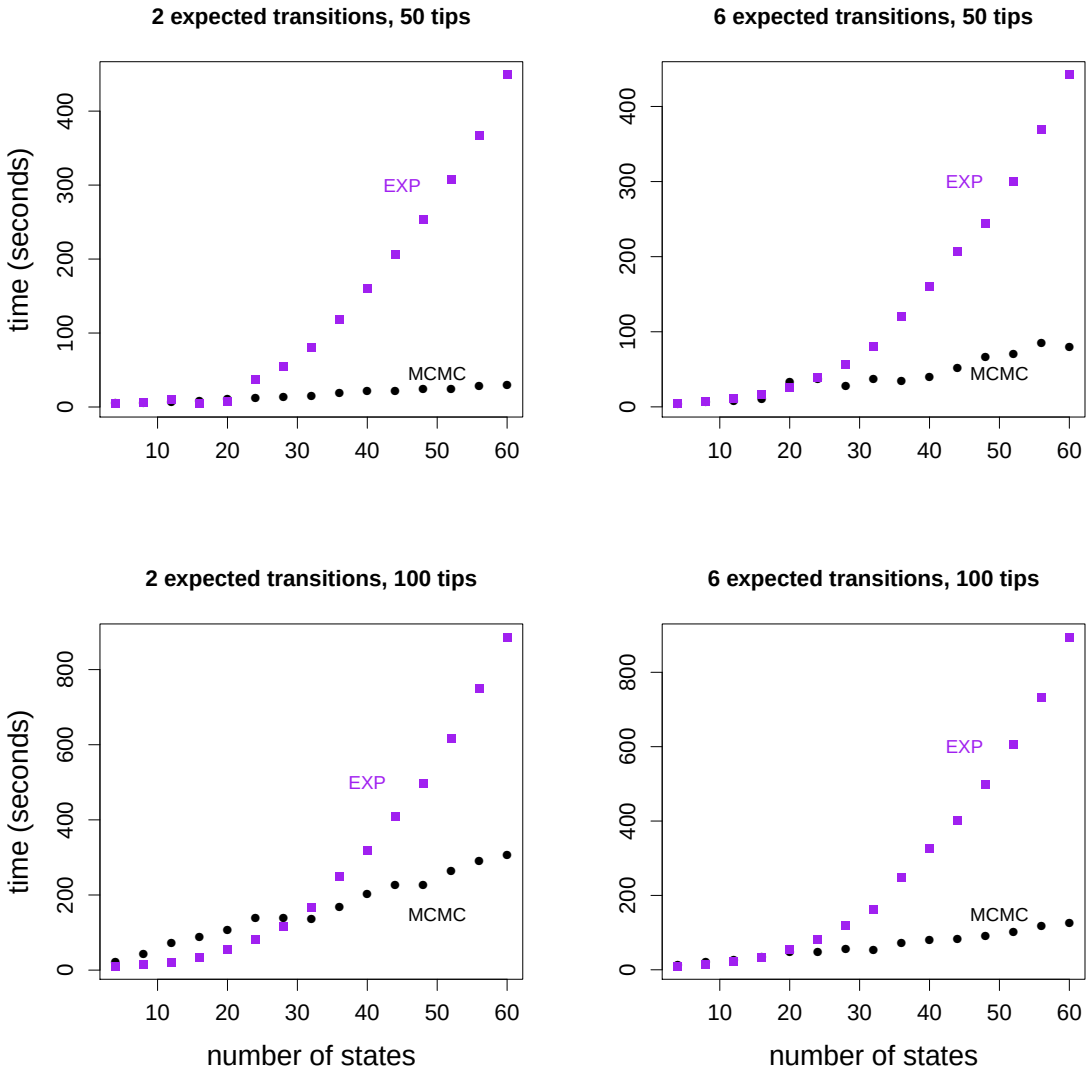

Figure 3.3: State space effect. All four plots show the amount of time required to obtain 10,000 effective samples as a function of the size of the state space for two methods, matrix exponentiation in purple squares and our MCMC sampler in black circles. The two plots in the top row show results for a randomly generated tree with 50 tips. The two plots in the bottom row show results for a randomly generated tree with 100 tips. The two plots in the left column show results for a rate matrix that was scaled to produce 2 expected transitions while the two plots in the right column show results for a rate matrix that was scaled to produce 6 expected transitions.

the scenario involving 50 tips, 2 expected transtions, and 60 states the MCMC method was about 15 times faster than the matrix exponentiation approach.

Our MCMC approach scales well beyond state spaces of size 60 though matrix exponentiation does not. Timing results for our MCMC approach at larger state space sizes can be found in Appendix A.4.

3.3.3 Effect of the Dominating Poisson Process Rate

Our tuning parameter, the dominating Poisson process rate Ω, balances speed against mixing for our MCMC approach. The larger Ω is the slower the MCMC runs and the better it mixes. The optimal value for Ω depends on the CTMC state space and on the entries of the CTMC rate matrix. In our experience, it is not difficult to find a reasonable value for Ω for a fixed tree and a fixed rate matrix by trying different Ω values. We show the results of this exploration in Figure 3.4 for two different values of the state space size, 4 and 60, and for two different trees, with 50 and 100 tips.

The top left plot in Figure 3.4 shows the balance between speed and mixing most clearly. The optimal value for Ω appears to be around 0.2 for 4 states and 50 tips. Our MCMC approach is clearly faster than the matrix exponentiation approach for a wide range of Ω values when the size of the state space is 60. When the size of the state space is 4 the matrix exponentiation approach can be faster, which is not surprising given the small size of the state space. Matrix exponentiation is about two times faster than our MCMC approach for the 100 tip tree with 4 states. Our MCMC approach can yield comparable speeds to the matrix exponentiation approach for the 50 tip tree with 4 states.

3.3.4 Effect of Sparsity

Unlike matrix exponentiation methods, our new MCMC sampler is able to take advantage of sparsity in the CTMC rate matrix. There are three steps in our algorithm that can take advantage of sparsity: computing the partial likelihood matrix, sampling internal node states, and resampling branch states. In all three situations we need to multiply $\mathbf{B}^M$ by

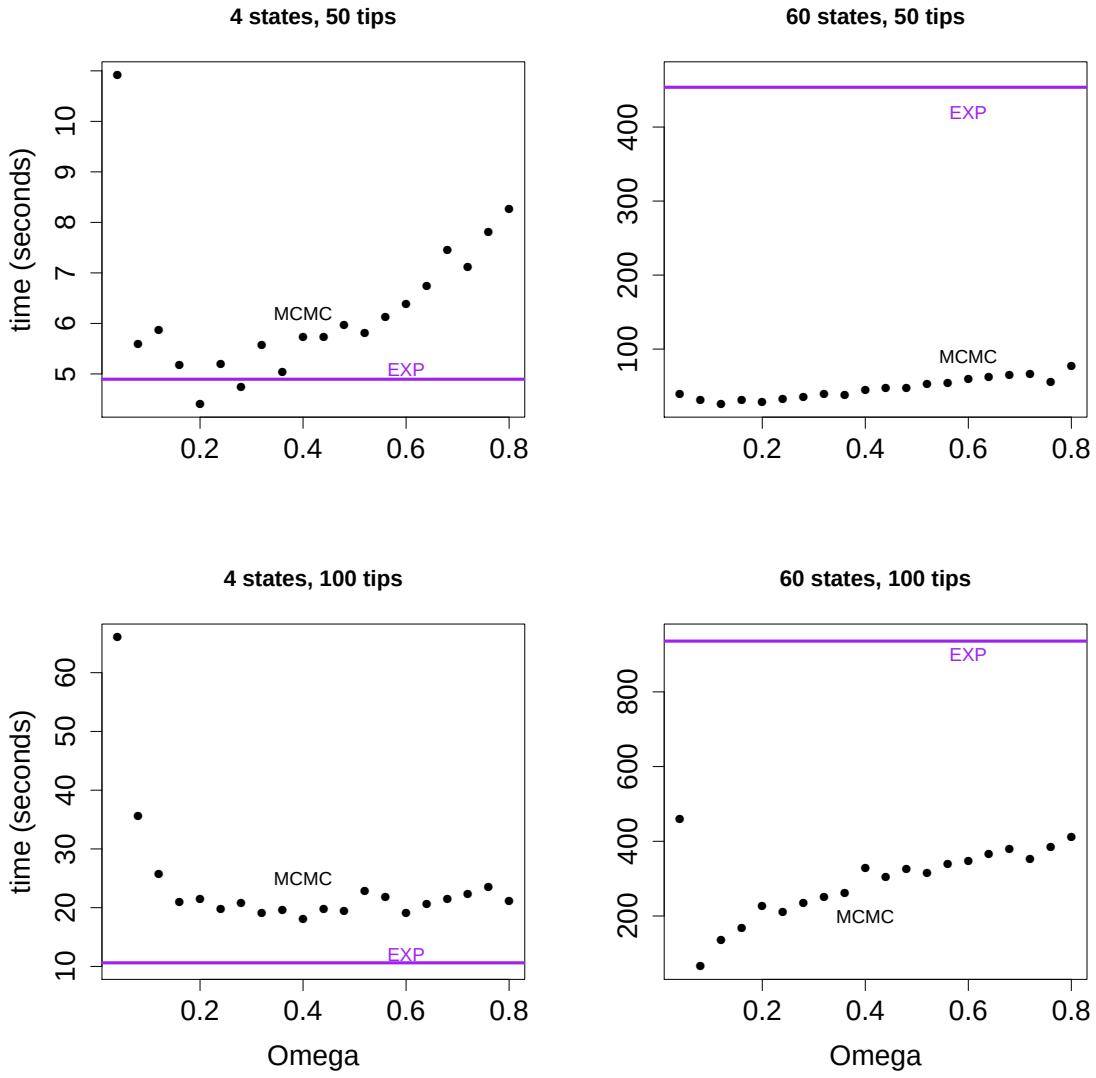

Figure 3.4: Time to obtain 10,000 effective samples as a function of the dominating Poisson process rate, Ω. All four plots show results of our MCMC sampler in black. Timing results for the matrix exponentiation method are represented by a purple horizontal line because the matrix exponentiation result does not vary as a function of Ω. The two plots in the top row show results for a randomly generated tree with 50 tips. The two plots in the bottom row show results for a randomly generated tree with 100 tips. The rate matrix for the plots in the left column had 4 states. The rate matrix for the plots in the right column had 60 states.

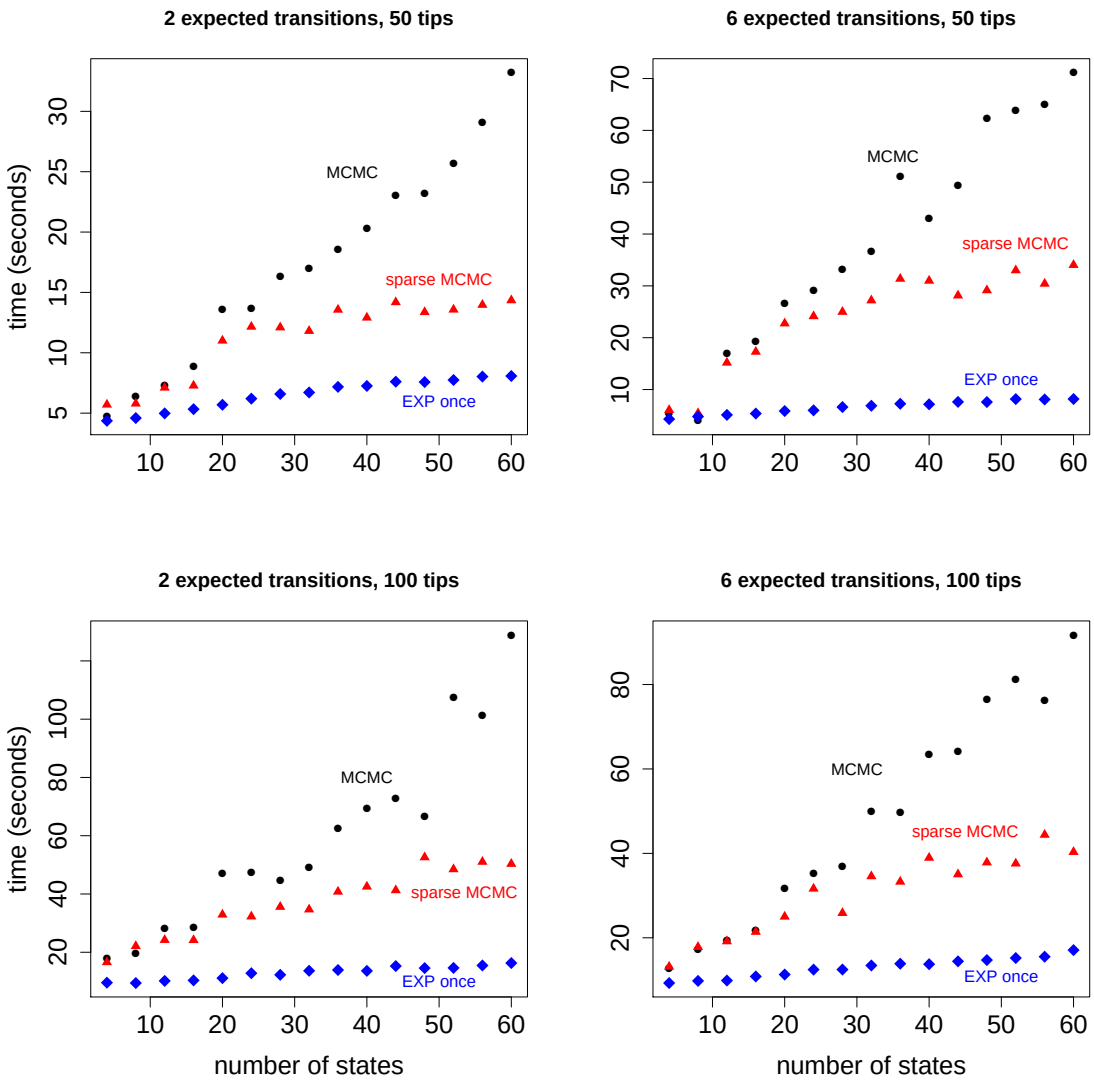

Figure 3.5: Time to obtain 10,000 effective samples as a function of the size of the state space. All four plots show results for three different implementations, our MCMC sampler in black, a sparse version of our MCMC sampler in red, and a matrix exponentiation approach that only exponentiates the rate matrix once per branch in blue. The rate matrix is tridiagonal and scaled to produce 2 expected transitions per tree (in the left column) or 6 expected transitions per tree (in the right column). The two plots in the top row show results for a randomly generated tree with 50 tips. The two plots in the bottom row show results for a randomly generated tree with 100 tips. The dominating Poisson process rate, Ω, is 0.2.

a vector of length s – the size of the state space. For a dense matrix this takes $\mathcal{O}(Ms^2)$ operations. When matrix $\mathbf{B}$ is sparse, the above multiplication requires fewer operations. For example, multiplying a vector by $\mathbf{B}^M$ takes $\mathcal{O}(Ms)$ operations when $\mathbf{B}$ is triadiagonal. It is interesting to note that while matrix exponentiation approaches cannot take advantage of sparsity when creating the partial likelihood matrix they can use sparsity when sampling branches via the uniformization technique of Lartillot [2006].

Speed increases due to sparsity depend on the size of the state space and the degree of sparsity in the probability transition matrix, $\mathbf{B}$. In Figure 3.5 we contrast the sparse implementation of our MCMC method with the implementation that does not take advantage of sparsity. Figure 3.5 also shows timing results for a matrix exponentiation method that only exponentiates the rate matrix once.

For a state space of size 60, the sparse implementation is about 2 times faster than the non-sparse implementation. Exponentiating the rate matrix once was always faster than the sparse implementation, sometimes by a factor of 4. We used uniformization to sample substitution histories for individual branches within the matrix exponentiation algorithm. This portion of the algorithm can take advantage of sparsity but there was not a large overall difference in run times between the sparse and non-sparse implementations.

3.3.5 Models of Protein Evolution

We now turn to the investigation of efficiency of our new phylogenetic stochastic mapping in the context of modeling protein evolution. Evolution of protein coding sequences can be modeled on the following state spaces: state space of 4 DNA bases/nucleotides, state space of 20 amino acids, and state space of 61 codons — nucleotide triplets — excluding the three stop codons. The codon state space is the most computationally demanding of the three, causing existing phylogenetic mapping approaches to slow down considerably. The increased complexity that comes from modeling protein evolution at the codon level enables investigations into selective pressures and makes efficient use of the phylogenetic information for phylogeny reconstruction [Ren et al., 2005].

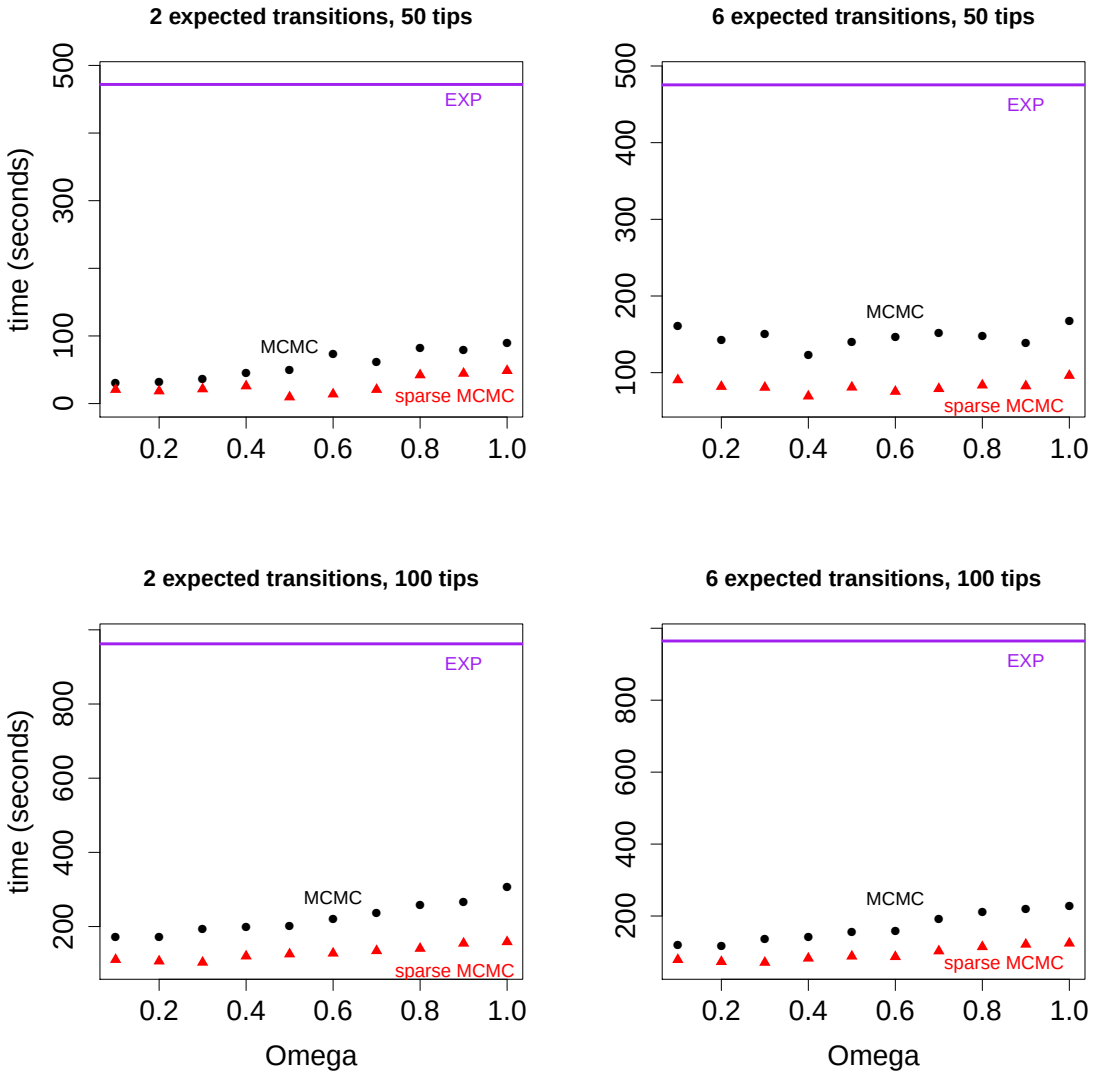

Figure 3.6: Time to obtain 10,000 effective samples as a function of the dominating Poisson process rate, Ω, for the GY94 codon rate matrix. All four plots show results for three different implementations: our MCMC sampler in black, a sparse version of our MCMC sampler in red, and a matrix exponentiation approach in purple. The GY94 rate matrix was scaled to produce 2 expected transitions per tree (in the left column) or 6 expected transitions per tree (in the right column). The two plots in the top row show results for a randomly generated tree with 50 tips. The two plots in the bottom row show results for a randomly generated tree with 100 tips.

In our numerical experiments, we use the Goldman-Yang-94 (GY94) model — a popular codon substitution model proposed by Goldman and Yang [1994], where the rate of substitution between codons depends on whether the substitution is synonymous (the codon codes for the same amino acid before and after the substitution) or nonsynonymous and whether the change is a transition ($A \leftrightarrow G$, $C \leftrightarrow T$) or a transversion. The rate matrix is parameterized by a synonymous/nonsynonymous rate ratio, ω, a transition/transversion ratio, κ, and a stationary distribution of the CTMC, $\boldsymbol{\pi}^c$. The non-diagonal entries of the GY94 rate matrix, as described are

$$q_{ab} = \begin{cases} \omega\kappa\pi_b^c & \text{if } a \to b \text{ is a non-synonymous transition,} \\ \omega\pi_b^c & \text{if } a \to b \text{ is a non-synonymous transversion,} \\ \kappa\pi_b^c & \text{if } a \to b \text{ is a synonymous transition,} \\ \pi_b^c & \text{if } a \to b \text{ is a synonymous transversion,} \\ 0 & \text{if } a \text{ and } b \text{ differ by 2 or 3 nucleotides.} \end{cases}$$

The diagonal rates are determined by the fact that the rows of $\mathbf{Q}$ must sum to zero. In our simulations, we used the default GY94 rate matrix as found in the phylosim R package [Sipos et al., 2011]. The dominating Poisson process rate, our tuning parameter Ω, ranged between being 8 times larger than the largest rate of leaving a state to being 80 times larger. Timing results for the GY94 codon model rate matrix can be found in Figure 3.6. GY94 contains structural zeros allowing our MCMC sampler to take advantage of sparsity and improve running times. Our MCMC approach was about 5 times faster than exponentiating the rate matrix at each iteration. A sparse version of our MCMC approach was about ten times faster than the matrix exponentiation method.

Encouraged by the computational advantage of our method on the codon state space, we also compared our new algorithm and the matrix exponentiation method on the amino acid state space. We used an amino acid substitution model called JTT, proposed by Jones et al. [1992]. The results can be found in Figure A.7 in the Appendix. We found that our MCMC approach is competitive even on the amino acid state space, but does not clearly outperform the matrix exponentiation method. This finding is not surprising in light of the fact that

the size of the amino acid state space is three times smaller than the size of the codon state space.

3.4 Discussion

We have extended the work of Rao and Teh [2011] on continuous time HMMs to phylogenetic stochastic mapping. Our new method avoids matrix exponentiation, an operation that all current state-of-the-art methods rely on. There are two advantages to avoiding matrix exponentiation: 1) matrix exponentiation is computationally expensive for large CTMC state spaces; 2) matrix exponentiation can be numerically unstable. We concentrated on the former advantage, because it is easier to quantify. However, it should be noted that numerical stability of matrix exponentiation is an obstacle faced by all phylogenetic inference methods. Currently, the most popular approach is to employ a reversible CTMC model, whose infinitesimal generator is similar to a symmetric matrix and therefore, can be robustly exponentiated via eigendecomposition [Schabauer et al., 2012]. Researchers typically shy away from non-reversible CTMC models, to a large extent, because of instability of the matrix exponentiation of these models' infinitesimal generators [Lemey et al., 2009]. In our new approach to phylogenetic stochastic mapping, we do not rely on properties of reversible CTMCs, making our method equally attractive for reversible and nonreversible models of evolution.

We believe our new method will be most useful when integrated into a larger MCMC targeting a joint distribution of phylogenetic tree topology, branch lengths, and substitution model parameters. Our optimism stems from the fact that stochastic mapping has already been successfully used in this manner in the context of complex models of protein evolution [Lartillot, 2006, Rodrigue et al., 2008b]. These authors alternate between using stochastic mapping to impute unobserved substitution histories and updating model parameters conditional on these histories. In the next chapter we extend our new algorithm to do exactly this, enabling model parameter updates in the MCMC. Since our MCMC algorithm operates on the state space of augmented substitution histories and model parameters, replacing Monte

Carlo with MCMC in phylogenetic stochastic mapping may have very little impact on the overall MCMC mixing and convergence. A careful study of properties of this new MCMC will be needed to justify this claim.

The computational advances made in [Lartillot, 2006, Rodrigue et al., 2008b] are examples of considerable research activity aimed at speeding up statistical inference under complex models of protein evolution, prompted by the emergence of large amounts of sequence data [Lartillot et al., 2013, Valle et al., 2014]. Challenges encountered in these applications also appear in statistical applications of many other models of evolution that operate on large state spaces: models of microsatellite evolution [Wu and Drummond, 2011], models of gene family size evolution [Spencer et al., 2006], phylogeography models [Lemey et al., 2009], and covarion models [Penny et al., 2001, Galtier, 2001]. Our new phylogenetic stochastic mapping without matrix exponentiation should be a boon for researchers using these models and should enable new analyses that, until now, were too computationally intensive to be attempted.

Chapter 4

HIDDEN RATES MODEL

4.1 Introduction

Bayesian phylogenetic stochastic mapping approximates posterior distributions of the history of trait changes on a phylogeny. Mapping traits onto a phylogeny is an important tool in computational evolutionary biology. Stochastic mapping has been used to improve our understanding of the ancestral reproductive modes of Squamata [Pyron and Burbrink, 2014, King and Lee, 2015], to establish the separate evolution of bacterial photophores multiple times over millions of years among cephalopods [Pankey et al., 2014], and to test hypotheses about morphological trait evolution [Huelsenbeck et al., 2003, Renner et al., 2007]. In some situations, investigators questioned the appropriateness of the CTMC model of evolution [King and Lee, 2015, Skinner, 2010], specifically in regards to the uniform speed of evolutionary changes across the phylogeny. Here, we present a Bayesian phylogenetic stochastic mapping technique that allows for heterogeneity in the rates of evolution across a phylogeny.

Stochastic mapping, as developed by Nielsen [2002], models the evolution of discrete traits of interest with a CTMC. Random evolutionary histories are sampled, conditional on observed data, through the use of a phylogenetic version of the forward filtering-backward sampling algorithm for hidden Markov models (HMMs) [Scott, 2002]. As larger data sets become available our ability to estimate parameters in more complicated CTMC models of evolution increases. Simple evolutionary models are reasonable for small phylogenies but fail to account for heterogeneity in the speed of evolution across larger phylogenies. Covarion [Penny et al., 2001] and covarion-like models of evolution [Galtier, 2001] extend the CTMC model of trait evolution to allow for rate variation across a phylogeny. Extending these ideas, Beaulieu et al. [2013] introduced a hidden rates model in the maximum likelihood

framework to enable ancestral state reconstruction. Beaulieu et al. [2013] implemented the Markov modulated Markov process in an R package, *corHMM*. In this chapter we introduce a Bayesian phylogenetic stochastic mapping method similar to Beaulieu et al. [2013]'s method that allows for different rates of evolution across a phylogeny.

Our primary objection to the maximum likelihood approach concerns uncertainty in the rate matrix parameters. Beaulieu et al. [2013]'s method estimates the most likely rate matrix and reconstructs ancestral states by conditioning on the estimated parameter values. While this approach does yield estimated ancestral states it does not provide a measure of uncertainty in these estimates. Additionally, conditioning on rate matrix parameters before reconstructing ancestral states fails to account for uncertainty in the rate matrix parameters themselves. Our Bayesian approach integrates over rate matrix parameters, yielding a posterior distribution of ancestral state values (and, indeed, entire trait histories). Moreover, we equip our method with an ability to integrate over a set of pre-defined phylogenetic trees.

4.2 Markov Modulated Markov Process

We assume that we have observations of a binary trait measured in a set of species, but we noted that all our methodology can be extended to arbitrary discrete trait observations. Beaulieu et al. [2013] described a hidden rates model of evolution of a binary trait taking values 0 and 1 that, in its simplest form, conjectures two hidden regimes of evolution, fast and slow. In the slow regime the rates of transitioning between states 0 and 1 are relatively low and in the fast regime the rates of transitioning are relatively higher. Our continuous time Markov model of evolution has been replaced with a Markov modulated Markov process model of evolution as illustrated in Figure 4.1.

This hidden rates model corresponds to a four state Markov process: (0, slow), (1, slow), (0, fast), and (1, fast). Beaulieu et al. [2013] parameterized the four state transition matrix with 8 parameters by assuming the Markov chain could transition between states 0 and 1, between the fast and slow regimes, but not between states and regimes simultaneously. The non zero, non diagonal elements of the rate matrix can be visualized by the dots in the

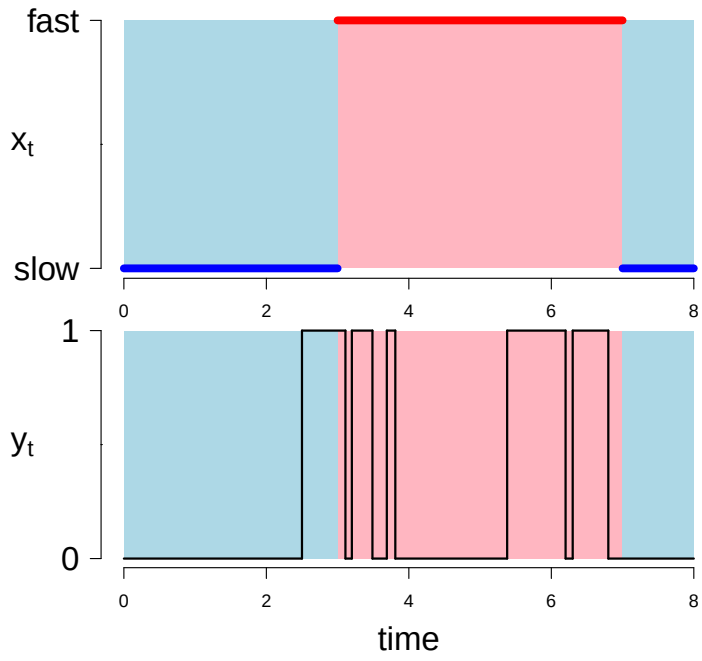

Figure 4.1: Example of a Markov modulated Markov process. Below, y_t, shows the path of a inhomogeneous Markov chain that is modulated by the homogeneous Markov chain above, x_t. The sections shaded in blue represent the slow transition rates and the sections shaded in pink represent the fast transition rates.

following rate matrix,

$$\begin{pmatrix} - & \bullet & \bullet & 0 \\ \bullet & - & 0 & \bullet \\ \bullet & 0 & - & \bullet \\ 0 & \bullet & \bullet & - \end{pmatrix}.$$

This rate matrix produces, in general, a non-reversible Markov chain. Computationally, it is known that exponentiating such matrices, a procedure that appears often in phylogenetic stochastic mapping algorithms, can be difficult [Lemey et al., 2009].

We have constrained the four state transition matrix further in a model with 5 free parameters, λ_{01}, λ_{10}, κ_{01}, κ_{10}, and γ. Our transition rate matrix, $\mathbf{Q}$, is

$$\begin{pmatrix} q_0 & \lambda_{01} & \kappa_{01} & 0 \\ \lambda_{10} & q_1 & 0 & \kappa_{01} \\ \kappa_{10} & 0 & q_2 & \gamma\lambda_{01} \\ 0 & \kappa_{10} & \gamma\lambda_{10} & q_3 \end{pmatrix},$$

where $q_0 = -\lambda_{01} - \kappa_{01}$, $q_1 = -\lambda_{10} - \kappa_{01}$, $q_2 = -\gamma\lambda_{01} - \kappa_{10}$, and $q_3 = -\gamma\lambda_{10} - \kappa_{10}$. The parameters λ_{01} and λ_{10} are the base transition rates between states 0 and 1. The transition rates κ_{01} and κ_{10} govern the rates of transitioning between regimes. Note that the rate of leaving either regime is independent of whether the Markov chain is in state 0 or 1. The two regimes' rates of transitioning between states 0 and 1 only differ by the multiplicative factor, γ.

4.2.1 More Regimes

We can extend the hidden regimes conjecture to include additional hidden regimes, for example slow, medium, and fast. An evolutionary model that uses $(k+1)$ hidden states results in a $2k+2$ state system whose rate matrix is parameterized by $2+3k$ variables and can be represented by the graph in Figure 4.2.

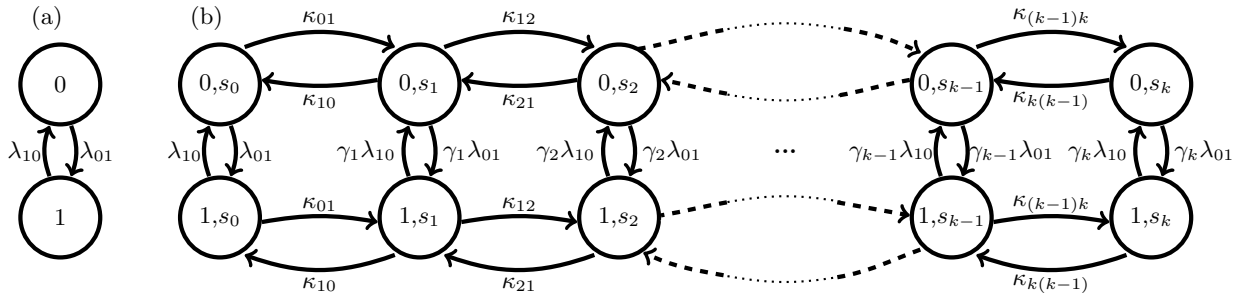

Figure 4.2: a) Two state model of evolution with two parameters, the transition rate between state 0 and state 1, λ_{01}, and the transition rate between state 1 and state 0, λ_{10}. b) $2k+2$ state model of evolution with $2+3k$ parameters.

The graph in Figure 4.2 (b) corresponds to a $(2k+2)$ by $(2k+2)$ rate matrix,

$$\mathbf{Q} = \begin{pmatrix}
q_0 & \gamma_0\lambda_{01} & \kappa_{01} & 0 & & & & & \\
\gamma_0\lambda_{10} & q_1 & 0 & \kappa_{01} & & & & & \\
\kappa_{10} & 0 & q_2 & \gamma_1\lambda_{01} & \kappa_{12} & 0 & & & \\
0 & \kappa_{10} & \gamma_1\lambda_{10} & q_3 & 0 & \kappa_{12} & & & \\
 & & & \ddots & \ddots & \ddots & & & \\
 & & & \kappa_{(k-1)(k-2)} & 0 & q_{2k-2} & \gamma_{k-1}\lambda_{01} & \kappa_{(k-1)k} & 0 \\
 & & & 0 & \kappa_{(k-1)(k-2)} & \gamma_{k-1}\lambda_{10} & q_{2k-1} & 0 & \kappa_{(k-1)k} \\
 & & & & & \kappa_{k(k-1)} & 0 & q_{2k} & \gamma_k\lambda_{01} \\
 & & & & & 0 & \kappa_{k(k-1)} & \gamma_k\lambda_{10} & q_{2k+1}
\end{pmatrix}.$$

The diagonal elements of the rate matrix are selected so that the rows of the rate matrix sum to 0. This means $q_o = -\lambda_{01} - \kappa_{01}$, $q_1 = -\lambda_{10} - \kappa_{01}$, $q_{2k} = -\gamma_k\lambda_{01} - \kappa_{k(k-1)}$, and $q_{2k+1} = -\gamma_k\lambda_{10} - \kappa_{k(k-1)}$. The remaining diagonal elements of the rate matrix are $q_i = -\kappa_{(i/2)(i/2-1)} - \gamma_{i/2}\lambda_{01} - \kappa_{(i/2)(i/2+1)}$ when $i \in [2, 2k-2]$ is even and $q_i = -\kappa_{((i-1)/2)((i-1)/2)-1} - \gamma_{(i-1)/2}\lambda_{10} - \kappa_{((i-1)/2)((i-1)/2+1)}$ when $i \in [2, 2k-2]$ is odd. The value of γ_0 is set to be 1.

As we increase the number of hidden regimes our model becomes more flexible but the number of parameters that needs to be estimated also increases. The amount of data required

to make useful inference using the larger models also needs to incease.

4.3 Stochastic Mapping with Unknown Rate Matrix Parameters

4.3.1 Review of our Phylogenetic Stochastic Mapping Algorithm

A substitution history for a phylogenetic tree is the complete list of transition events (CTMC jumps), including the times of each transition ($\mathcal{T}$) and the types of each transition ($\mathcal{S}$). In chapter 3, we augmented these substitution histories by including viritual transitions. Virtual transitions, or self jumps, are locations on the tree where the trait value 'transitions' from a specified state s to the same state s (e.g, $2 \to 2$). The set of virtual transitions is denoted by $\mathcal{U}$. The introduction of self transitions allows us to create an ergodic Markov chain on the state space of augmented substitution histories whose stationary distribution is $p(\mathcal{V}, \mathcal{W}|\mathbf{y})$, where $\mathcal{W}$ encodes the location of all transitions (both self and real) and $\mathcal{V}$ encodes the type of each transtion. Our algorithm is summarized below.

Algorithm 2 produces a new augmented substitution history

1: start with an augmented substitution history

2: sample $\mathcal{V}|\mathcal{W}, \mathbf{y}$

  (i) sample internal node states

    (a) create partial likelihoods starting at tips

    (b) sample internal node states starting at root

  (ii) sequentially sample states of branch segments

3: sample $\mathcal{U}|\mathcal{S}, \mathcal{T}$

  (i) remove virtual jumps

  (ii) sample virtual jumps

4.3.2 Update Rate Matrix Parameters, in General

In this chapter we introduce a third Markov kernel allowing us to target the posterior distribution, $p(\mathcal{S}, \mathcal{T}, \boldsymbol{\theta}|\boldsymbol{\tau}, \mathbf{y})$, where $\boldsymbol{\tau}$ represents the topology and branch lengths of the phylogeny. The third kernel uses a Metropolis-Hastings procedure to sample a new rate matrix conditional on an augmented substitution history. We wish to sample from $p(\boldsymbol{\theta}|\boldsymbol{\tau}, \mathbf{y}, \mathcal{S}, \mathcal{T})$. At each iteration of the MCMC the Metropolis-Hastings procedure accepts or rejects a newly proposed $\boldsymbol{\theta}$ vector. Let $g(\boldsymbol{\theta}'|\mathcal{V}, \mathcal{W}, \boldsymbol{\theta})$ be the proposal density. Let $p(\boldsymbol{\theta})$ be the prior density of $\boldsymbol{\theta}$. The density of the augmented substitution history conditional on a specified rate matrix is

$$p(\mathcal{V}, \mathcal{W}|\boldsymbol{\theta}) = \pi_{\text{root}} e^{-\Omega t} \prod_{s,h} (\Omega b_{sh})^{n_{sh}},$$

where t is the sum of all the branch lengths and n_{sh} is the number of transitions from state s to state h over the entire tree. After proposing a new vector, $\boldsymbol{\theta}'$, we accept the new vector with probability equal to,

$$\min\left(1, \frac{p(\mathcal{V}, \mathcal{W}|\boldsymbol{\theta}')}{p(\mathcal{V}, \mathcal{W}|\boldsymbol{\theta})} \frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta})} \frac{g(\boldsymbol{\theta}'|\mathcal{V}, \mathcal{W}, \boldsymbol{\theta})}{g(\boldsymbol{\theta}|\mathcal{V}, \mathcal{W}, \boldsymbol{\theta}')}\right),$$

and leave $\boldsymbol{\theta}$ unchanged at the next iteration of the MCMC if we fail to accept the proposal. Our new algorithim is summarized below.

Algorithm 3 produces a new augmented substitution history and rate matrix

---

1: start with an augmented substitution history and rate matrix

2: sample $\mathcal{V}|\mathcal{W}, \mathbf{y}$

3: sample $\mathcal{U}|\mathcal{S}, \mathcal{T}$

4: sample a new rate matrix $\boldsymbol{\theta}'|\mathcal{V}, \mathcal{W}, \boldsymbol{\theta}$

(i) propose new parameters from $g(\boldsymbol{\theta}'|\mathcal{V}, \mathcal{W}, \boldsymbol{\theta})$

(ii) accept/reject the proposed parameters via a Metropolis-Hastings procedure

---

4.3.3 Update Rate Matrix Parameters, for the Hidden Regime Model

We propose updating the $2+3k$ parameters in the hidden regime rate matrix one at a time, so some elements of $\mathbf{Q}$ may be accepted while others are left unchanged. To actually implement our third Markov kernel we need to develop good proposal densities for the parameters of the rate matrix. Lartillot [2006] illustrates a conjugate Gibbs approach for sampling rate matrix parameters conditional on a substitution history. This is close to what we want but subtly different because we want to sample rate matrix parameters conditional on an *augmented* substitution history. Nonetheless, Lartillot [2006]'s conjugate Gibbs approach gives us an excellent proposal density.

The $2+3k$ parameters all have gamma prior distributions with potentially different shape and rate parameters, α and β. A priori, the density of each rate matrix parameter is,

$$\frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}.$$

Subscripts on α and β parameters denote the associated rate matrix parameter. For example, $\alpha_{\lambda_{01}}$ and $\beta_{\lambda_{01}}$ are the shape and rate parameters of the prior on the rate matrix parameter, λ_{01}. Our proposal density for λ_{01} is again a gamma distribution whose parameters depend on the prior parameters $\alpha_{\lambda_{01}}$ and $\beta_{\lambda_{01}}$, the current values of the γ parameters, transition counts between states, n_{sh} and dwell times, t_i. The amount of time spent in state s over the entire phylogeny is t_s, a dwell time. The density of our proposal for λ_{01} given $(\boldsymbol{\theta}, \mathcal{V}, \mathcal{W}, \mathbf{y})$ is $p(\lambda'_{01}|\boldsymbol{\theta}, \mathcal{V}, \mathcal{W}, \mathbf{y})$, which is the gamma distribution,

$$\Gamma\left(\alpha_{\lambda_{01}} + \sum_{i=0}^{k} n_{(2i)(2i+1)}, \beta_{\lambda_{01}} + \sum_{i=0}^{k} \gamma_i t_{2i}\right),$$

where $\left(\alpha_{\lambda_{01}} + \sum_{i=0}^{k} n_{(2i)(2i+1)}\right)$ is the shape parameter and $\left(\beta_{\lambda_{01}} + \sum_{i=0}^{k} \gamma_i t_{2i}\right)$ is the rate parameter. The density of our proposal for λ_{10} is $p(\lambda'_{10}|\boldsymbol{\theta}, \mathcal{V}, \mathcal{W}, \mathbf{y})$, which is the gamma distribution,

$$\Gamma\left(\alpha_{\lambda_{10}} + \sum_{i=0}^{k} n_{(2i+1)(2i)}, \beta_{\lambda_{10}} + \sum_{i=0}^{k} \gamma_i t_{2i+1}\right).$$

The density of our proposals for $\kappa_{j(j+1)}$ is $p(\kappa'_{j(j+1)}|\boldsymbol{\theta}, \mathcal{V}, \mathcal{W}, \mathbf{y})$, which is the gamma distribution,

$$\Gamma\left(\alpha_{\kappa_{j(j+1)}} + n_{(2j)(2j+2)} + n_{(2j+1)(2j+3)}, \beta_{\kappa_{j(j+1)}} + t_{2j} + t_{2j+1}\right),$$

where the index j ranges from 0 to $k-1$. The density of our proposals for $\kappa_{j(j-1)}$ is $p(\kappa'_{j(j-1)}|\boldsymbol{\theta}, \mathcal{V}, \mathcal{W}, \mathbf{y})$, which is the gamma distribution,

$$\Gamma\left(\alpha_{\kappa_{j(j-1)}} + n_{(2j)(2j-2)} + n_{(2j+1)(2j-1)}, \beta_{\kappa_{j(j-1)}} + t_{2j} + t_{2j+1}\right),$$

where the index j ranges from 1 to k. The density of our proposals for γ_r is $p(\gamma'_r|\boldsymbol{\theta}, \mathcal{V}, \mathcal{W}, \mathbf{y})$, which is the gamma distribution,

$$\Gamma\left(\alpha_{\gamma_r} + n_{(2r)(2r+1)} + n_{(2r+1)(2r)}, \beta_{\gamma_r} + \lambda_{01} t_{2r} + \lambda_{10} t_{2r+1}\right),$$

where the index r ranges from 1 to k.

Given the proposal distributions, the prior distributions, and the likelihood of the augmented substitution history we can now step through the Metropolis-Hastings machinery to generate (possibly) new values of $\boldsymbol{\theta}$, yielding our third Markov kernel.

4.3.4 Hastings Ratios for Proposed Rate Matrix Parameters

Sample a new value for λ_{01}, (λ'_{01}), from a gamma distribution, $\lambda'_{01} \sim \text{gamma}(\alpha', \beta')$, where $\alpha' = \alpha_{\lambda_{01}} + \sum_{i=0}^{k} n_{(2i)(2i+1)}$ and $\beta' = \beta_{\lambda_{01}} + \sum_{i=0}^{k} \gamma_i t_{2i}$. The Hastings ratio for this proposal is

$$\frac{\frac{(\beta')^{\alpha'}}{\Gamma(\alpha')}(\lambda_{01})^{\alpha'-1} e^{-\beta'\lambda_{01}}}{\frac{(\beta')^{\alpha'}}{\Gamma(\alpha')}(\lambda'_{01})^{\alpha'-1} e^{-\beta'\lambda'_{01}}} = (\lambda_{01}/\lambda'_{01})^{\alpha'-1} e^{-\beta'(\lambda_{01}-\lambda'_{01})}.$$

Sample a new value for λ_{10}, (λ'_{10}), from a gamma distribution, $\lambda'_{10} \sim \text{gamma}(\alpha', \beta')$, where $\alpha' = \alpha_{\lambda_{10}} + \sum_{i=0}^{k} n_{(2i+1)(2i)}$ and $\beta' = \beta_{\lambda_{10}} + \sum_{i=0}^{k} \gamma_i t_{2i+1}$. The Hastings ratio for this proposal is,

$$(\lambda_{10}/\lambda'_{10})^{\alpha'-1} e^{-\beta'(\lambda_{10}-\lambda'_{10})}.$$

Sample a new value for $\kappa_{j(j+1)}$, $(\kappa'_{j(j+1)})$, from a gamma distribution, $\kappa'_{j(j+1)} \sim \text{gamma}(\alpha', \beta')$, where $\alpha' = \alpha_{\kappa_{j(j+1)}} + n_{(2j)(2j+2)} + n_{(2j+1)(2j+3)}$ and $\beta' = \beta_{\kappa_{j(j_1)}} + t_{2j} + t_{2j+1}$. The Hastings ratio for this proposal is,

$$(\kappa_{j(j+1)}/\kappa'_{j(j+1)})^{\alpha'-1} e^{-\beta'(\kappa_{j(j+1)} - \kappa'_{j(j+1)})}.$$

The index j ranges from 0 to $k-1$.

Sample a new value for $\kappa_{j(j-1)}$, $(\kappa'_{j(j-1)})$, from a gamma distribution, $\kappa'_{j(j-1)} \sim \text{gamma}(\alpha', \beta')$, where $\alpha' = \alpha_{\kappa_{j(j-1)}} + n_{(2j)(2j-2)} + n_{(2j+1)(2j-1)}$ and $\beta' = \beta_{\kappa_{j(j-1)}} + t_{2j} + t_{2j+1}$. The Hastings ratio for this proposal is,

$$(\kappa_{j(j-1)}/\kappa'_{j(j-1)})^{\alpha'-1} e^{-\beta'(\kappa_{(j+1)j} - \kappa'_{(j+1)j})}.$$

The index j ranges from 1 to k.

Sample a new value for γ_r, (γ'_r), from a gamma distribution, $\gamma'_r \sim \text{gamma}(\alpha', \beta')$, where $\alpha' = \alpha_{\gamma_r} + n_{(2r)(2r+1)} + n_{(2r+1)(2r)}$ and $\beta' = \beta_{\gamma_r} + \lambda_{01} t_{2r} + \lambda_{10} t_{2r+1}$. The Hastings ratio for this proposal is,

$$(\gamma_r/\gamma'_r)^{\alpha'-1} e^{-\beta'(\gamma_r - \gamma'_r)}.$$

The index r ranges from 1 to k.

4.3.5 Metropolis Ratios for Proposed Rate Matrix Parameters

Proposing a new value for one of the rate matrix parameters yields a new probability transition matrix, $\mathbf{B}'$. The Metropolis ratio is,

$$\frac{p(\mathcal{V}, \mathcal{W}|\mathbf{Q}')p(\mathbf{Q}')}{p(\mathcal{V}, \mathcal{W}|\mathbf{Q})p(\mathbf{Q})} = \frac{\prod_{i,h}(\Omega b'_{ih})^{n_{ih}}}{\prod_{i,h}(\Omega b_{ih})^{n_{ih}}} \times \frac{p(\mathbf{Q}')}{p(\mathbf{Q})}.$$

When updating λ_{01} the density of the augmented substitution history conditional on the rate matrix parameters is,

$$p(\mathcal{V}, \mathcal{W}|\mathbf{Q}) \propto (\lambda_{01})^{\sum_{i=0}^{k} n_{(2i)(2i+1)}} \prod_{i=0}^{k} (\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda_{01})^{n_{(2i)(2i)}},$$

where $\kappa_{0(-1)}$ and $\kappa_{k(k+1)}$ are both 0. The Metropolis ratio is,

$$\left(\frac{\lambda'_{01}}{\lambda_{01}}\right)^{\sum_{i=0}^{k} n_{(2i)(2i+1)}} \prod_{i=0}^{k} \left(\frac{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda'_{01}}{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda_{01}}\right)^{n_{(2i)(2i)}} \times \left(\frac{\lambda'_{01}}{\lambda_{01}}\right)^{\alpha-1} \frac{e^{-\beta\lambda'_{01}}}{e^{-\beta\lambda_{01}}},$$
$$= \exp\left[-\beta(\lambda'_{01} - \lambda_{01})\right] \left(\frac{\lambda'_{01}}{\lambda_{01}}\right)^{\alpha-1+\sum_{i=0}^{k} n_{(2i)(2i+1)}} \prod_{i=0}^{k} \left(\frac{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda'_{01}}{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda_{01}}\right)^{n_{(2i)(2i)}}.$$

When updating λ_{10} the density of the augmented substitution history conditional on the rate matrix parameters is,

$$p(\mathcal{V}, \mathcal{W}|\mathbf{Q}) \propto (\lambda_{10})^{\sum_{i=0}^{k} n_{(2i+1)(2i)}} \prod_{i=0}^{k} (\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda_{10})^{n_{(2i+1)(2i+1)}},$$

where $\kappa_{0(-1)}$ and $\kappa_{k(k+1)}$ are both 0. The Metropolis ratio is,

$$\left(\frac{\lambda'_{10}}{\lambda_{10}}\right)^{\sum_{i=0}^{k} n_{(2i+1)(2i)}} \prod_{i=0}^{k} \left(\frac{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda'_{10}}{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda_{10}}\right)^{n_{(2i+1)(2i+1)}} \times \left(\frac{\lambda'_{10}}{\lambda_{10}}\right)^{\alpha-1} \frac{e^{-\beta\lambda'_{10}}}{e^{-\beta\lambda_{10}}},$$
$$= \exp\left[-\beta(\lambda'_{10} - \lambda_{10})\right] \left(\frac{\lambda'_{10}}{\lambda_{10}}\right)^{\alpha-1+\sum_{i=0}^{k} n_{(2i+1)(2i)}} \prod_{i=0}^{k} \left(\frac{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda'_{10}}{\Omega - \kappa_{i(i+1)} - \kappa_{i(i-1)} - \gamma_i \lambda_{10}}\right)^{n_{(2i+1)(2i+1)}}.$$

When updating $\kappa_{j(j+1)}$ the density of the augmented substitution history conditional on the rate matrix parameters is,

$$p(\mathcal{V}, \mathcal{W}|\mathbf{Q}) \propto (\kappa_{j(j+1)})^{n_{(2j)(2j+2)}+n_{(2j+1)(2j+3)}} (\Omega - \kappa_{j(j-1)} - \kappa_{j(j+1)} - \gamma_j \lambda_{01})^{n_{(2j)(2j)}}$$
$$\times (\Omega - \kappa_{j(j-1)} - \kappa_{j(j+1)} - \gamma_j \lambda_{10})^{n_{(2j+1)(2j+1)}},$$

where $\kappa_{0(-1)}$ and $\kappa_{k(k+1)}$ are both 0. The Metropolis ratio is,

$$\begin{aligned}
&\left(\frac{\kappa'_{j(j+1)}}{\kappa_{j(j+1)}}\right)^{n_{(2j)(2j+2)}+n_{(2j+1)(2j+3)}} \\
&\quad\times\left(\frac{\Omega-\kappa_{j(j-1)}-\kappa'_{j(j+1)}-\gamma_j\lambda_{01}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}\right)^{n_{(2j)(2j)}}\left(\frac{\Omega-\kappa_{j(j-1)}-\kappa'_{j(j+1)}-\gamma_j\lambda_{10}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}\right)^{n_{(2j+1)(2j+1)}} \\
&\qquad\times\left(\frac{\kappa'_{j(j+1)}}{\kappa_{j(j+1)}}\right)^{\alpha-1}\frac{e^{-\beta\kappa'_{j(j+1)}}}{e^{-\beta\kappa_{j(j+1)}}} \\
&=\exp\left(-\beta(\kappa'_{j(j+1)}-\kappa_{j(j+1)})\right)\left(\frac{\kappa'_{j(j+1)}}{\kappa_{j(j+1)}}\right)^{\alpha-1+n_{(2j)(2j+2)}+n_{(2j+1)(2j+3)}} \\
&\quad\times\left(\frac{\Omega-\kappa_{j(j-1)}-\kappa'_{j(j+1)}-\gamma_j\lambda_{01}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}\right)^{n_{(2j)(2j)}}\left(\frac{\Omega-\kappa_{j(j-1)}-\kappa'_{j(j+1)}-\gamma_j\lambda_{10}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}\right)^{n_{(2j+1)(2j+1)}}.
\end{aligned}$$

When updating $\kappa_{j(j-1)}$ the density of the augmented substitution history conditional on the rate matrix parameters is,

$$\begin{aligned}
p(\mathcal{V},\mathcal{W}|\mathbf{Q}) &\propto \left(\kappa_{j(j-1)}\right)^{n_{(2j)(2j-2)}+n_{(2j+1)(2j-1)}} \\
&\quad\times\left(\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}\right)^{n_{(2j)(2j)}}\left(\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}\right)^{n_{(2j+1)(2j+1)}},
\end{aligned}$$

where $\kappa_{0(-1)}$ and $\kappa_{k(k+1)}$ are both 0. The Metropolis ratio is,

$$\begin{aligned}
&\left(\frac{\kappa'_{j(j-1)}}{\kappa_{j(j-1)}}\right)^{n_{(2j)(2j-2)}+n_{(2j+1)(2j-1)}} \\
&\quad\times\left(\frac{\Omega-\kappa'_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}\right)^{n_{(2j)(2j)}}\left(\frac{\Omega-\kappa'_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}\right)^{n_{(2j+1)(2j+1)}} \\
&\qquad\times\left(\frac{\kappa'_{j(j-1)}}{\kappa_{j(j-1)}}\right)^{\alpha-1}\frac{e^{-\beta\kappa'_{j(j-1)}}}{e^{-\beta\kappa_{j(j-1)}}} \\
&=\exp\left[-\beta(\kappa'_{j(j-1)}-\kappa_{j(j-1)})\right]\left(\frac{\kappa'_{j(j-1)}}{\kappa_{j(j-1)}}\right)^{\alpha-1+n_{(2j)(2j-2)}+n_{(2j+1)(2j-1)}} \\
&\quad\times\left(\frac{\Omega-\kappa'_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}\right)^{n_{(2j)(2j)}}\left(\frac{\Omega-\kappa'_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}\right)^{n_{(2j+1)(2j+1)}}.
\end{aligned}$$

When updating γ_r the density of the augmented substitution history conditional on the rate matrix parameters is,

$$p(\mathcal{V},\mathcal{W}|\mathbf{Q}) \propto (\gamma_r)^{n_{(2r)(2r+1)}+n_{(2r+1)(2r)}} \times \left(\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}\right)^{n_{(2r)(2r)}} \left(\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{10}\right)^{n_{(2r+1)(2r+1)}},$$

where $\kappa_{0(-1)}$ and $\kappa_{k(k+1)}$ are both 0. The Metropolis ratio is,

$$\begin{aligned}
&\left(\frac{\gamma_r'}{\gamma_r}\right)^{n_{(2r)(2r+1)}+n_{(2r+1)(2r)}} \\
&\quad\times\left(\frac{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r'\lambda_{01}}{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}}\right)^{n_{(2r)(2r)}} \left(\frac{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r'\lambda_{10}}{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}}\right)^{n_{(2r+1)(2r+1)}} \\
&\qquad\times\left(\frac{\gamma_r'}{\gamma_r}\right)^{\alpha-1}\frac{e^{-\beta\gamma_r'}}{e^{-\beta\gamma_r}} \\
&=\exp\left[-\beta(\gamma_r'-\gamma_r)\right]\left(\frac{\gamma_r'}{\gamma_r}\right)^{\alpha-1+n_{(2r)(2r+1)}+n_{(2r+1)(2r)}} \\
&\quad\times\left(\frac{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r'\lambda_{01}}{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}}\right)^{n_{(2r)(2r)}} \left(\frac{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r'\lambda_{10}}{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}}\right)^{n_{(2r+1)(2r+1)}}.
\end{aligned}$$

4.3.6 Metropolis-Hastings Acceptance Probabilities for Proposed Rate Matrix Parameters

After proposing a new value of a λ (or γ) from the specified gamma(α',β') distribution we accept the proposed value with probability,

$$\min\left(\frac{\prod_{k,h}(\Omega b'_{kh})^{n_{kh}}}{\prod_{k,h}(\Omega b_{kh})^{n_{kh}}}\left(\frac{\lambda}{\lambda'}\right)^{\alpha'-\alpha}\frac{e^{-(\beta'-\beta)\lambda}}{e^{-(\beta'-\beta)\lambda'}},1\right),$$

where α' and β' are the α' and β' from the proposal section and the new value is always rejected if it causes the absolute value of a diagonal element of the rate matrix to be larger than Ω.

Acceptance Probability for λ_{01}

After proposing a new value for λ_{01} we reject the new value if any of the new diagonal elements of the rate matrix are larger in absolute value than Ω. If the proposed value is not

yet rejected we accept the proposed value with probability,

$$\min\left[\exp\left((\lambda'_{01}-\lambda_{01})\sum_{i=0}^{k}\gamma_i t_{2i}\right)\prod_{i=0}^{k}\left(\frac{\Omega-\kappa_{i(i+1)}-\kappa_{i(i-1)}-\gamma_i\lambda'_{01}}{\Omega-\kappa_{i(i+1)}-\kappa_{i(i-1)}-\gamma_i\lambda_{01}}\right)^{n_{(2i)(2i)}},1\right].$$

Acceptance Probability for λ_{10}

After proposing a new value for λ_{10} we reject the new value if any of the new diagonal elements of the rate matrix are larger in absolute value than Ω. If the proposed value is not yet rejected we accept the proposed value with probability,

$$\min\left[\exp\left((\lambda'_{10}-\lambda_{10})\sum_{i=0}^{k}\gamma_i t_{2i+1}\right)\prod_{i=0}^{k}\left(\frac{\Omega-\kappa_{i(i+1)}-\kappa_{i(i-1)}-\gamma_i\lambda'_{10}}{\Omega-\kappa_{i(i+1)}-\kappa_{i(i-1)}-\gamma_i\lambda_{10}}\right)^{n_{(2i+1)(2i+1)}},1\right].$$

Acceptance Probability for $\kappa_{j(j+1)}$

After proposing a new value for $\kappa_{j(j+1)}$ we reject the new value if any of the new diagonal elements of the rate matrix are larger in absolute value than Ω. If the proposed value is not yet rejected we accept the proposed value with probability,

$$\min\left[e^{\left[(\kappa'_{j(j+1)}-\kappa_{j(j+1)})(t_{2j}+t_{2j+1})\right]}\times\left(\frac{\Omega-\kappa_{j(j-1)}-\kappa'_{j(j+1)}-\gamma_j\lambda_{01}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}\right)^{n_{(2j)(2j)}}\times\left(\frac{\Omega-\kappa_{j(j-1)}-\kappa'_{j(j+1)}-\gamma_j\lambda_{10}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}\right)^{n_{(2j+1)(2j+1)}},1\right].$$

Acceptance Probability for $\kappa_{j(j-1)}$

After proposing a new value for $\kappa_{j(j-1)}$ we reject the new value if any of the new diagonal elements of the rate matrix are larger in absolute value than Ω. If the proposed value is not yet rejected we accept the proposed value with probability,

$$\min\left[e^{\left[(\kappa'_{j(j-1)}-\kappa_{j(j-1)})(t_{2j}+t_{2j+1})\right]}\times\left(\frac{\Omega-\kappa'_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{01}}\right)^{n_{(2j)(2j)}}\times\left(\frac{\Omega-\kappa'_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}{\Omega-\kappa_{j(j-1)}-\kappa_{j(j+1)}-\gamma_j\lambda_{10}}\right)^{n_{(2j+1)(2j+1)}},1\right].$$

Acceptance Probability for γ_r

After proposing a new value for γ_r we reject the new value if any of the new diagonal elements of the rate matrix are larger in absolute value than Ω. If the proposed value is not yet rejected we accept the proposed value with probability,

$$\min\left[e^{[(\gamma_r'-\gamma_r)(\lambda_{01}t_{2r}+\lambda_{10}t_{2r+1})]}\times\left(\frac{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r'\lambda_{01}}{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}}\right)^{n_{(2r)(2r)}}\times\left(\frac{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r'\lambda_{10}}{\Omega-\kappa_{r(r-1)}-\kappa_{r(r+1)}-\gamma_r\lambda_{01}}\right)^{n_{(2r+1)(2r+1)}},1\right].$$

These acceptance ratios complete the conceptual framework needed to implement our MCMC algorithm for jointly sampling augmented substitution histories and rate matrix parameters.

4.4 Multiple Trees

Conditioning on a single topology with set branch lengths ignores potentially important uncertainty in a reconstructed phylogeny. Often we do not want to analyze our data by conditioning on one specified tree, instead we wish to integrate out our uncertainty in tree space. In these situations we would like to sample from,

$$p(\mathcal{V},\mathcal{W},\boldsymbol{\theta},\boldsymbol{\tau}|\mathbf{y}),$$

where $\boldsymbol{\tau}$ represents both the topology and the branch lengths of a phylogeny and where $\mathbf{y}$ represents both trait data and DNA sequence data. We do not know how to jointly sample augmented substitution histories and topologies. Instead of sampling topologies we present a work-around, averaging over a pre-specified set of phylogenies.

Our algorithm as explained thus far starts with a single phylogeny, an augmented substitution history, and a rate matrix. After one step the algorithm returns a new augmented substitution history and a new rate matrix. To average over a set of phylogenies we instead start with an augmented substitution history for each phylogeny in the pre-specified set of phylogenies, and a single rate matrix. We sample new augmented substitution histories for

each phylogeny separately and then pick one at random. The augmented substitution history of the randomly selected phylogeny is added to the Markov chain of augmented substitution histories and then used to update the rate matrix parameters. This procedure yields a new rate matrix and a new set of augmented subsitution histories for the next iteration.

Constructing a good set of phylogenies is a task that is primarily informed by DNA sequence data (as opposed to non-sequence trait data). One way to construct such a set is to approximate the posterior distribution of phylogenies using MCMC and use the phylogenies accepted during this MCMC to form a set of phylogenies to be used in estimation of hidden rates model parameters. Another way to construct a set of phylogenies is by bootstrapping aligned DNA sequences. Columns of the aligned DNA sequences (where each row corresponds to a species) are sampled with replacement. The new bootstrapped alignment can then be used to produce a phylogeny by using a maximum likelihood estimate. Each bootstrapped alignment can be used to estimate a new phylogeny to be included in our set of phylogenies.

Chapter 5

SQUAMATES AND CEPHALOPODS

We found that accounting for rate matrix heterogeneity via a hidden regimes model dramatically altered conclusions concerning the ancestral reproductive mode of the most recent common ancestor of Squamata. On the other hand, using the hidden regimes model did not significantly change our conclusions about the number of times bioluminesence evolved in cephalopods. In both cases we used our new Bayesian phylogenetic stochastic mapping approach. For the squamate data we conditioned the entire analysis on a single phylogeny and for the cephalopod data we averaged over a set of 1,000 phylogenies based on bootstraps of DNA sequence alignments.

5.1 Squamata

Squamates, of the order Squamata, compose a class of scaled reptiles that includes over 9,400 species of lizards and snakes. Squamates are found on every continent save Antartica and are even found in the Indian and Pacific oceans. These creatures have been the focus of many phylogenetic studies designed to address questions concerning trait evolution [Huey and Bennett, 1987, Losos, 1990]. Phylogenies are crucial to these studies but they are also quite difficult to create and different approaches have resulted in seriously different evolutionary relationships between species [Pyron et al., 2013]. Pyron et al. [2013] created a phylogeny containing 4,161 squamate species using up to 12,896 base pairs per species. This DNA data came from 12 genes, 7 nuclear and 5 mitochondrial, and were gathered from many collaborative projects. Pyron et al. [2013] used RAxML to generate the maximum likelihood tree that we used for our analysis.

The reproductive mode of Squamata species falls into two basic categories, oviparity (egg

laying), and viviparity (live birth). Pyron and Burbrink [2014] assembled a phylogenetic tree with 3,951 squamate species at the tips, see Figure 5.1. Of these 3,951 species, 3,108 are oviparous while 845 are viviparous. The tree from Pyron and Burbrink [2014] has fewer taxa at the tips than the tree from [Pyron et al., 2013] because Pyron and Burbrink [2014]'s tree was pruned to only include species that came with reproductive parity data.

5.1.1 Question of Interest

Ancestral parity mode reconstruction found strong support for an early origin of viviparity in the most recent common ancestor of Squamata [Pyron and Burbrink, 2014]. Pyron and Burbrink [2014] used the framework associated with Maddison et al. [2007]'s binary-state speciation and extinction model of evolution (BiSSE) to reconstruct this ancestral state. Under this model, the rates of speciation depend on the trait state, which allows for modeling of interdependent speciation and trait evolution processes. The result of Pyron and Burbrink [2014] is surprising because the ancestral state of Squamata was generally believed to be oviparous. King and Lee [2015] hypothesized that the surprising result was an artifact arising from the assumption that the transition rates between oviparity and viviparity were constant over the entire phylogeny. King and Lee [2015] considered two different approaches to allow for heterogeneity in transtion rates across the tree. Their first approach divided the phylogeny into clades, allowing oviparity/viviparity transition rates to be estimated separately for each clade. This approach was slightly *ad-hoc* as the clades were selected *a priori*. King and Lee [2015]'s second approach used the hidden rates model of Beaulieu et al. [2013] that allows for different rates of evolution across a phylogeny. King and Lee [2015] investigated models that involved up to 5 rate categories. Beaulieu et al. [2013]'s R package, corHMM, implements a maximum likelihood approach to estimating transition rates and ancestral state reconstructions. We performed a Bayesian phylogenetic mapping analysis that incorporated Beaulieu et al. [2013]'s hidden rates idea to address the question of interest, "was the most recent common ancestor (MRCA) of squamates oviparous or viviparous?"

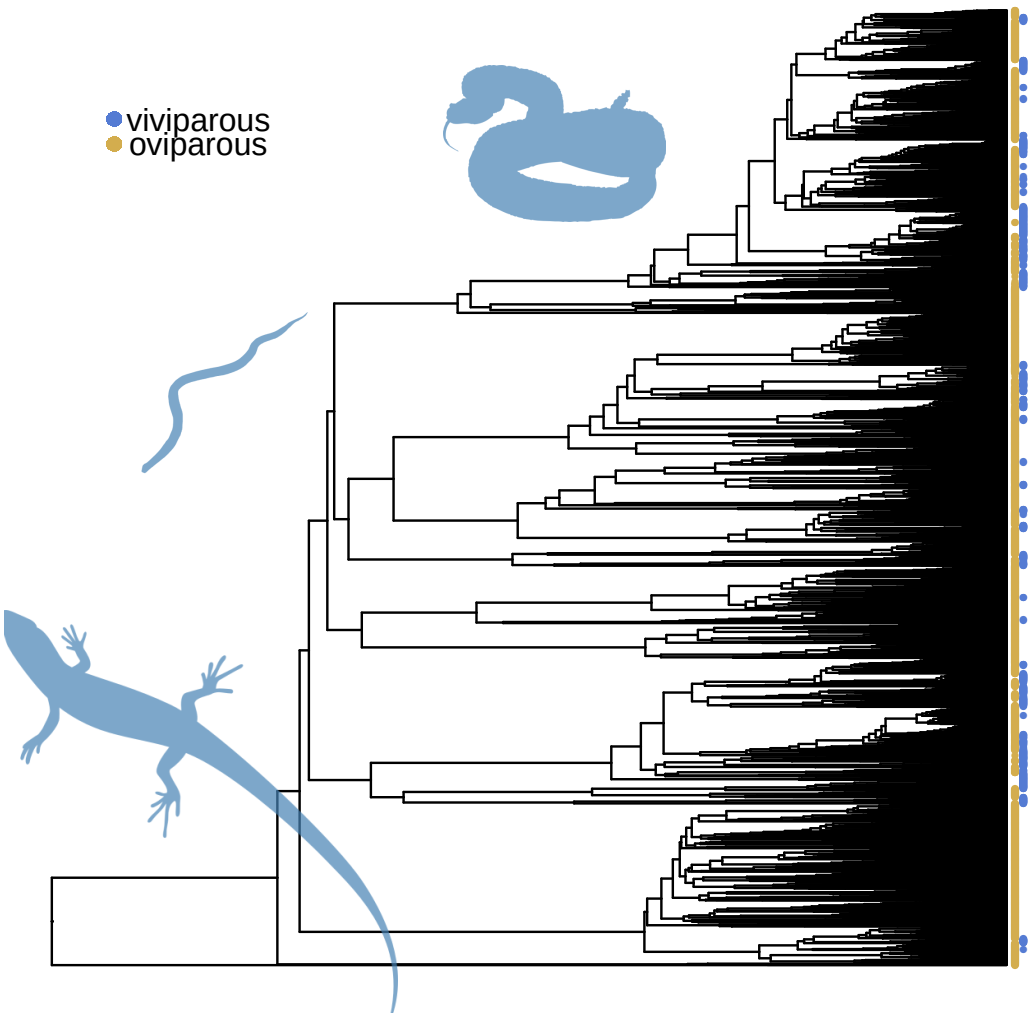

Figure 5.1: Squamate phylogeny. Tips with blue dots correspond to viviparous species. Tips with brown dots correspond to oviparous species.

Bayes Factors

We computed Bayes factors to address our question of interest after producing the posterior distribution of the trait history of squamate reproductive modes. Our null hypothesis was that the MRCA of squamates was oviparous, $H_0 : X_0 = \text{ovi}$, where X_0 represents the state of the root node and ovi represents oviparity. The alternative hypothesis was that the MRCA of squamates was viviparous, $H_a : X_0 = \text{vivi}$, where vivi represents viviparity. A Bayes factor test [Kass and Raftery, 1995] allowed us to compare these two hypotheses,

$$\text{BF} = \frac{\Pr(\mathbf{y}|H_0)}{\Pr(\mathbf{y}|H_a)} = \frac{\Pr(H_0|\mathbf{y})/\Pr(H_0)}{\Pr(H_a|\mathbf{y})/\Pr(H_a)} = \frac{\Pr(X_0 = \text{ovi}|\mathbf{y})/\Pr(X_0 = \text{ovi})}{\Pr(X_0 = \text{vivi}|\mathbf{y})/\Pr(X_0 = \text{vivi})}.$$

The smaller the Bayes factor is in this context the more support there is for the alternative hypothesis of live birth.

Two State Markov Model Results

The two state analysis with no hidden regimes used gamma priors for the rate matrix parameters, λ_{01} and λ_{10}. The shape and rate parameters for both priors were 1 and 100 respectively. The prior distribution on the state of the root node was uniform over the two possible states. The posterior probability that the root was in state 1 (viviparity) was just under 60%. This corresponds to a Bayes factor of 0.7, which is not compelling evidence in either direction. Posterior distributions of transition counts between the two states over the entire tree can be found in Figure 5.2. The number of transitions from viviparity to oviparity are far larger than biologists deem plausible.

Interestingly, our analysis does not reproduce the results found in Pyron and Burbrink [2014] and King and Lee [2015], namely that the two state analysis produces strong support for a viviparous root. Of course, our model of evolution was simpler than the model used in Pyron and Burbrink [2014] and King and Lee [2015] as both papers used a six parameter model that estimated state-dependent speciation and extinction rates in addition to character transition rates.

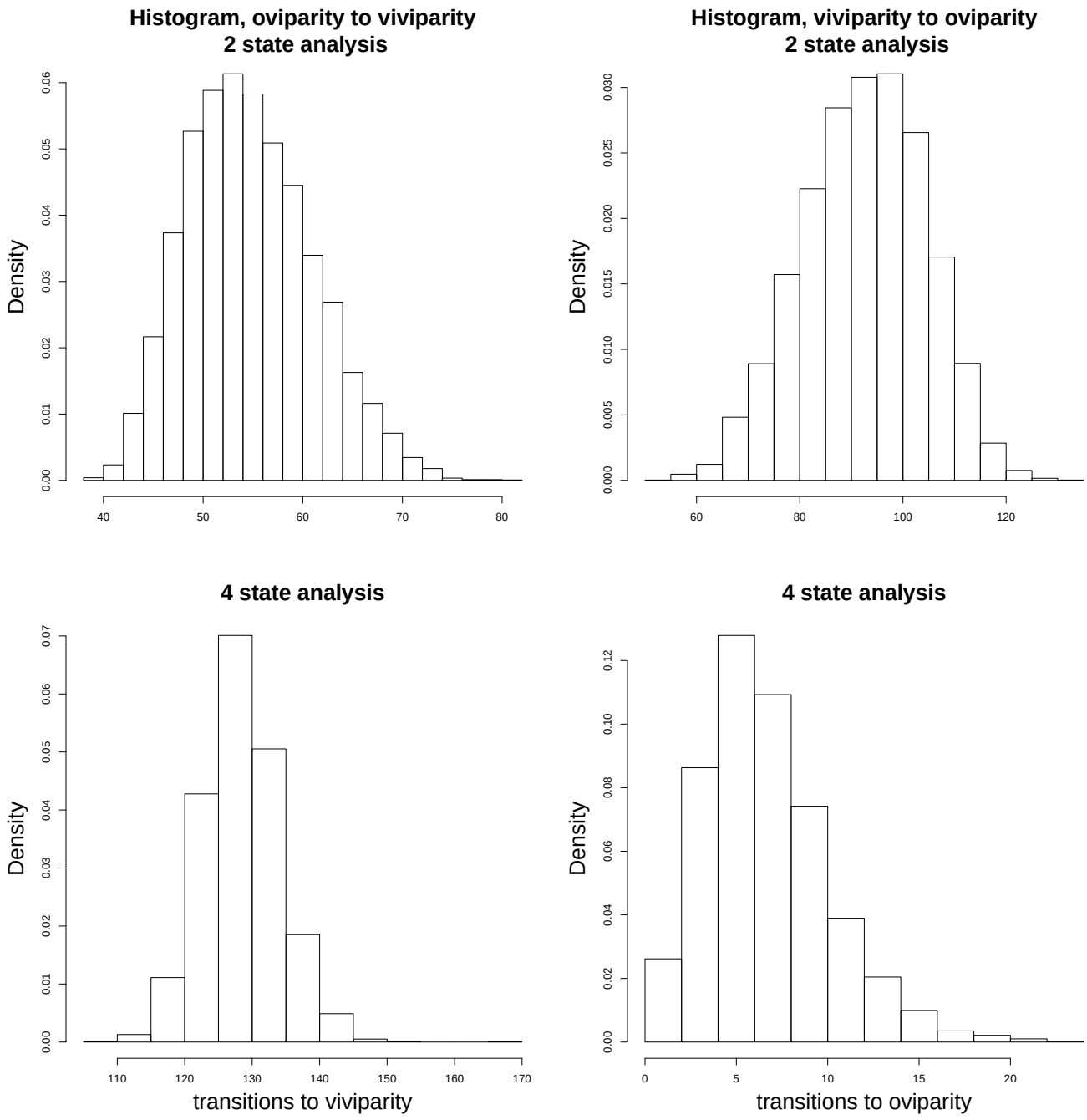

Figure 5.2: Histograms of the posterior distributions concerning the number of transitions between oviparity and viviparity in squamates. The plots in the top row were produced by the two state analysis. The priors on λ_{01} and λ_{10} were $\Gamma(1, 100)$. The plots in the bottom row were produced by the four state analysis with $\Gamma(1, 550)$ priors on λ_{01} and λ_{10}. These two sets of priors are compared because they produced about the same average number of transitions to viviparity in simulations.

Four State Markov Model Results

The four state analysis with slow and fast hidden regimes used gamma priors for the five rate matrix parameters. The shape and rate parameters for the λ_{01} and λ_{10} priors were 1 and 100 respectively. The shape and rate parameters for the κ_{01} and κ_{10} priors were 1 and 10 respectively. The shape and rate parameters for the γ prior were 5 and 0.5 respectively. The prior distribution on the root state was uniform across the four possible states.

The posterior distribution on the state of the root node of the most recent common ancestor of Squamata places 0.2 percent of the weight on live birth. 99.8% of the posterior distribution supports oviparity. This corresponds to a Bayes factor of 470, very strong support for the null hypothesis. We considered three alternative sets of priors for the λ_{01} and λ_{10} parameters (while leaving the priors for κ_{01}, κ_{10}, and γ unchanged). Results derived from these priors can be seen in Figure 5.3. All four sets of priors resulted in support for an oviparous root ranging from postive to very strong. *A posteriory*, the number of transitions from viviparity to oviparity was below 20 with high probability, with the posterior mode of 5 (see Figure 5.2). Such counts are far more biologically plausible than the counts derived from the 2 state analysis, and also help explain why we see such strong support for an oviparous ancestor. To explain widespread oviparity at the tips of the tree, a viviparous root, and a small number of transitions from viviparity to oviparity would strongly suggest at least one viviparous to oviparous transition near the root of the tree. However, the small number of viviparous to oviparous transitions we do find in the posterior distribution of substitution histories do not occur near the root, instead they occur after oviparous to viviparous transitions. Many transitions to viviparity along with the small number of reversions to oviparity that do not occur near the root allow us to conclude that there is strong support for an oviparous ancestor.

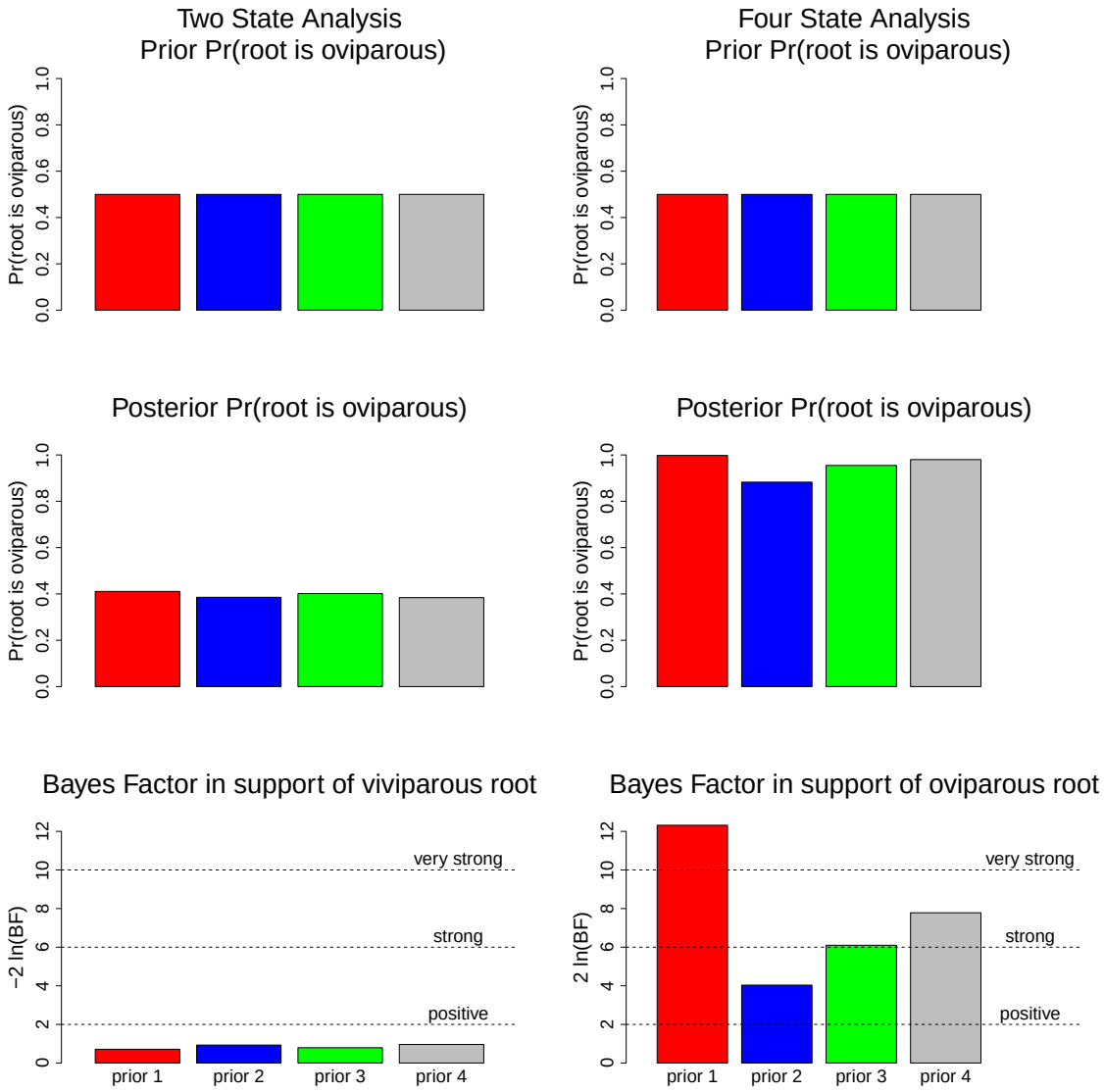

Figure 5.3: Squamate results concerning the state of the root node using a two state model of evolution (left) and a four state model of evolution (right). The four different colors represent four different sets of prior distributions on λ_{01} and λ_{10}. Each set of priors was composed of two independent gamma distributions. The shape and rate parameters in the red section for both λ_{01} and λ_{10} were 1 and 100 respectively. The shape and rate parameters in the blue section for both λ_{01} and λ_{10} were 1 and 10 respectively. The shape and rate parameters in the green section were 1 and 10 for λ_{01} and 1 and 100 for λ_{10}. The shape and rate parameters in the grey section were 1 and 100 for λ_{01} and 1 and 10 for λ_{10}. The first row shows the prior probability of the root being in state 0 (oviparous). The second row shows the posterior probability of the root being in state 0. The third row shows a transformed version of the Bayes factor in support of the null hypothesis, the root of squamata was oviparous.

More Regimes

We investigated the sensitivity of our conclusions to increasing the number of hidden regimes. We considered 3 regimes/6 states, 4 regimes/8 states, and 5 regimes/10 states. We used gamma priors for the rate matrix parameters. The shape and rate parameters for the λ_{01} and λ_{10} priors were 1 and 100 respectively. The shape and rate parameters for all the κ priors were 1 and 10 respectively. The shape and rate parameters for all the γ priors were 5 and 0.5 respectively. The prior distribution on the root state was uniform across the possible states. We found strong support for oviparity in each case. The Bayes factors in support of oviparity were 285, 475, and 243, for the 6 state, 8 state, and 10 state analyses respectively.

Squamate Conclusion

Our analysis found a marked difference between the 2 state and the 4 state analysis. The 4 state analysis found very strong support for an oviparous ancestor of squamates while the 2 state analysis did not find strong evidence in either direction. These results do not agree with Pyron and Burbrink [2014] and King and Lee [2015]. Both papers report strong support for viviparity at the root using a 2 state anaysis that estimated state-dependent speciation and extinction rates in addition to character transition rates. The lack of agreement between phylomap and BiSSE may be due to the additional parameters that Maddison et al. [2007] estimates.

The 4 state analysis also disagrees with results in King and Lee [2015]. King and Lee [2015] found that anayses with 2 or 3 rate categories (4 or 6 states) supported a viviparous ancestor while analyses with 4 or 5 rate categories supported an oviparous ancestor.

5.2 Cephalopods

Cephalopods are marine animals with tentacles; the class includes both octopuses and squid. Some cephalopods are bioluminescent, they can produce their own light. We were interested in a specific type of bioluminescence produced by photophores, an organ containing biolumi-

nescent bacteria. Pankey et al. [2014] found that bacterial photophores arose separately at least twice during the course of cephalopod evolution. The data used in their analysis came from 70 cephalopod species. Evolutionary trees relating these species were estimated from concatenated nucleotide sequences from up to 13 loci. An example phylogeny can be see in Figure 5.4. 20 of the species contained bacterial photophores while 50 did not. We were interested in extending the analysis from Pankey et al. [2014] to allow for a more complicated model of binary trait evolution.

Stochastic mapping allowed us to create a posterior distribution for n_{01}, the number of times bacterial photophores arose over the cephalopod phylogeny. The root node was assumed to start in state 1, the bioluminescent state. State 0 corresponds to the lack of biolumenescent bacterial photophores. This assumption on the state of the root node was made in the interest of producing a conservative analysis, as we expect bioluminescence to develop rarely. The priors for the two rate parameters, λ_{01} and λ_{10}, were both gamma distributions. For each set of priors we estimated the prior and posterior probabilities of the number of gains, averaging over all 1,000 trees. From these probabilities we computed Bayes factors. The null hypothesis in this situation was that $n_{01} \leq k$ while the alternative hypothesis was that $n_{01} > k$. A Bayes factor approach allowed us to compare these two hypotheses,

$$\text{BF} = \frac{\Pr(\mathbf{y}|n_{01} \leq k)}{\Pr(\mathbf{y}|n_{01} > k)} = \frac{\Pr(n_{01} \leq k|\mathbf{y})/\Pr(n_{01} \leq k)}{\Pr(n_{01} > k|\mathbf{y})/\Pr(n_{01} > k)}.$$

Smaller Bayes factors meant more support for the alternative hypothesis, the number of gains was greater than k. Results from the two state, no hidden regime model can be found in Figure 5.5. Results from the four state, two regime model can be found in Figure 5.6. The two state analysis led us to conclude that bacterial photophores were gained at least four times on the phylogeny of cephalopods. This result was upheld across a variety of prior distributions. The four state analysis was less conclusive but still suggests that there were at least two gains.

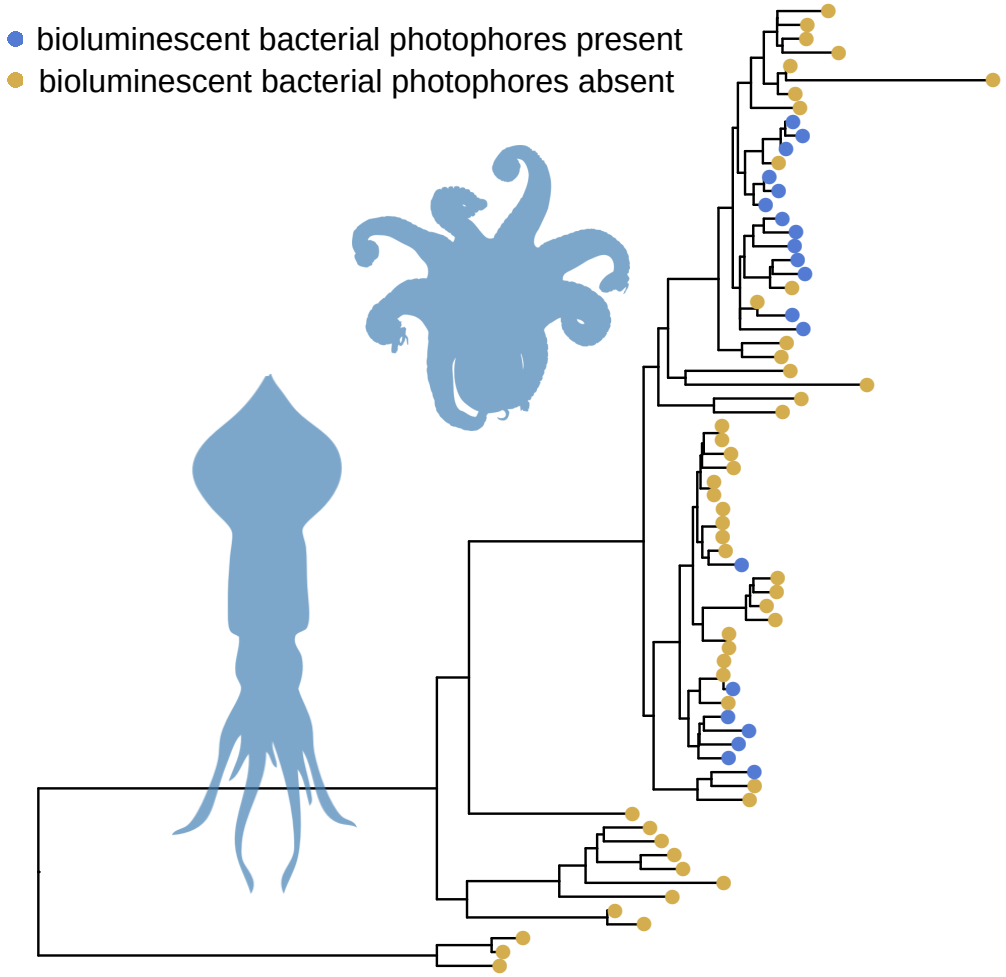

Figure 5.4: Example cephalopod phylogeny. Tips with red dots correspond to species that have bioluminescent bacterial photophores. Tips with black dots correspond to species that do not have bioluminescent bacterial photophores.

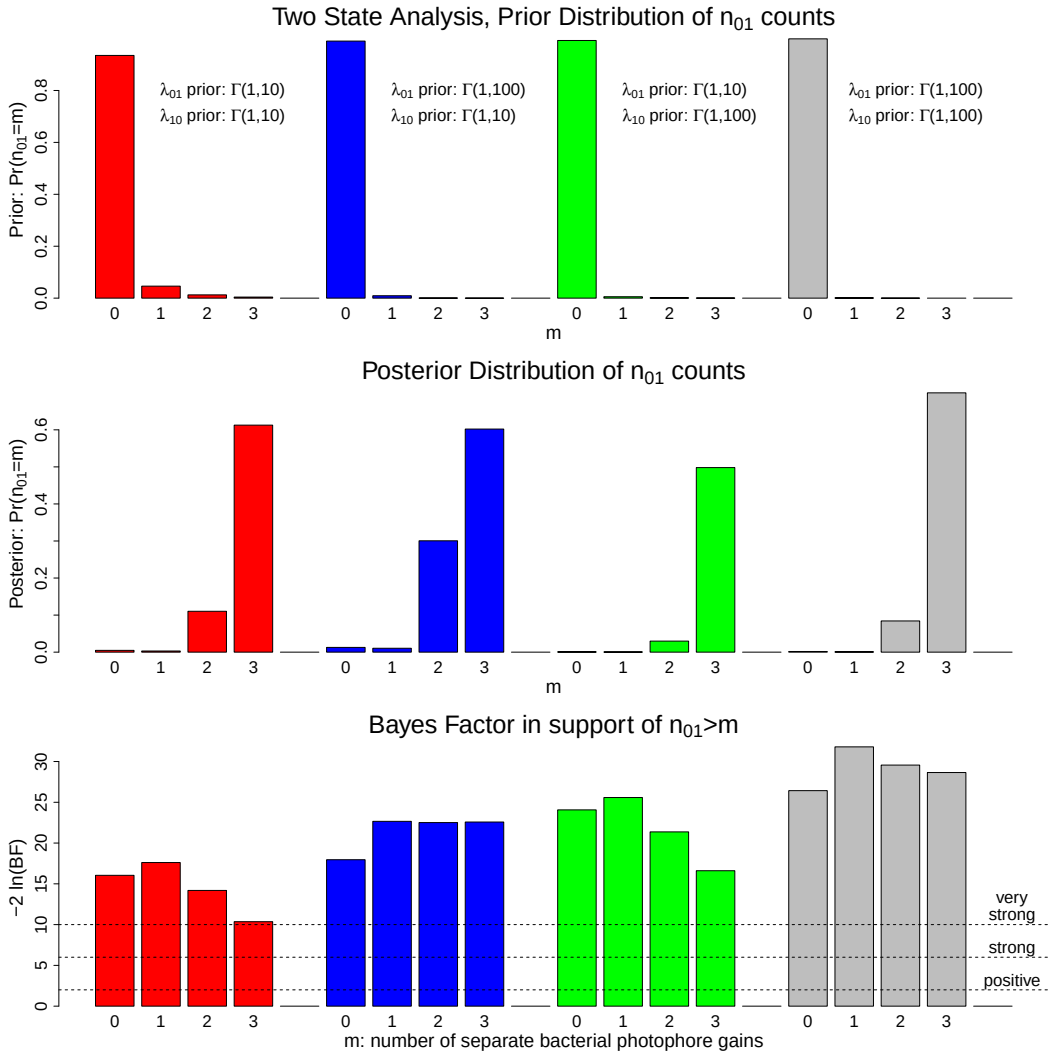

Figure 5.5: Cephalopod results concerning the number of times bacterial photophores separately arose using a 2-state model of evolution. The root node was fixed in the analysis to be in state 1 (presence of bacterial photophores). The four different colors represent four different sets of prior distributions on λ_{01} and λ_{10}. Each set of priors was composed of two independent gamma distributions. The shape and rate parameters in the red section for both λ_{01} and λ_{10} were 1 and 10 respectively. The shape and rate parameters in the blue section were 1 and 100 for λ_{01} and 1 and 10 for λ_{10}. The shape and rate parameters in the green section were 1 and 10 for λ_{01} and 1 and 100 for λ_{10}. The shape and rate parameters in the grey section for both λ_{01} and λ_{10} were 1 and 100 respectively. The first row shows the prior probability of having 0, 1, 2, and 3 transitions from state 0 to state 1. The second row shows the posterior probability of having 0, 1, 2, and 3 transitions from state 0 to state 1. The third row shows a transformed version of the Bayes factor in support of the alternative hypothesis, the number of transitions from state 0 to state 1 was greater than m.

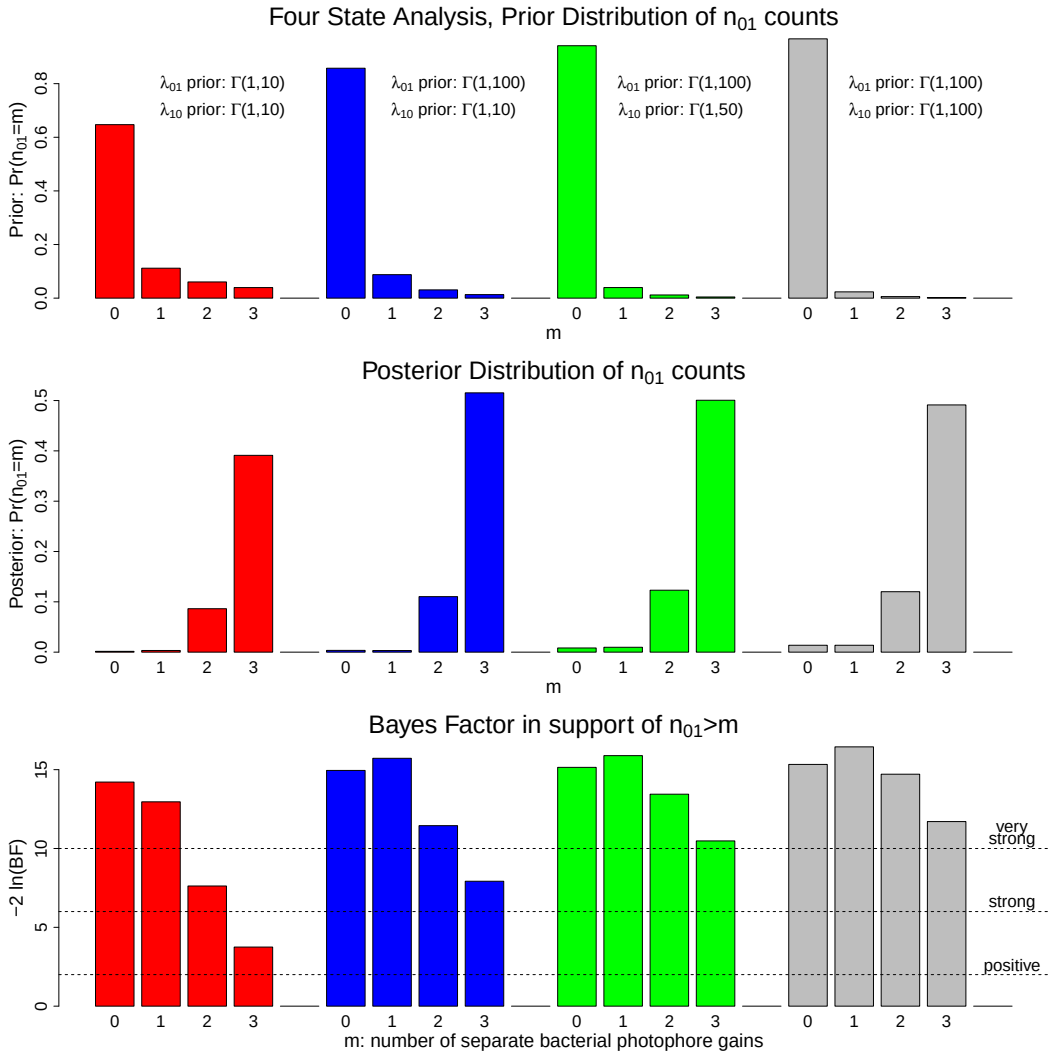

Figure 5.6: Cephalopod results concerning the number of times bacterial photophores separately arose using a 4-state model of evolution. The root node was fixed in the analysis to be in state 1 (presence of bacterial photophores). The four different colors represent four different sets of prior distributions on λ_{01} and λ_{10}. The priors on κ_{01}, κ_{10}, and γ did not change. The κ priors were gamma distributions with rate and shape parameters of 1 and 10 respectively. The γ prior was a gamma distribution with rate and shape parameters 5 and 0.5 respectively. Each set of priors for λ_{01} and λ_{10} was composed of two independent gamma distributions. The shape and rate parameters in the red section for both λ_{01} and λ_{10} were 1 and 10 respectively. The shape and rate parameters in the blue section were 1 and 100 for λ_{01} and 1 and 10 for λ_{10}. The shape and rate parameters in the green section were 1 and 10 for λ_{01} and 1 and 100 for λ_{10}. The shape and rate parameters in the grey section for both λ_{01} and λ_{10} were 1 and 100 respectively. The first row shows the prior probability of having 0, 1, 2, and 3 transitions from state 0 to state 1. The second row shows the posterior probability of having 0, 1, 2, and 3 transitions from state 0 to state 1. The third row shows a transformed version of the Bayes factor in support of the alternative hypothesis, the number of transitions from state 0 to state 1 was greater than m.

5.3 Discussion

We have introduced a new Bayesian hidden rates model for phylogenetic stochastic mapping akin to Beaulieu et al. [2013]'s maximum likelihood approach. We prefer the Bayesian approach because the maximum likelihood approach conditions on the most likely value of rate matrix parameters before drawing conclusions about a trait's history. Uncertainty about the rate matrix parameters is therefore left out of the maximum likelihood analysis. Our new method accounts for the uncertainty in rate matrix parameters during the stochastic mapping procedure.

By implementing rate matrix parameter updates we have improved our stochastic mapping procedure from Irvahn and Minin [2014], which originally was only shown to work for a fixed rate matrix. This advancement is crucial as it will allow our algorithm to be used for phylogenetic stochastic mapping on large state spaces with unknown rate matrix parameters.

We demonstrated the importance of using a hidden rates model in the analysis of the reproductive mode of squamate phylogenies. Failure to account for rate matrix heterogeneity across the large phylogeny led to what is likely an erroneous conclusion. We found that there is strong support for an oviparous root node in Squamata, a result that is not found by using a simple two state model of evolution. We also showed that the conclusion about multiple separate origins of bacterial photophores in cephalopods was still supported when we moved from the two state model of photophore evolution to the hidden rate model to account for gain and loss rate heterogeneity across the cephalopod phylogeny.

Finally, we created a new way to average over the uncertainty in the topologies relating the species of interest. This approach is both unsatisfying from a methodological point of view and slow from a practical point of view. Sampling substitution histories for an entire set of trees at each iteration of the MCMC is far slower than sampling a single substitution history at each iteration. This clearly points the way to future methodological work. We need to develop a way to jointly sample substitution histories and topologies.

5.4 Model Checking

In this chapter we investigated specific biological hypotheses using different models of evolution, a simple 2-state binary model and a hidden rate 4-state model. We investigated the 4-state model because we were concerned that our results were sensitive to our choice of model. In the case of cephalopods we found that there was little need to be concerned; the 2-state and 4-state analyses reached similar conclusions. In the case of squamates we found that our choice of model really did matter. The 4-state analysis found support for an egg-laying ancestor while the 2-state analysis found little evidence in either direction.

After addressing our questions, and running the anayses with both models we were no closer to knowing which model was more appropriate. In this section we investigate the observed data's ability to distinguish between data generating models, the 2-state or the 4-state (or the 6-state, or more). This is a model comparison problem and there are multiple frameworks that can be used to address our questions. We investigated two frameworks, the deviance information criterion (DIC), and posterior predictive distributions.

5.4.1 Deviance Information Criterion

The DIC [Spiegelhalter et al., 2002] was developed to compare alternative models with the goal of identifying a class of models that adequately described information in the data. This issue had been explored earlier, notably via the the Akaike information criterion (AIC) [Akaike, 1973] and the Bayesian information criterion (BIC) [Kass and Raftery, 1995]. The basic concern addressed by these information criteria is that more complex models fit the data better so we need methods to balance model complexity against model fit. Spiegelhalter et al. [2002] developed a measure of the effective number of parameters, p_D, that can be used to create the deviance information criterion as discussed below. We follow the presentation described in Spiegelhalter et al. [2014].

Our rate matrix is parameterized by the vector, $\boldsymbol{\theta}$. The deviance, $D(\boldsymbol{\theta}) = -2\log[p(y|\boldsymbol{\theta})]$, is a function of $\boldsymbol{\theta}$. The effective number of parameters suggested in Spiegelhalter et al. [2002]

is

$$p_D = E_{\boldsymbol{\theta}|y}\left\{-2\log[p(y|\boldsymbol{\theta})]\right\} + 2\log\left[p\left(y|\tilde{\boldsymbol{\theta}}(y)\right)\right].$$

If $\tilde{\boldsymbol{\theta}} = E[\boldsymbol{\theta}|y]$, the posterior mean of $\boldsymbol{\theta}$, then p_D is the 'posterior mean deviance — deviance of posterior means' [Spiegelhalter et al., 2014]. Combining the deviance and the effective number of parameters yields the deviance information criterion,

$$\text{DIC} = D(\bar{\boldsymbol{\theta}}) + 2p_D.$$

Given competing models and a data set we prefer the model that produces the smallest DIC value.

Deviance Information Criterion, Augmented

In the interest of avoiding matrix exponentiation when calculating DIC we investigated a modified version of DIC that conditions on the number of transitions on each branch as well as the rate matrix parameters. The number of transitions on each branch is denoted by the vector, $\mathbf{m}$. The augmented likelihood, $p(y|\boldsymbol{\theta}, \mathbf{m})$, is the likelihood of the observed trait data at the tips of the tree conditional on the parameter values, $\boldsymbol{\theta}$, and on the vector of branch transition counts, $\mathbf{m}$. This augmented likelihood can be created by combining part of the partial likelihood matrix with the vector of prior probabilities placed on the state of the root,

$$p(y|\boldsymbol{\theta}, \mathbf{m}) = \mathbf{l}_{(\text{root -})}\boldsymbol{\pi}.$$

The vector, $\mathbf{l}_{(\text{root -})}$, is the row of the partial likelihood matrix (described in Chapter 3) corresponding to the root node. The k^{th} element of $\mathbf{l}_{(\text{root -})}$, $l_{(\text{root } k)}$, is the likelihood of the observed trait data at the tips of the tree conditional on the root being in state k, conditional on $\boldsymbol{\theta}$, and conditional on $\mathbf{m}$. The augmented deviance, $D_a(\boldsymbol{\theta}) = -2\log(p(y|\boldsymbol{\theta}, \mathbf{m}))$, is a function of $\boldsymbol{\theta}$ and $\mathbf{m}$. The effective number of parameters is

$$p_{D_a} = E_{\boldsymbol{\theta},\mathbf{m}|y}\left[-2\log(p(y|\boldsymbol{\theta}, \mathbf{m}))\right] + 2\log\left[p\left(y|\tilde{\boldsymbol{\theta}}(y), \tilde{\mathbf{m}}(y)\right)\right].$$

We used the posterior median of the number of transitions on each branch for $\tilde{\mathbf{m}}(y)$. Combining the augmented deviance and the effective number of parameters yields the augmented deviance information criterion,

$$\mathrm{DIC}_a = D_a(\bar{\boldsymbol{\theta}}, \tilde{\mathbf{m}}) + 2p_{D_a}.$$

In our investigations DIC_a values were very close to the corresponding DIC values.

Data Simulation

Using a 70 tip tree associated with the cephalopod data we simulated 40 sets of tip data, 20 from a 2-state model and 20 from a 4-state model. The rate matrix for the 2-state model was

$$\mathbf{Q} = \begin{pmatrix} -0.22 & 0.22 \\ 0.67 & -0.67 \end{pmatrix}.$$

The rate matrix for the 4-state model was,

$$\mathbf{Q} = \begin{pmatrix} -0.3 & 0.1 & 0.2 & 0.0 \\ 0.1 & -0.3 & 0.0 & 0.2 \\ 0.2 & 0.0 & -1.2 & 1.0 \\ 0.0 & 0.2 & 1.0 & -1.2 \end{pmatrix}.$$

Priors

Some results are sensitive to the choice of prior so we investigated 3 different sets of priors, restricted, diffuse, and spike.

Restricted Prior The restricted prior sets produced similar n_{01} counts under the 2-state and 4-state models. The shape and rate parameters of the gamma priors for λ_{01} and λ_{10} in the 2-state analysis were 0.55 and 1. The average rate matrix produced by these priors was,

$$\begin{pmatrix} -0.55 & 0.55 \\ 0.55 & -0.55 \end{pmatrix}.$$

The shape and rate parameters of the gamma priors for λ_{01} and λ_{10} in the 4-state analysis were 1 and 10. The shape and rate parameters of the gamma priors for κ_{01} and κ_{10} in the 4-state analysis were 2 and 10. The shape and rate parameters of the gamma prior for γ in the 4-state analysis were 20 and 2. The average rate matrix produced by these priors was,

$$\begin{pmatrix} -0.3 & 0.1 & 0.2 & 0.0 \\ 0.1 & -0.3 & 0.0 & 0.2 \\ 0.2 & 0.0 & -1.2 & 1.0 \\ 0.0 & 0.2 & 1.0 & -1.2 \end{pmatrix},$$

exactly the same as the actual rate matrix used to produce the 4-state simulated data.

Both the 2-state and 4-state priors were used to simulate many rate matrices which were then used to sample substitution histories on our tree. The distribution of n_{01} resulting from these two sets of rate matrix priors can be seen in Figure 5.7. The two prior sets produce very similar distributions for n_{01}.

Diffuse Prior We also considered a diffuse prior where the shape and rate parameters of the priors for each parameter were 1 and 0.1 respectively. The means of these priors were larger than desired but they were spread out and did not have a spike at 0. Since the parameters tend to take larger values the number of trait gains these prior sets produced were also large. At the same time these priors allowed for a small number of trait gains as seen in Figure 5.8.

Spike Prior We also considered a spike prior where the shape and rate parameters of the priors for each parameter were 0.01 and 0.01 respectively. These priors placed a lot of mass at 0 while being quite flat elsewhere.

Squamate Tree In addition to the 40 sets of tip data associated with the 70 tip cephalopod tree we simulated 20 sets of tip data using the 3,951 tip squamate tree, 10 from a 2-state

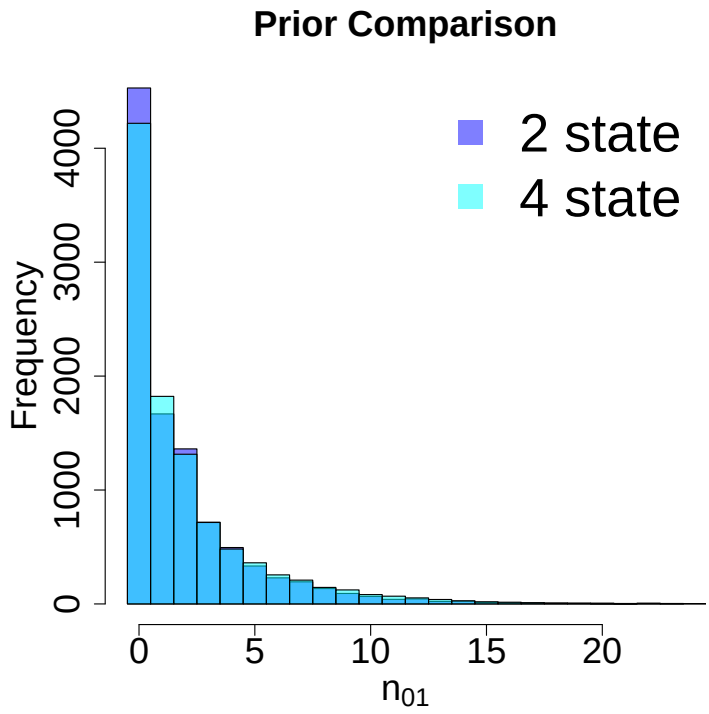

Figure 5.7: Distribution of the total number of trait gains (n_{01}) for the restricted prior set using a 2-state model (dark blue) and a 4-state model (light blue).

model and 10 from a 4-state model. The rate matrix for the 2-state model was,

$$\mathbf{Q} = \begin{pmatrix} -0.001 & 0.001 \\ 0.006 & -0.006 \end{pmatrix}.$$

The rate matrix for the 4-state model was,

$$\mathbf{Q} = \begin{pmatrix} -0.002 & 0.001 & 0.001 & 0.0 \\ 0.001 & -0.002 & 0.0 & 0.001 \\ 0.03 & 0.0 & -0.046 & 0.016 \\ 0.0 & 0.03 & 0.016 & -0.046 \end{pmatrix}.$$

We analyzed these 20 simulated data sets using the same three sets of priors, restricted, diffuse, and spike.

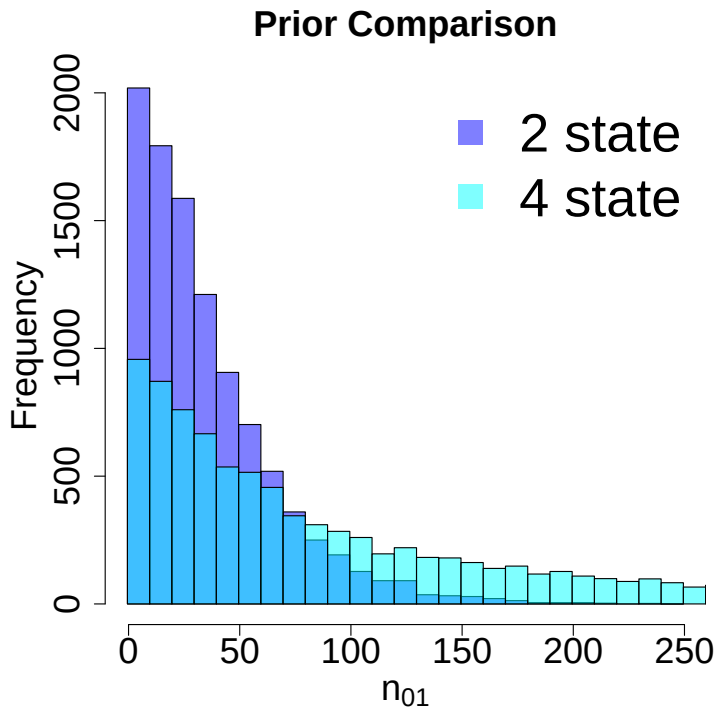

Figure 5.8: Distribution of the total number of trait gains (n_{01}) for two specific priors on a 2-state rate matrix model (dark blue) and a 4-state rate matrix model (light blue) using diffuse priors.

Simulation Results, Cephalopod and Squamate Trees

A summary of the results produced by the simulations on the squamate tree can be found in Table 5.3. The DIC results were not able to reliably select the true data generating model. For comparison purposes we calculated AIC values, as implemented in Beaulieu et al. [2013]'s corHMM program, for our simulated data sets. The AIC approach also did not reliably select the true data generating model.

For the 70 tip tree we simulated 40 sets of tip data, 10 from a 2-state model with the root fixed in state 1, 10 from a 4-state model with the root fixed in state 1, 10 from a 2-state model with the root not fixed, and 10 from a 4-state model with the root not fixed. The

	restricted	diffuse	spike
2-state DIC < 4-state DIC (true model: 2-state)	9/10	1/10	10/10
4-state DIC < 2-state DIC (true model: 4-state)	2/10	8/10	1/10

Table 5.1: DIC comparisons between 2-state and 4-state analyses of simulated data on a 70 tip tree with the root fixed in state 1.

	restricted	diffuse	spike	AIC
2-state DIC < 4-state DIC ('true' model: 2-state)	4/10	2/10	9/10	10/10
4-state DIC < 2-state DIC ('true' model: 4-state)	3/10	8/10	3/10	0/10

Table 5.2: DIC (and AIC) comparisons between 2-state and 4-state analyses of simulated data on a 70 tip tree with the root state not fixed.

	restricted	diffuse	spike	corHMM AIC
2-state DIC < 4-state DIC (true model: 2-state)	1/10	5/10	9/10	10/10
4-state DIC < 2-state DIC (true model: 4-state)	10/10	8/10	5/10	1/10

Table 5.3: DIC (and AIC) comparisons between 2-state and 4-state analyses of simulated data on a 3,951 tip tree.

results of our analyses are summarized in Tables 5.1 and 5.2. In this setting we found that DIC (and AIC) was not able to reliably select the true data generating model.

DIC Conclusions

DIC did not have power to discriminate between the 2-state and 4-state data generating regimes we investigated. Our augmented DIC_a values were very close to the corresponding DIC values and the smaller DIC_a value matched the smaller DIC value in almost all analyses. DIC conclusions were sensitive to the choice of prior. We found that DIC was not a useful metric for model selection when we were concerned with the possibility of hidden rate regimes

in binary trait evolution.

5.4.2 Posterior Predictive Assessment

The basic idea behind posterior predicitive assessment is that replicated data, $\mathbf{y}^{\text{rep}}$, should look similar to the observed data, $\mathbf{y}$, when we select a reasonable model. We follow the explanantion of posterior predicitive assessments as it was outlined in Gelman et al. [1996]. We denote the model with the letter, H, and the parameters of H with $\boldsymbol{\theta}$. The test statistic, T, is a function from the data to the real numbers. A set of auxiliary statistics, $A(\mathbf{y})$, are functions of the data that are held constant in replicated data sets such as sample size. The distribution of replicated data sets is $P_A(\mathbf{y}^{\text{rep}}|H,\boldsymbol{\theta}) = P_A(\mathbf{y}^{\text{rep}}|H,\boldsymbol{\theta},A(\mathbf{y}^{\text{rep}}) = A(\mathbf{y}))$. The classical p-value based on T is $p_c(\mathbf{y},\boldsymbol{\theta}) = P_A(T(\mathbf{y}^{\text{rep}}) \geq T(\mathbf{y})|H,\boldsymbol{\theta})$. As noted in Gelman et al. [1996] p-values close to zero indicate a lack of model fit because the test statistic was improbabale given the model.

The classical formulation has difficulty when the distribution of the test statistic depends on specific and unknown values of the model parameters, $\boldsymbol{\theta}$. In a Bayesian framework we can rework the distribution of replicated data sets by integrating out our uncertainty in $\boldsymbol{\theta}$ over its posterior distribution, $P(\boldsymbol{\theta}|H,\mathbf{y})$. In this context the reference distribution of replicated data sets is, $P_A(\mathbf{y}^{\text{rep}}|H,\mathbf{y}) = \int P_A(\mathbf{y}^{\text{rep}}|H,\boldsymbol{\theta})P(\boldsymbol{\theta}|H,\mathbf{y})d\boldsymbol{\theta}$. Again, we can use this distribution of $\mathbf{y}^{\text{rep}}$ to calculate a p-value by computing a tail area probability, $p_b(\mathbf{y}) = P_A(T(\mathbf{y}^{\text{rep}}) \geq T(\mathbf{y})|H,\mathbf{y}) = \int p_c(\mathbf{y},\boldsymbol{\theta})P(\boldsymbol{\theta}|H,\mathbf{y})d\boldsymbol{\theta}$. This is the posterior predictive p-value.

One of the nice properties of posterior predicitve assessment is the introduction of a discrepancy, $D(\mathbf{y},\boldsymbol{\theta})$ that can take the place of the test statistic, $T(\mathbf{y})$. The use of a discrepancy is important because it allows us to compare the discrepancy between the observed data and the model rather than the best fit of the model. Conditioning on a specific $\boldsymbol{\theta}$ that we do not know is undesirable and as it turns out, unnecessary. Instead of considering the distribution of replicated data sets we can consider the joint distribution of replicated data sets and model parameters, $P_A(\mathbf{y}^{\text{rep}},\boldsymbol{\theta}|H,\mathbf{y}) = P_A(\mathbf{y}^{\text{rep}}|H,\boldsymbol{\theta})P(\boldsymbol{\theta}|H,\mathbf{y})$. In this setting we can not calculate a single discrepancy value for the observed data because of the dependence

on $\boldsymbol{\theta}$ but we can calculate a tail area probability, $p_b(\mathbf{y}) = p_A(D(\mathbf{y}^{\text{rep}}, \boldsymbol{\theta}) \geq D(\mathbf{y}, \boldsymbol{\theta})|H, \mathbf{y})$. One must take care in interpreting this probability as it has been shown that this quantity is not distributed unifomly between 0 and 1 under the true model [Robins et al., 2000]. We might be able to identify unreasonable models by comparing the distributions of posterior predictive discrepancies of replicated data to the posterior predictive discrepancies of the observed data.

Computing discrepancies for replicated data and observed data can be accomplished by incorporating a few additional computations at each iteration of a Monte Carlo analysis. For each newly sampled set of parameters, $\boldsymbol{\theta}^j$, we sample a replicated data set, $\mathbf{y}^{\text{rep}\ j}$, from the sampling distribution, $P_A(\mathbf{y}^{\text{rep}}|H, \boldsymbol{\theta}^j)$. Then we calculate and record $D(\mathbf{y}, \boldsymbol{\theta}^j)$ and $D(\mathbf{y}^{\text{rep}\ j}, \boldsymbol{\theta}^j)$. Once a Monte Carlo sample of $D(\mathbf{y}, \boldsymbol{\theta}^j)$ and $D(\mathbf{y}^{\text{rep}\ j}, \boldsymbol{\theta}^j)$ is assembled the two discrepancies can be plotted against each other allowing a graphical assessment of the realized discrepancies.

The particular models we were interested in investigating were the 2-state and 4-state hidden regime models for a binary trait. The discrepancy we used was the expected number of transitions from state 0 to state 1 (in any regime), $E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta})$. For a four state analysis n_{01} encompasses both transitions from state 0 to state 1 AND transitions from state 2 to state 3. At each iteration of the MCMC we simulated a new data set, $\mathbf{y}^{\text{rep}}$, conditional on current parameter values, $\boldsymbol{\theta}^{\text{curr}}$. The new data set was then used to sample 100 independent substitution histories using matrix exponentiation stochastic mapping. The sampled substitution histories were used to approximate the test statistic, $E(\mathrm{n}_{01}|\mathbf{y}^{\text{rep}}, \boldsymbol{\theta}^{\text{curr}})$. At each iteration of the MCMC we also sampled 100 independent substitution histories using matrix exponentiation for the observed data. These substitution histories were used to approximate $E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta}^{\text{curr}})$.

Data Simulation

Using a 70 tip tree associated with the cephalopod data we simulated 10 trait histories resulting in 10 sets of tip data using a 4-state model of evolution using the following rate

matrix,

$$\mathbf{Q} = \begin{pmatrix} -0.3 & 0.1 & 0.2 & 0.0 \\ 0.1 & -0.3 & 0.0 & 0.2 \\ 0.2 & 0.0 & -1.2 & 1.0 \\ 0.0 & 0.2 & 1.0 & -1.2 \end{pmatrix}.$$

The 'true' number of trait gains for each of the 10 trait histories was 3, 1, 6, 4, 0, 1, 1, 1, 3, and 8. The average number of trait gains using this rate matrix was 2.9. The distribution of the number of trait gains that the rate matrix produced can be seen in Figure 5.9.

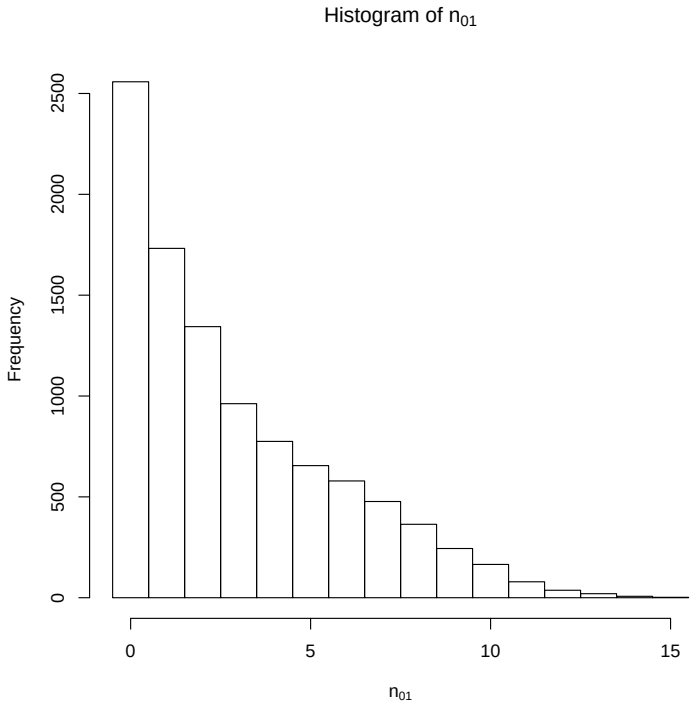

Figure 5.9: Distribution of the total number of trait gains (n_{01}) for a specific 4-state rate matrix and 70 tip tree.

Using a 70 tip tree associated with the cephalopod data we also simulated 10 trait histories resulting in 10 sets of tip data using a 2-state model of evolution using the following rate

matrix,

$$\mathbf{Q} = \begin{pmatrix} -0.22 & 0.22 \\ 0.67 & -0.67 \end{pmatrix}.$$

The 'true' number of trait gains for each of the 10 trait histories was 0, 0, 0, 2, 4, 1, 2, 1, 2, and 3. The posterior predictive reference distributions were sensitive to the choice of prior so we investigated 3 different sets of priors, restricted, diffuse, and spike. These were the same priors described in the DIC section.

Posterior Predictive Results

We found that replicated data from both the 2-state and the 4-state models were, in general, similar to the observed data when considering the discrepancy, $E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta})$. Posterior predictive plots for all 20 data sets (10 produced from the 2-state model and 10 produced from the 4-state model) can be found in Appendix B. Tail probabilities, $p_A(D(y^{\mathrm{rep}}, \theta) \geq D(y, \theta)|H, y)$, for these simulations can be found in Tables 5.4 and B.1. One set of example plots can be found in Figure 5.10. The simulated trait history used to produce Figure 5.10 contained 6 trait gains and we found that the distribution of $E(\mathrm{n}_{01}|\mathbf{y})$ was more peaked around 6 than was the distribution of $E(\mathrm{n}_{01}|\mathbf{y}^{\mathrm{rep}})$.

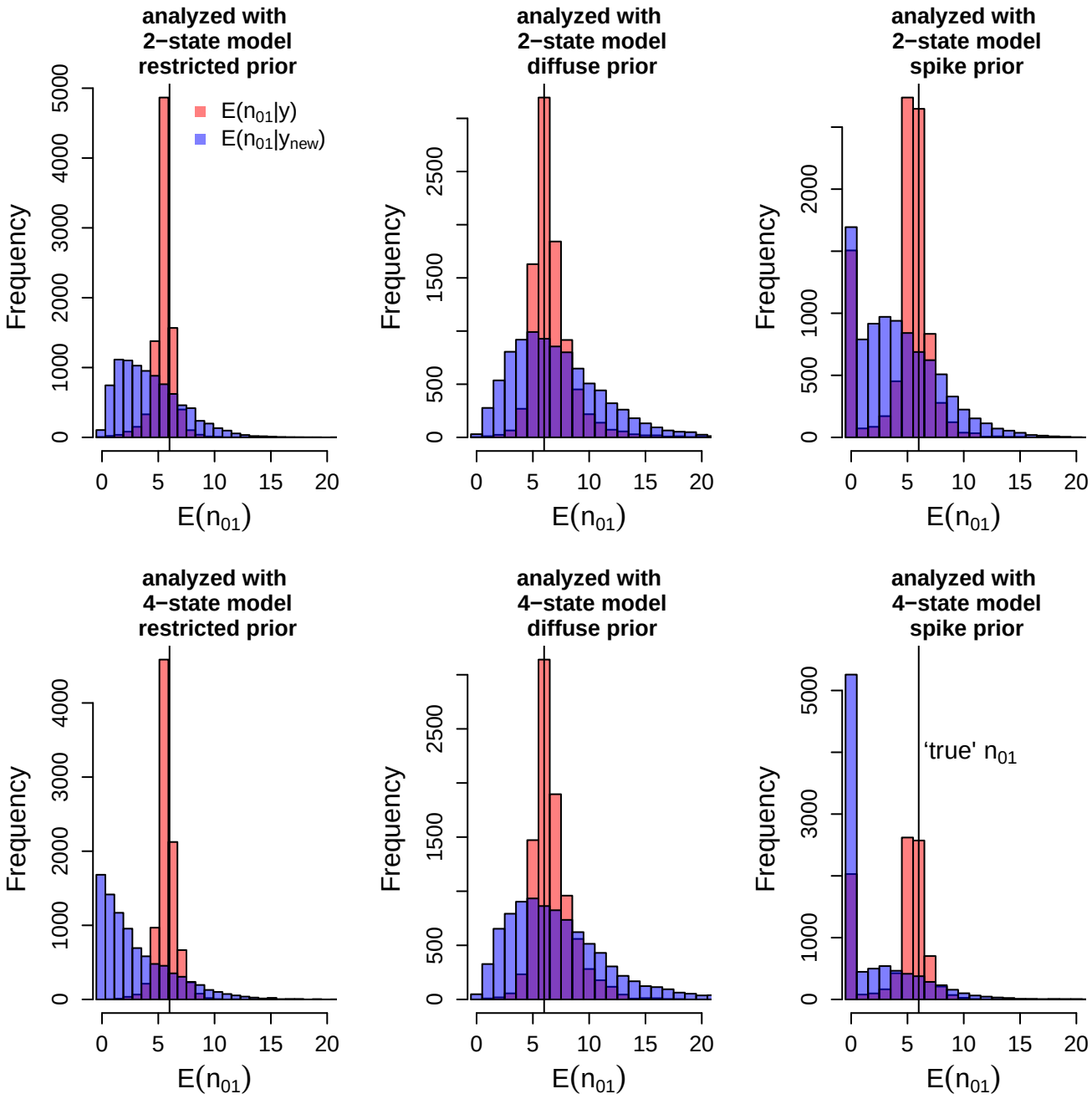

Figure 5.10: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

'true' n_{01}	analysis model	restricted	diffuse	spike
3	2-state	0.62	0.79	0.95
	4-state	0.50	0.77	0.93
1	2-state	0.59	0.81	0.94
	4-state	0.58	0.77	0.93
6	2-state	0.29	0.53	0.46
	4-state	0.18	0.50	0.36
4	2-state	0.25	0.52	0.18
	4-state	0.30	0.49	0.09
0	2-state	0.88	0.56	0.97
	4-state	0.74	0.69	0.96
1	2-state	0.59	0.69	0.95
	4-state	0.54	0.77	0.91
1	2-state	0.33	0.47	0.33
	4-state	0.61	0.46	0.33
1	2-state	0.65	0.74	0.93
	4-state	0.45	0.74	0.89
3	2-state	0.29	0.57	0.32
	4-state	0.19	0.49	0.17
8	2-state	0.10	0.28	0.05
	4-state	0.09	0.33	0.03

Table 5.4: Tail probabilities of our discrepancy for ten simulated data sets on a 70 tip tree. The data sets were simulated from a 4-state model and the 'true' number of trait gains associated with each simulated data set can be found in the first column. Our discrepancy is $D(y, \boldsymbol{\theta}) = E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta})$ and the tail probability is $p_b(y) = p_A(D(y^{\mathrm{rep}}, \theta) \geq D(y, \theta)|H, y)$. We computed tail probabilities for each data set six times, using three different prior sets (restricted, diffuse, and spike) for the 2-state model and for the 4-state model.

Chapter 6

DISCUSSION AND FUTURE WORK

In this thesis we extended and applied the methodology of phylogenetic stochastic mapping. Modern stochastic mapping is built on the framework developed by Nielsen [2002] and later refined by others [Lartillot, 2006, Hobolth, 2008]. All this work relies on exponentiating CTMC rate matrices, a process that is not too problematic when done once or even when done many times for relatively small matrices. Biological state spaces of interest, however, are not always small and researchers need to account for uncertainty in rate matrices. This requires the repeated exponentiation of matrices at each iteration of an MCMC calculation. The algorithmic complexity of matrix exponentiation is $\mathcal{O}(s^3)$ (where s is the size of the CTMC state space) leading to prohibitively long computing times for large state spaces. This led to the work found in chapter 3 where we developed a new phylogenetic stochastic mapping method with algorithmic complexity $\mathcal{O}(s^2)$. The key element of this work was the synthesis of Nielsen [2002]'s Monte Carlo stochastic mapping algorithm and Rao and Teh [2011]'s work on hidden Markov models. In chapter 3 we explored, via simulation, a few scenarios showing when our new MCMC method is faster than matrix exponentiation methods.

Because our new method is matrix-exponentiation free we may be able to map large state spaces that were not computationally tractable up until this point. Future work will involve applying our new stochastic mapping algorithm to some of these large problems including the state spaces of codons, phylogeography, microsatellite evolution, and gene family sizes. Large state spaces can be created from small state spaces by accounting for rate matrix heterogeneity through the use of hidden rates models. We developed a hidden rates model for binary traits and expect the development of new hidden rates models for traits that

take on more than two distinct values. Covarion models for more complex traits will use new parameterizations of the rate matrix which will require the development of more complicated methods.

Modelling multiple traits jointly is another possible application of matrix-exponentiation free stochastic mapping. When traits are assumed to evolve independently they can be modelled separately and the size of the state spaces considered remains relatively small. However, potentially covarying traits should not be modelled separately, they should be modelled jointly. This can easily lead to state spaces that are far too large for existing methods, for example eight binary traits modelled jointly create a state space with 256 different values. Our work will need to be extended to accomodate amalgamations of traits into one larger trait. Parameterizations for such rate matrices will have to be treated carefully. In this thesis we have demonstrated how a hidden rates rate matrix for a binary trait may be constructed but there were alternative modelling choices. Each rate matrix parameterization will require proposal distributions and Metropolis-Hastings acceptance ratios. We worked out the mathematical details for proposals coming from gamma distributions for one particular rate matrix construction. Extending these ideas into new trait domains will be valuable to biologists with varied and interesting questions.

Some of the new rate matrices will be sparse; not all possible transitions will be allowed. In chapter 3 we investigated the interaction between our new method and sparse rate matrices. Matrix exponentiation destroys sparsity in the rate matrix, while our new method is able to make use of sparse matrix operations to gain additional efficiency. We anticipate models that will make use of sparse rate matrices and yet have little prior knowledge concerning which matrix elements should be zero. These scenarios will require sparsity inducing priors such as double exponential, horseshoe, and spike and slab priors [Polson and Scott, 2010, Carvalho et al., 2010, Mitchell and Beauchamp, 1988]. Each of these priors will have different properties and the implications of using one over another will need to be investigated.

In chapter 4 we extended our new stochastic mapping algorithm to incorporate uncertainty in rate matrix parameters. This required the addition of a Metropolis-Hastings ac-

cept/reject step for each rate matrix parameter. We selected an almost conjugate proposal, which led to a well-mixed MCMC and allowed us to create joint posterior distributions over trait histories and rate matrices. We recognized that conditioning on one phylogeny ignores important uncertainty in both tree topology and branch lengths. To address these concerns we showed that our method is capable of averaging substitution histories over a pre-specified set of trees. This pre-specified set of trees generated by DNA data represents a sample from the posterior distribution of topologies and branch lengths. To the degree that our trait of interest does not inform topologies and branch lengths, separating the creation of the tree topology/branch length posterior from the creation of the stochastic mapping/rate matrix posterior is not problematic. Ideally, we will develop techniques to create a single joint posterior for tree topologies, branch lengths, rate matrix parameters, and trait histories. Lartillot [2006] developed a conjugate Gibbs approach for updating branch lengths conditional on trait histories by using an intensive rather than an extensive parameterization of substitution events along each branch. Unfortunately, it is not clear how we could reasonably propose new tree topolgies conditional on trait histories. However, we believe it may be possible to jointly propose new trait histories with new topologies in our stochastic mapping framework but this will require further research. Expanding the scope of the methods in this fashion will entail the incorporation of DNA data directly into the analysis.

We used our new hidden rates model for binary traits to address two stochastic mapping questions of interest in chapter 5. The motivation for developing a hidden rates model came from concerns that a single rate matrix was not a good model of evolution across large phylogenies. There were concerns that the rates of trait transitions could be faster in some parts of the tree and slower in other parts. A hidden rates model allows for this possibility, though it does require the estimation of a greater number of rate matrix parameters.

The first stochastic mapping question we addressed concerned the ancestral reproductive parity mode of the most recent common ancestor of all squamates. Squamates, snakes and lizards, either give live birth or lay eggs. This is the binary trait of interest and many biologists believe that the transition from an egg-laying state (oviparity) to a live birth state

(viviparity) occurs at a higher rate than the transition from viviparity to oviparity. It is generally believed that the most recent common ancestor of Squamata was oviparous. A recent large scale stochastic mapping binary trait analysis of Squamata found support for a viviparous ancestor [Pyron and Burbrink, 2014]. In this thesis we found that accounting for rate matrix heterogeneity across the phylogeny using a hidden rates model produced evidence that strongly supported an egg laying ancestor while a simpler 2-state model found little evidence one way or the other.

The second stochastic mapping question we addressed concerned the development of bioluminescent bacterial photophores in cephalopods. Pankey et al. [2014] found that a binary trait analysis yields evidence that this type of bioluminescence evolved multiple times on the cephalopod phylogeny but there remained concerns that the simple 2-state model of evolution may be inappropriate as it does not account for rate matrix heterogeneity across the tree. Using our hidden rate model we re-analyzed the cephalopod data and found that accounting for rate matrix heterogeneity did not change the overall conclusion. Even when accounting for some rate matrix heterogeneity we found evidence to support the evolution of bioluminescence multiple times on the cephalopod phylogeny.

When considering the use of hidden traits models it is natural to question when the basic binary model should be used and when a model incorporating hidden rates should be used. We investigated the possibility of answering this question in chapter 5 either through the use of posterior predictive distributions or through the use of the deviance information criterion. Both approaches were sensitive to our choice of prior distributions. Using simulated data we found that neither approach was able to consistently distinguish between two state and four state data generating processes. Further work will need to be done to either find a good model selection procedure or to decide that model selection in this context is not feasible.

Our new phylogenetic stochastic mapping algorithm should be a boon for researchers and should enable new analyses that, until now, were too computationally intensive to be attempted.

BIBLIOGRAPHY

Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrov and F. Caski, editors, Proceeding of the Second International Symposium on Information Theory, pages 267–281, 1973.

Jeremy M Beaulieu, Brian C O'Meara, and Michael J Donoghue. Identifying hidden rate changes in the evolution of a binary morphological character: the evolution of plant habit in Campanulid angiosperms. Systematic Biology, 2013.

Julian Besag, Peter Green, David Higdon, and Kerrie Mengersen. Bayesian computation and stochastic systems. Statistical Science, pages 3–41, 1995.

Carlos Carvalho, Nicholas Polson, and James Scott. The horseshoe estimator for sparse signals. Biometrika, page asq017, 2010.

Jason de Koning, Wanjun Gu, and David D Pollock. Rapid likelihood analysis on large phylogenies using partial sampling of substitution histories. Molecular Biology and Evolution, 27(2):249–265, 2010.

Alexei J Drummond, Simon YW Ho, Matthew J Phillips, Andrew Rambaut, et al. Relaxed phylogenetics and dating with confidence. PLoS Biology, 4(5):699, 2006.

Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. Journal of Statistical Software, 40(8):1–18, 2011.

Dirk Eddelbuettel and Conrad Sanderson. RcppArmadillo: Accelerating R with high-performance C++linear algebra. Computational Statistics & Data Analysis, 71:1054–1063, 2014.

James S Farris. Methods for computing wagner trees. Systematic Zoology, pages 83–92, 1970.

Paul Fearnhead and Chris Sherlock. An exact Gibbs sampler for the Markov-modulated Poisson process. Journal of the Royal Statistical Society, Series B, 68(5):767–784, 2006.

Joseph Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution, 17(6):368–376, 1981.

Joseph Felsenstein. Phylogenies and the comparative method. American Naturalist, pages 1–15, 1985.

Richard G FitzJohn. Diversitree: comparative phylogenetic analyses of diversification in R. Methods in Ecology and Evolution, 3(6):1084–1092, 2012.

Nicolas Galtier. Maximum-likelihood phylogenetic analysis under a covarion-like model. Molecular Biology and Evolution, 18(5):866–873, 2001.

Nicolas Galtier, Nicolas Tourasse, and Manolo Gouy. A nonhyperthermophilic common ancestor to extant life forms. Science, 283(5399):220–221, 1999.

Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. Statistica Sinica, 6(4):733–760, 1996.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. Journal of the Royal Statistical Society: Series B, 73:123–214, 2011.

Nick Goldman and Ziheng Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. Molecular Biology and Evolution, 11(5):725–736, 1994.

Petter Guttorp. Stochastic Modeling of Scientific Data. Chapman & Hall, Suffolk, Great Britain, 1995.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57(1):97–109, 1970.

Asger Hobolth. A Markov chain Monte Carlo expectation maximization algorithm for statistical analysis of DNA sequence evolution with neighbor-dependent substitution rates. Journal of Computational and Graphical Statistics, 17(1):138–162, 2008.

Asger Hobolth and Eric A Stone. Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution. The Annals of Applied Statistics, 3(3):1204, 2009.

Chris C Holmes and Leonhard Held. Bayesian auxiliary variable models for binary and multinomial regression. Bayesian Analysis, 1:145–168, 2006.

John P Huelsenbeck and Rasmus Nielsen. Variation in the pattern of nucleotide substitution across sites. Journal of Molecular Evolution, 48(1):86–93, 1999.

John P Huelsenbeck, Rasmus Nielsen, and Jonathan P Bollback. Stochastic mapping of morphological characters. Systematic Biology, 52(2):131–158, April 2003.

Raymond B Huey and Albert F Bennett. Phylogenetic studies of coadaptation: preferred temperatures versus optimal performance temperatures of lizards. Evolution, pages 1098–1115, 1987.

Jan Irvahn and Vladimir N Minin. Phylogenetic stochastic mapping without matrix exponentiation. Journal of Computational Biology, 21(9):676–690, 2014.

Arne Jensen. Markoff chains as an aid in the study of Markoff processes. Scandinavian Actuarial Journal, 1953(sup1):87–91, 1953.

David T Jones, William R Taylor, and Janet M Thornton. The rapid generation of mutation data matrices from protein sequences. Computer Applications in the Biosciences: CABIOS, 8(3):275–282, 1992.

Robert E Kass and Adrian E Raftery. Bayes factors. Journal of the American Statistical Association, 90(430):773–795, 1995.

Benedict King and Michael SY Lee. Ancestral state reconstruction, rate heterogeneity, and the evolution of reptile viviparity. Systematic Biology, page syv005, 2015.

Nicolas Lartillot. Conjugate Gibbs sampling for Bayesian phylogenetic models. Journal of Computational Biology, 13(10):1701–1722, 2006.

Nicolas Lartillot and Hervé Philippe. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Molecular Biology and Evolution, 21(6): 1095–1109, 2004.

Nicolas Lartillot, Nicolas Rodrigue, Daniel Stubbs, and Jacques Richer. PhyloBayes MPI: Phylogenetic reconstruction with infinite mixtures of profiles in a parallel environment. Systematic Biology, 62(4):611–615, 2013.

Philippe Lemey, Andrew Rambaut, Alexei J Drummond, and Marc A Suchard. Bayesian phylogeography finds its roots. PLoS Computational Biology, 5(9):e1000520, 2009.

Philippe Lemey, Vladimir N Minin, Filip Bielejec, Sergei L Kosakovsky Pond, and Marc A Suchard. A counting renaissance: combining stochastic mapping and empirical Bayes to quickly detect amino acid sites under positive selection. Bioinformatics, 28(24):3248–3256, 2012.

Jonathan B Losos. The evolution of form and function: morphology and locomotor performance in West Indian Anolis lizards. Evolution, pages 1189–1203, 1990.

Wayne P Maddison, Peter E Midford, and Sarah P Otto. Estimating a binary character's effect on speciation and extinction. Systematic Biology, 56(5):701–710, 2007.

Emilia P Martins and Theodore Garland Jr. Phylogenetic analyses of the correlated evolution of continuous characters: a simulation study. Evolution, pages 534–557, 1991.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The Journal of Chemical Physics, 21(6):1087–1092, 1953.

Vladimir N Minin and Marc A Suchard. Counting labeled transitions in continuous-time Markov models of evolution. Journal of Mathematical Biology, 56(3):391–412, 2008.

Vladimir N Minin, John D O'Brien, and Arseni Seregin. Imputation estimators partially correct for model misspecification. Statistical Applications in Genetics and Molecular Biology, 10(1):1–24, 2011.

Toby Mitchell and John Beauchamp. Bayesian variable selection in linear regression. Journal of the American Statistical Association, 83(404):1023–1032, 1988.

Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix. SIAM Review, 20(4):801–836, 1978.

Rasmus Nielsen. Mapping mutations on phylogenies. Systematic Biology, 51(5):729–739, 2002.

Rasmus Nielsen and Ziheng Yang. Likelihood models for detecting positively selected amino acid sites and applications to the HIV-1 envelope gene. Genetics, 148(3):929–936, 1998.

John D O'Brien, Vladimir N Minin, and Marc A Suchard. Learning to count: robust estimates for labeled distances between molecular sequences. Molecular Biology and Evolution, 26(4):801–814, 2009.

Mónica AG Otálora, Isabel Martínez, Gregorio Aragón, and M Carmen Molina. Phylogeography and divergence date estimates of a lichen species complex with a disjunct distribution pattern. American Journal of Botany, 97(2):216–223, 2010.

Mark Pagel. Seeking the evolutionary regression coefficient: an analysis of what comparative methods measure. Journal of Theoretical Biology, 164(2):191–205, 1993.

Mark Pagel. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proceedings of the Royal Society of London B: Biological Sciences, 255(1342):37–45, 1994.

Mark Pagel. Inferring the historical patterns of biological evolution. Nature, 401(6756): 877–884, 1999a.

Mark Pagel. The maximum likelihood approach to reconstructing ancestral character states of discrete characters on phylogenies. Systematic Biology, 48(3):612–622, 1999b.

Mark Pagel and Andrew Meade. Bayesian analysis of correlated evolution of discrete characters by reversible-jump Markov chain Monte Carlo. The American Naturalist, 167(6): 808–825, 2006.

Sabrina Pankey, Vladimir N Minin, Greg C Imholte, Marc A Suchard, and Todd H Oakley. Predictable transcriptome evolution in the convergent and complex bioluminescent organs of squid. Proceedings of the National Academy of Sciences, 111(44):E4736–E4742, 2014.

David Penny, Bennet McComish, Michael Charleston, and Michael Hendy. Mathematical elegance with biochemical realism: the covarion model of molecular evolution. Journal of Molecular Evolution, 53(6):711–723, 2001.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. Coda: Convergence diagnosis and output analysis for MCMC. R News, 6(1):7–11, 2006.

Nicholas Polson and James Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. Bayesian Statistics, 9:501–538, 2010.

Alexander Pyron and Frank Burbrink. Early origin of viviparity and multiple reversions to oviparity in squamate reptiles. Ecology letters, 17(1):13–21, 2014.

Alexander Pyron, Frank Burbrink, and John Wiens. A phylogeny and revised classification

of Squamata, including 4161 species of lizards and snakes. BMC Evolutionary Biology, 13 (1):93, 2013.

Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11), Corvallis, Oregon, 2011. AUAI Press.

Fengrong Ren, Hiroshi Tanaka, and Ziheng Yang. An empirical examination of the utility of codon-substitution models in phylogeny reconstruction. Systematic Biology, 54(5):808–818, 2005.

Susanne Renner, L Beenken, Guido Grimm, Alexander Kocyan, and Robert Ricklefs. The evolution of dioecy, heterodichogamy, and labile sex expression in Acer. Evolution, 61(11): 2701–2719, 2007.

James M Robins, Aad van der Vaart, and Valérie Ventura. Asymptotic distribution of p values in composite null models. Journal of the American Statistical Association, 95(452): 1143–1156, 2000.

Nicolas Rodrigue, Nicolas Lartillot, and Hervé Philippe. Bayesian comparisons of codon substitution models. Genetics, 180(3):1579–1591, 2008a.

Nicolas Rodrigue, Hervé Philippe, and Nicolas Lartillot. Uniformization for sampling realizations of Markov processes: applications to Bayesian implementations of codon substitution models. Bioinformatics, 24(1):56–62, 2008b.

Hannes Schabauer, Mario Valle, Cristoph Pacher, Heinz Stockinger, Alexandros Stamatakis, Marc Robinson-Rechavi, Ziheng Yang, and Nicolas Salamin. SlimCodeML: An optimized version of CodeML for the branch-site model. In Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International, pages 706–714, 2012.

Steven L Scott. Bayesian methods for hidden Markov models. Journal of the American Statistical Association, 97(457):337–351, 2002.

Adam Siepel, Katherine S Pollard, and David Haussler. New methods for detecting lineage-specific selection. In Research in Computational Molecular Biology, pages 190–205. Springer, 2006.

Botond Sipos, Tim Massingham, Gregory E Jordan, and Nick Goldman. PhyloSim-Monte Carlo simulation of sequence evolution in the R statistical computing environment. BMC bioinformatics, 12(1):104, 2011.

Adam Skinner. Rate heterogeneity, ancestral character state reconstruction, and the evolution of limb morphology in Lerista (Scincidae, Squamata). Systematic Biology, page syq055, 2010.

Matthew Spencer, Edward Susko, and Andrew J Roger. Modelling prokaryote gene content. Evolutionary Bioinformatics Online, 2:165–186, 2006.

David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 64(4):583–639, 2002.

David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Linde. The deviance information criterion: 12 years on. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 76(3):485–493, 2014.

Jeffrey L Thorne, Hirohisa Kishino, and Ian S Painter. Estimating the rate of evolution of the rate of molecular evolution. Molecular Biology and Evolution, 15(12):1647–1657, 1998.

Erin A Tripp and Paul S Manos. Is floral specialization an evolutionary dead-end? pollination system transitions in Ruellia (Acanthaceae). Evolution, 62(7):1712–1737, 2008.

Mario Valle, Hannes Schabauer, Christoph Pacher, Heinz Stockinger, Alexandros Stamatakis, Marc Robinson-Rechavi, and Nicolas Salamin. Optimization strategies for fast detection of positive selection on phylogenetic trees. Bioinformatics, in press, 2014. doi: 10.1093/bioinformatics/btt760.

Chieh-Hsi Wu and Alexei J Drummond. Joint inference of microsatellite mutation models, population history and genealogies using transdimensional Markov chain Monte Carlo. Genetics, 188:151–164, 2011.

Ziheng Yang. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. Molecular Biology and Evolution, 10(6):1396–1401, 1993.

Emile Zuckerkandl and Linus Pauling. Molecular disease, evolution and genetic heterogeneity. Horizons in Biochemistry, 1962.

Appendix A

FIXED RATE MATRIX SUPPLEMENT

A.1 Posterior Distribution Checks

We present evidence supporting the claim that the stationary distribution of our new MCMC sampler is the posterior distribution, $p(\mathcal{V}, \mathcal{W}|\mathbf{Y})$. This posterior has many aspects that could be examined, but for simplicity we focus on univariate statistics: the amount of time spent in each state and the number of transitions between each pair of states.

We compare the results of five different implementations. The first is a sampler implemented in the diversitree package [FitzJohn, 2012], labeled diversitree or DIV. The second is our version of the same method, labeled EXP. The third is the same method that only exponentiates the rate matrix once, labeled EXP ONCE or ONCE. The fourth is our new method, labeled MCMC. The fifth is a sparse version of our new method, labeled SPARSE or SPA.

We present results for four regimes, two different sizes of state spaces, and two different sets of transition rates. The smaller state space has 4 states and the larger has 20. The lower transition rates correspond to 2 expected transitions per tree and the higher transition rates correspond to 20 expected transitions per tree. In an effort to reduce the number of plots we focus only on states that were observed at the tips of the tree. All four simulated trees had 20 tips.

Our first example used the smaller state size, 4, with the smaller number of expected transitions per tree, 2. A random simulation resulted in two unique tip states, states 1 and 3. For each method we produced 100,000 state history samples. Figure A.1 contains plots of four univariate statistics pulled from the posterior distributions. All five implementations produced the same results.

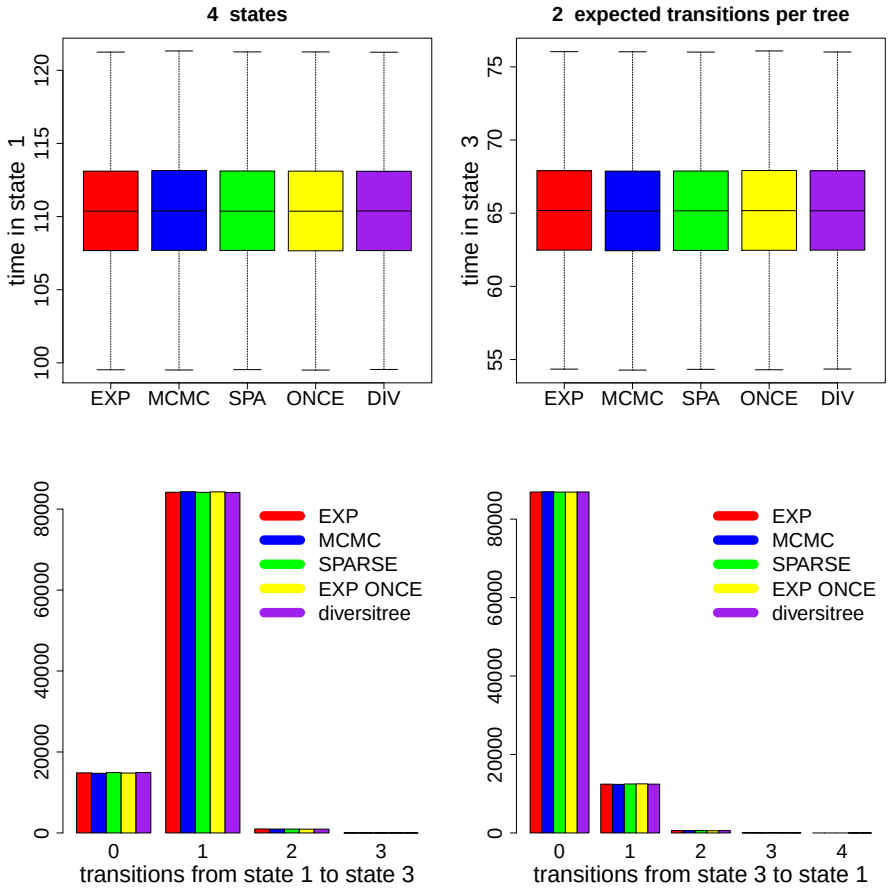

Figure A.1: Univariate summaries for five implementations of state history sampling of a 20 tip tree. There were 4 states and 2 expected transitions per tree. The top plots contain boxplots illustrating the distribution of the amount of time spent in state 1 and state 3. Outliers were not included though all five implementations showed the same outlier behavior. The bottom plots contain histograms illustrating the posterior distribution of the number of transitions between state 1 and state 3.

Our second example used the smaller state size, 4, with the larger number of expected transitions per tree, 20. A random simulation resulted in three unique tip states, states 1, 2, and 4. Figure A.2 contains four boxplots pulled from the posterior distributions. Figure A.3 contains six histograms pulled from the posterior distributions.

Our third example used the larger state size, 20, with the smaller number of expected

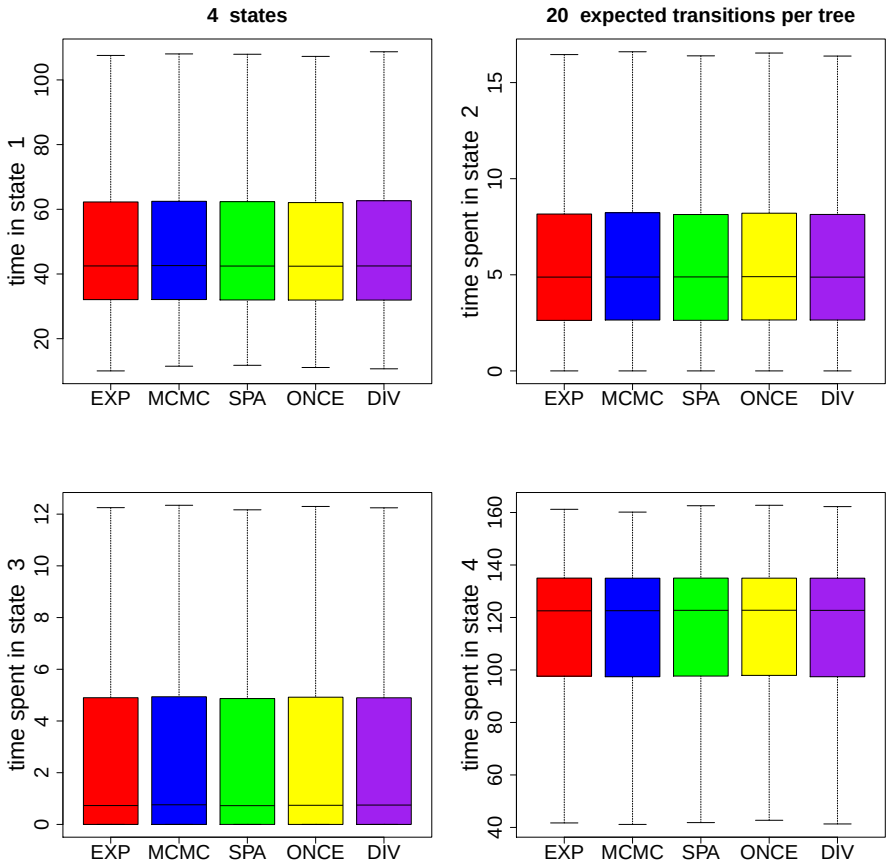

Figure A.2: Boxplots illustrating the posterior distribution of the amount of time spent in each state. Outliers were not included though all five implementations showed the same outlier behavior. There were 4 states and 20 expected transitions per tree.

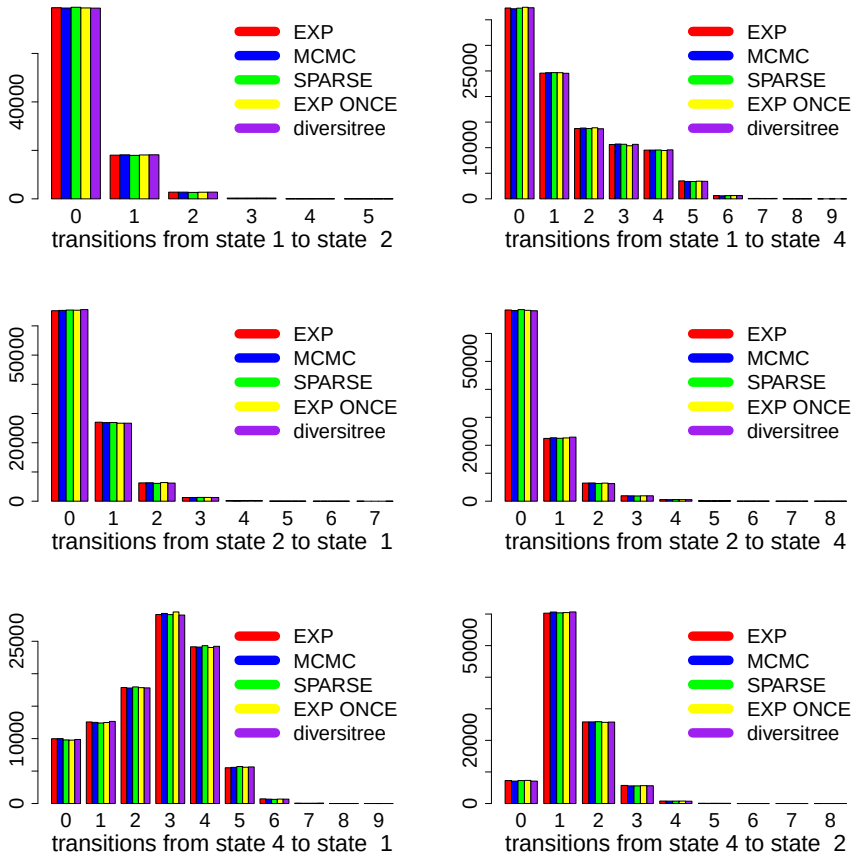

Figure A.3: Histograms illustrating the posterior distribution of the number of transitions between states 1, 2, and 4. There were 4 states and 20 expected transitions per tree.

transitions per tree, 2. A random simulation resulted in two unique tip states, states 1 and 5. Figure A.4 contains plots of four univariate statistics pulled from the posterior distributions

Our fourth example used the larger state size, 20, with the larger number of expected transitions per tree, 20. A random simulation resulted in six unique tip states, states 1, 4, 8, 10, 12, and 15. Figure A.5 contains six boxplots pulled from the posterior distributions. Figure A.6 contains six histograms pulled from the posterior distributions.

A.2 Tuning Parameter

One state space of interest in molecular evolution is the amino acid state space. Jones et al. [1992] proposed a rate matrix for an amino acid CTMC substitution model, called JTT. Figure A.7 contains timing results for this rate matrix and a tree with 40 tips. When our MCMC approach used an appropriately tuned value of Ω we saw slightly faster running times as compared to the matrix exponentiation approach. We examined other scenarios for the JTT rate matrix in which we saw faster running times with the matrix exponentiation approach.

A.3 Convergence

Our MCMC sampler seems to converge to stationarity quickly. Figure A.8 shows two convergence plots, one for fast evolution and one for slow evolution. In both cases we started the chain with an augmented substitution history containing one transition in the middle of each branch leading to a tip whose state was different from an arbitrarily chosen root state. In the case of slow evolution this substitution history was a poor starting point but the log likelihood of the chain appeared to achieve stationarity quickly. In both cases, the tree had 50 tips and the size of the state space was 10.

A.4 Sparsity

Our MCMC method scales well with the size of the state space even when state space sizes exceed 100. Figure A.9 shows timing results for state space sizes going out to 300. We show

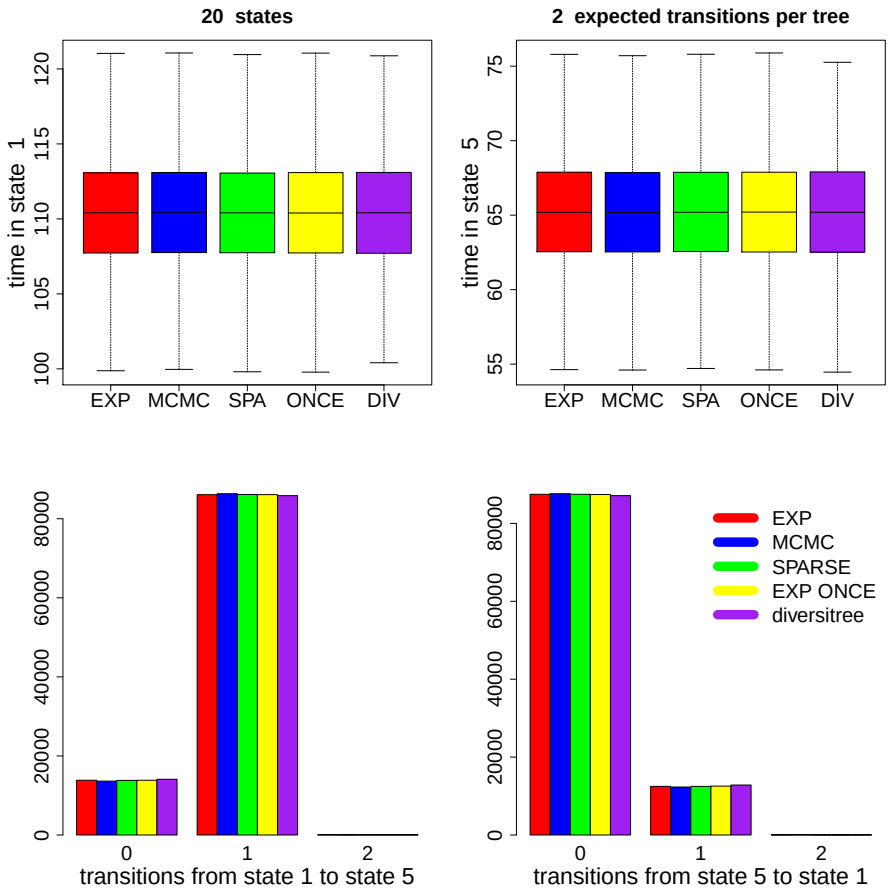

Figure A.4: Univariate summaries for 5 implementations of state history sampling of a 20 tip tree. There were 20 states and 2 expected transitions per tree. The top plots contain boxplots illustrating the posterior distribution of the amount of time spent in state 1 and state 5. Outliers were not included though all five implementations showed the same outlier behavior. The bottom plots contain histograms illustrating the posterior distribution of the number of transitions between state 1 and state 5.

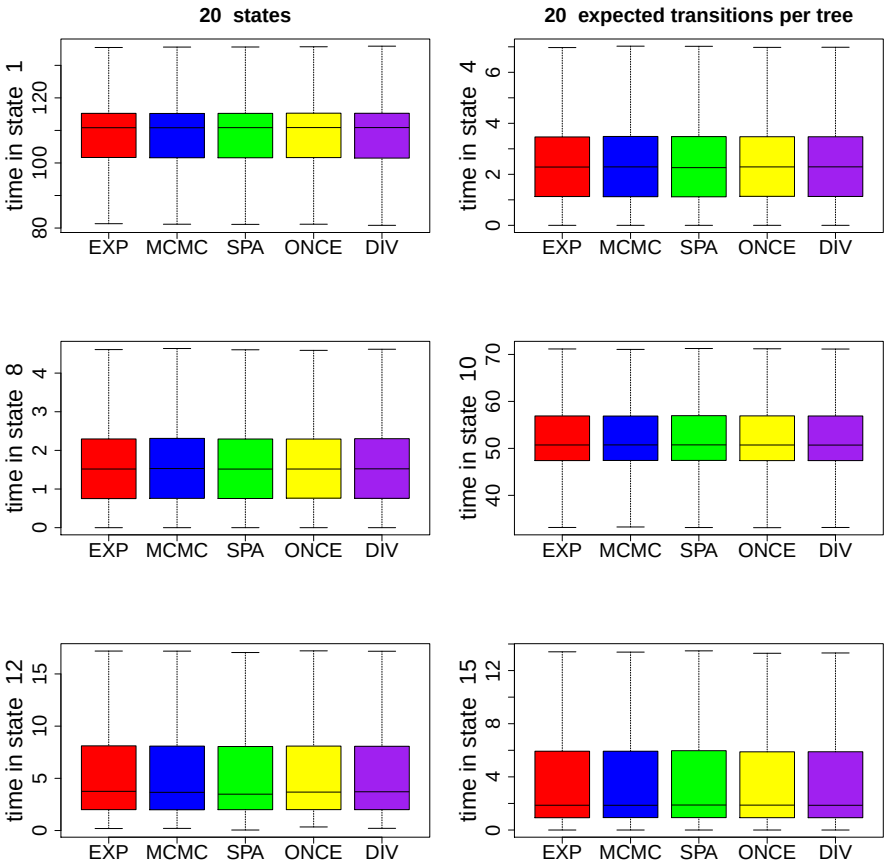

Figure A.5: Boxplots illustrating the posterior distribution of the amount of time spent in each tip state. Outliers were not included though all five implementations showed the same outlier behavior. There were 20 states and 20 expected transitions per tree.

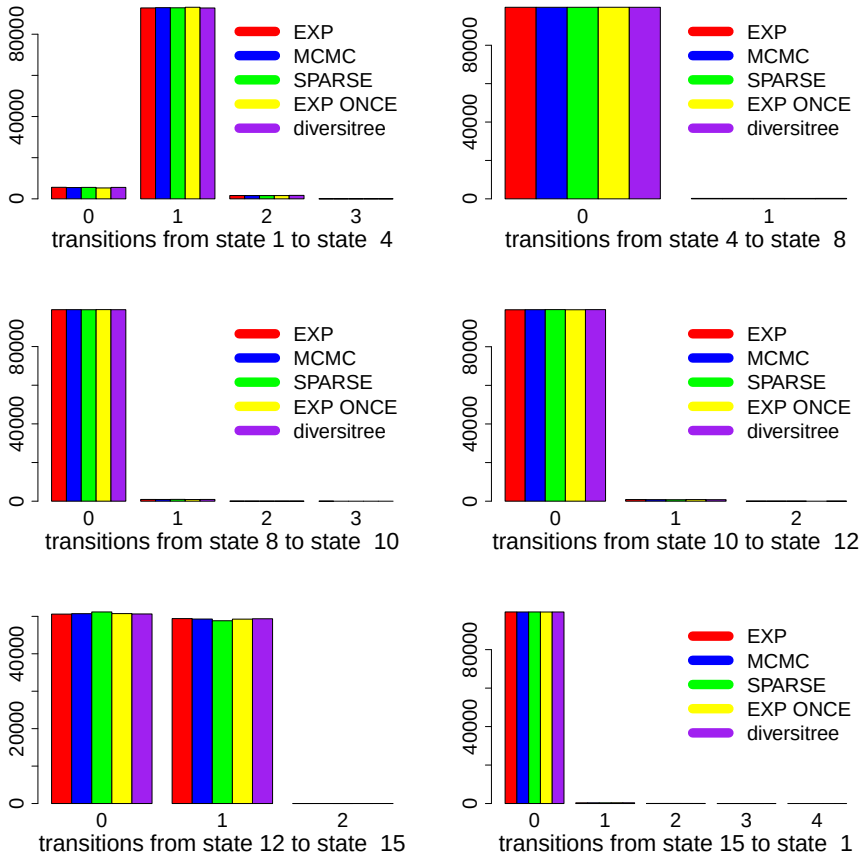

Figure A.6: Histograms illustrating the posterior distribution of the number of transitions between a subset of the tip states. There were 20 states and 20 expected transitions per tree.

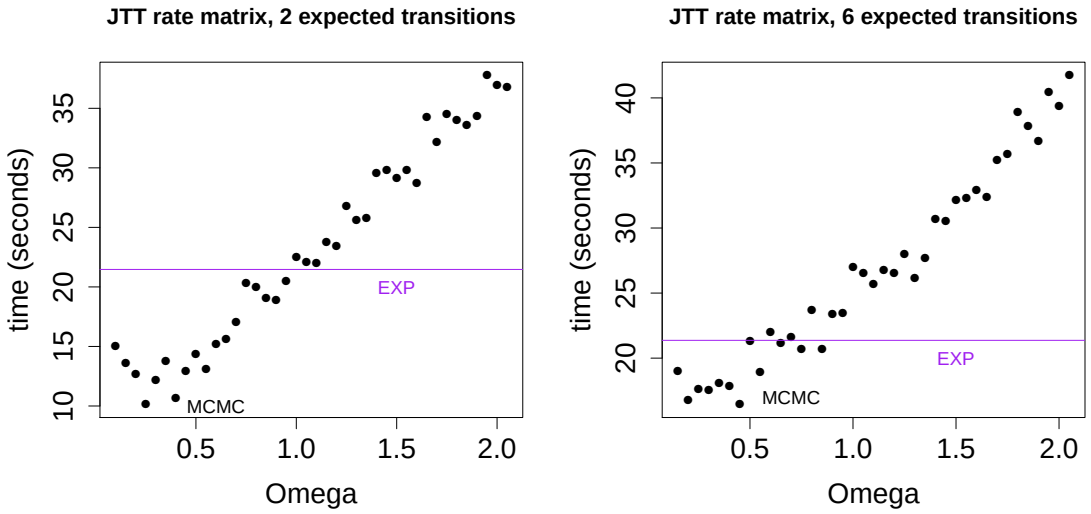

Figure A.7: Time to obtain 10,000 effective samples as a function of the dominating Poisson process rate, Ω, for the JTT amino acid rate matrix as found in the phylosim R package. Results for our MCMC sampler are shown in black. Timing results for the matrix exponentiation method are represented by a purple horizontal line because the matrix exponentiation result does not vary as a function of Ω. The randomly generated tree had 40 tips. The JTT rate matrix was scaled to produce 2 expected transtions in the left hand plot. The JTT rate matrix was scaled to produce 6 expected transtions in the right hand plot.

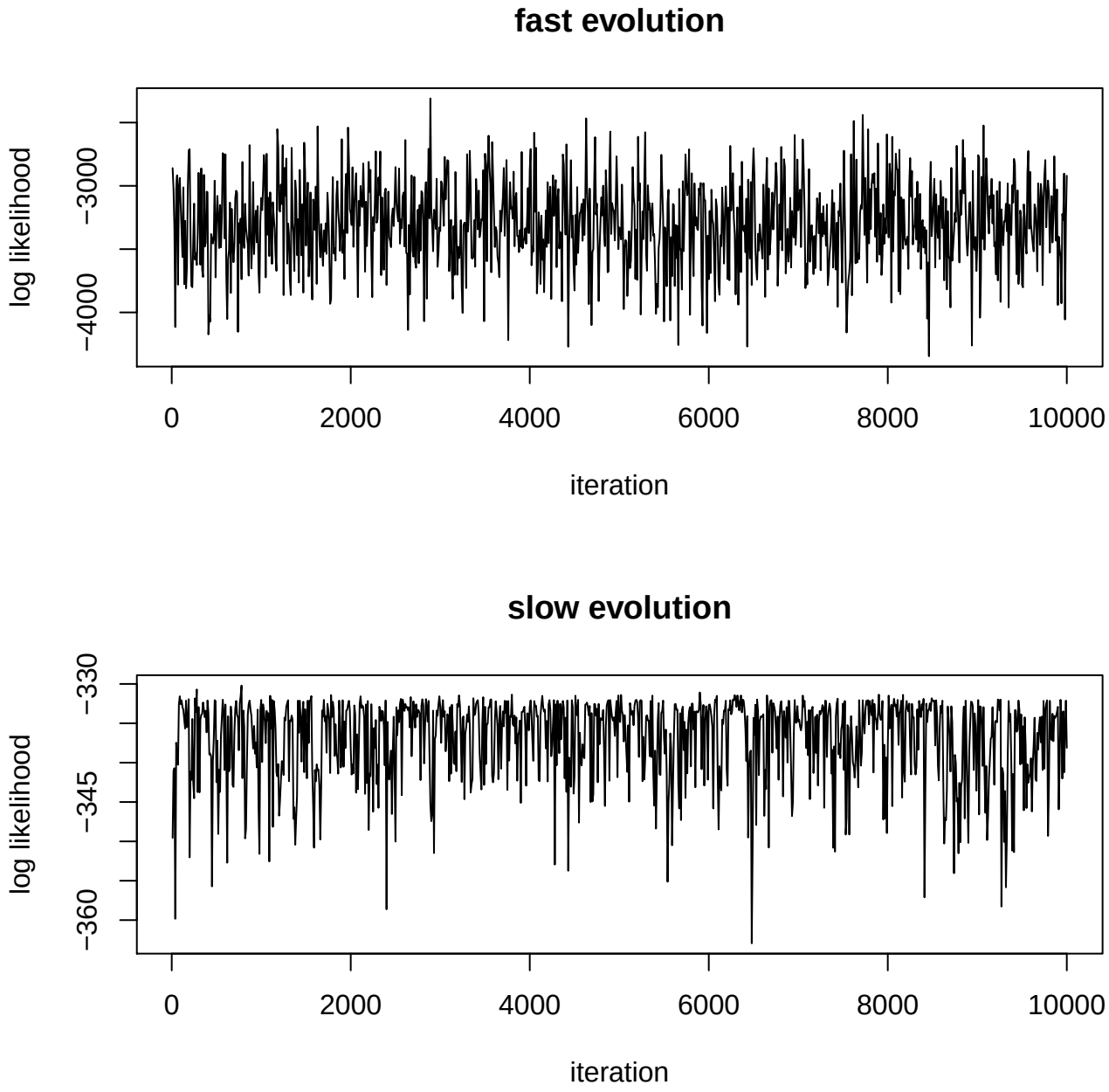

Figure A.8: MCMC trace plots. We show the log density of substitution histories for two MCMC chains at every tenth iteration. The top plot shows results for a trait that evolved quickly (with 6 expected substitutions). The bottom plot shows results for a trait that evolved slowly (with 2 expected substitutions). In both cases, the tree had 50 tips and the size of the state space was 10.

results for our MCMC method and a sparse version of our MCMC method using tridiagonal rate matrices.

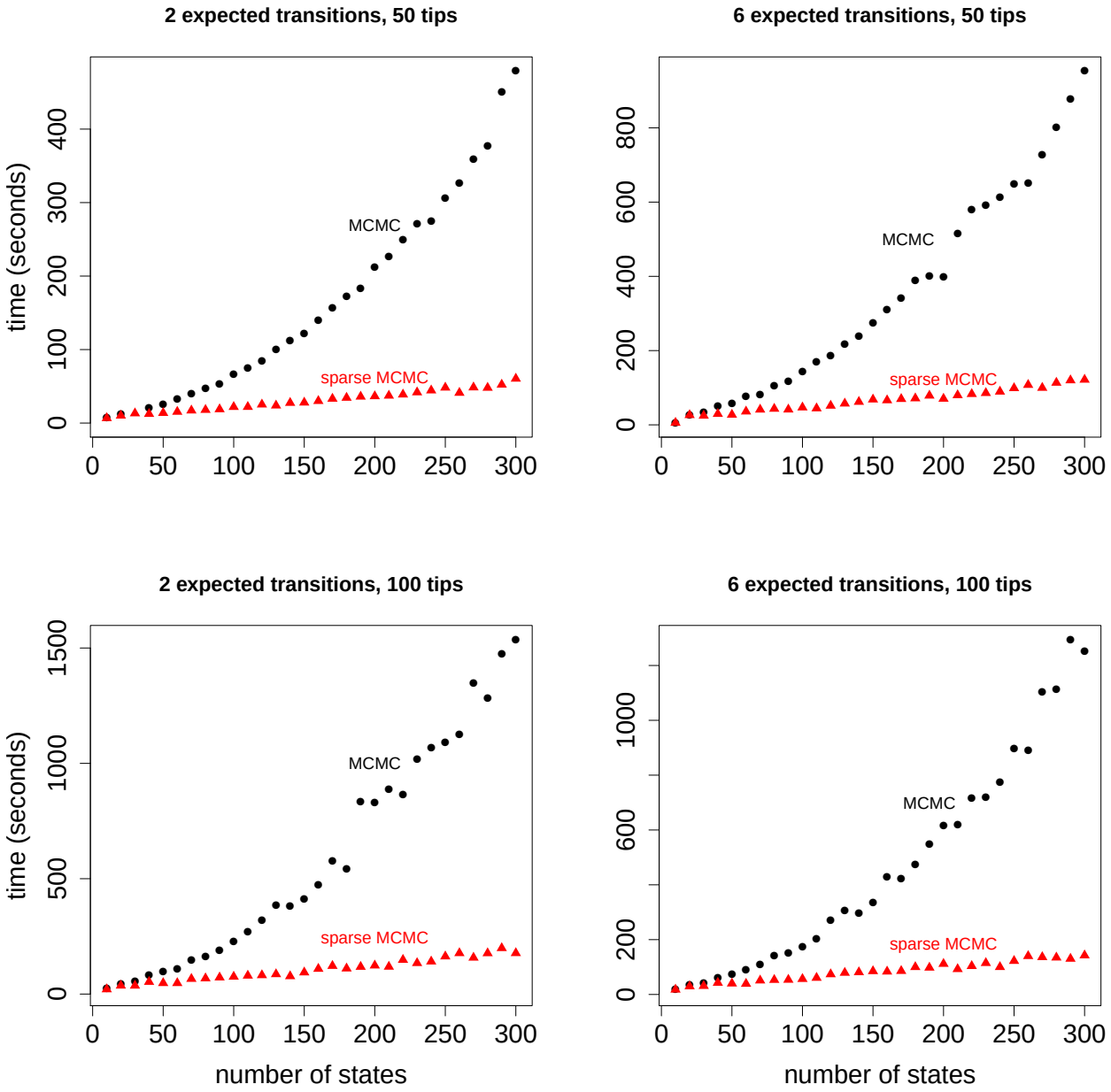

Figure A.9: State space effect for a tridiagonal rate matrix. All four plots show the amount of time required to obtain 10,000 effective samples as a function of the size of the state space for two methods, our MCMC sampler in black circles and a sparse version of our MCMC sampler in red triangles. The two plots in the top row show results for a randomly generated tree with 50 tips. The two plots in the bottom row show results for a randomly generated tree with 100 tips. The two plots in the left column show results for a rate matrix that was scaled to produce 2 expected transitions while the two plots in the right column show results for a rate matrix that was scaled to produce 6 expected transitions.

Appendix B

POSTERIOR PREDICTIVE PLOTS

Data simulated from a 4-state model on a 70 tip tree

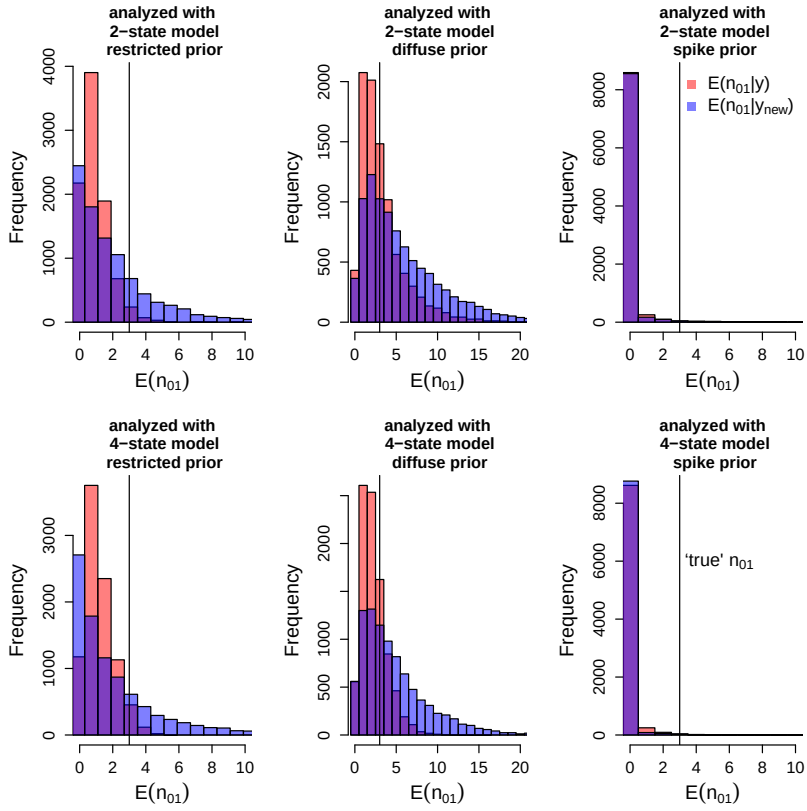

Figure B.1: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

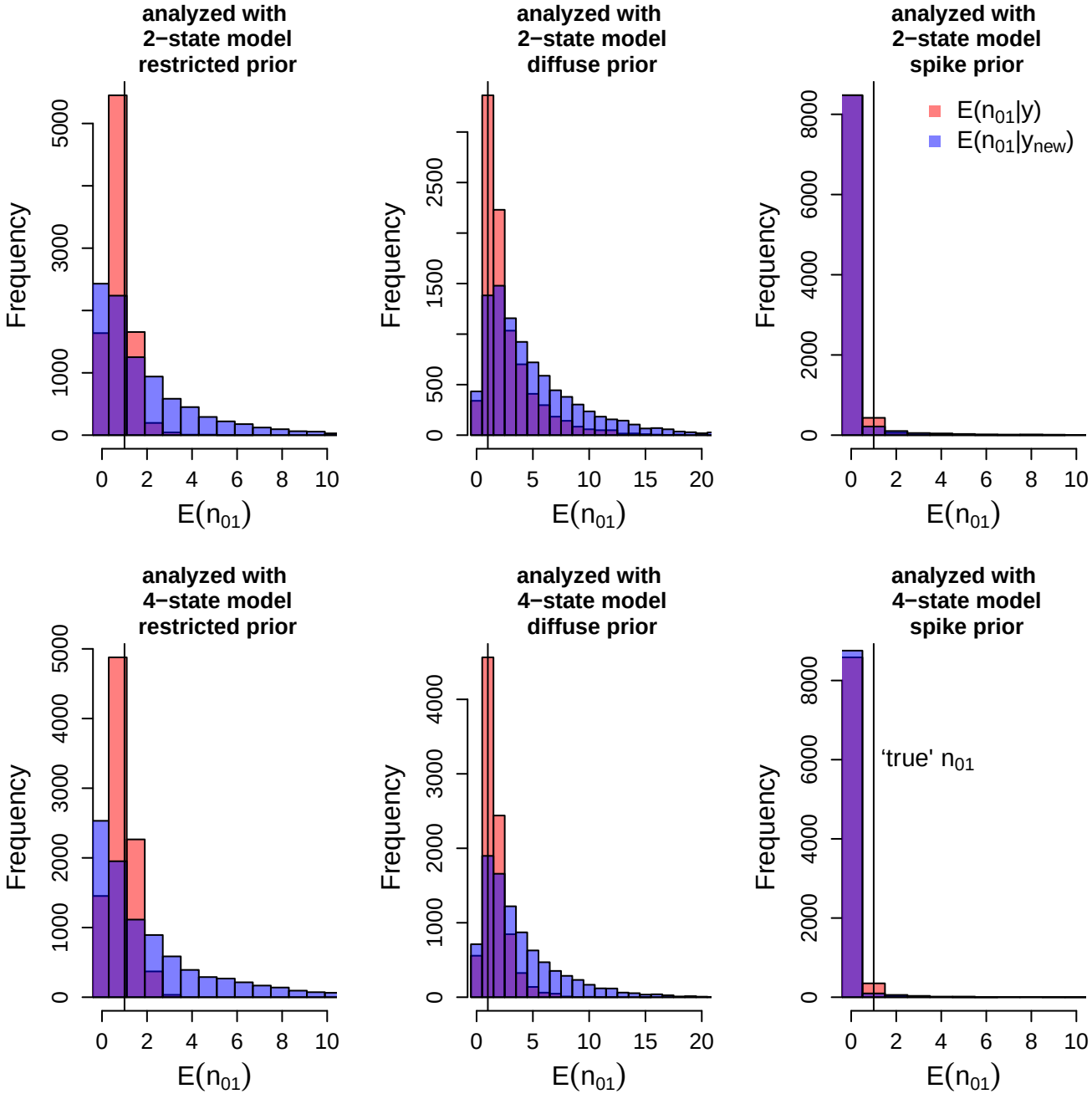

Figure B.2: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

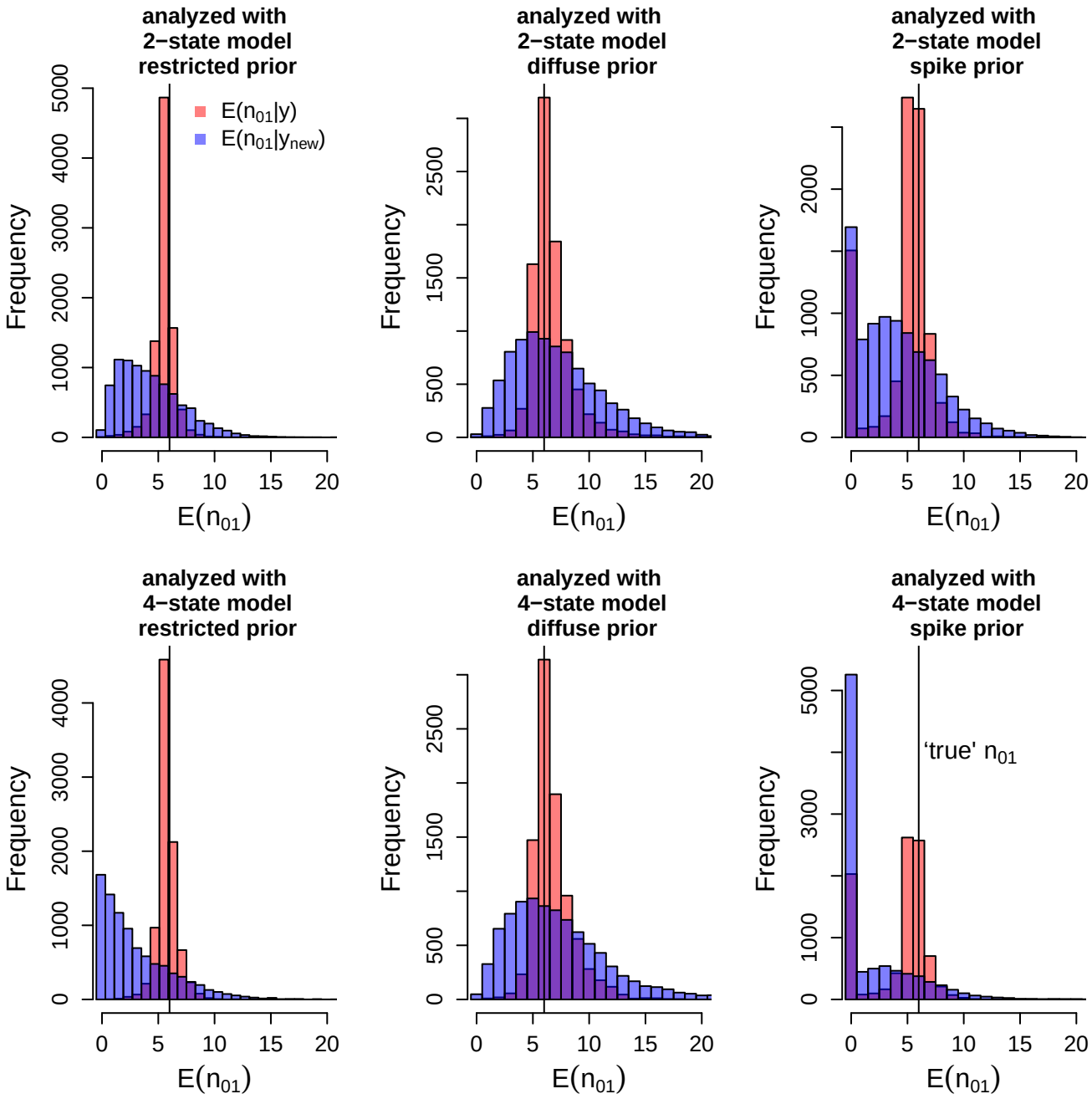

Figure B.3: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

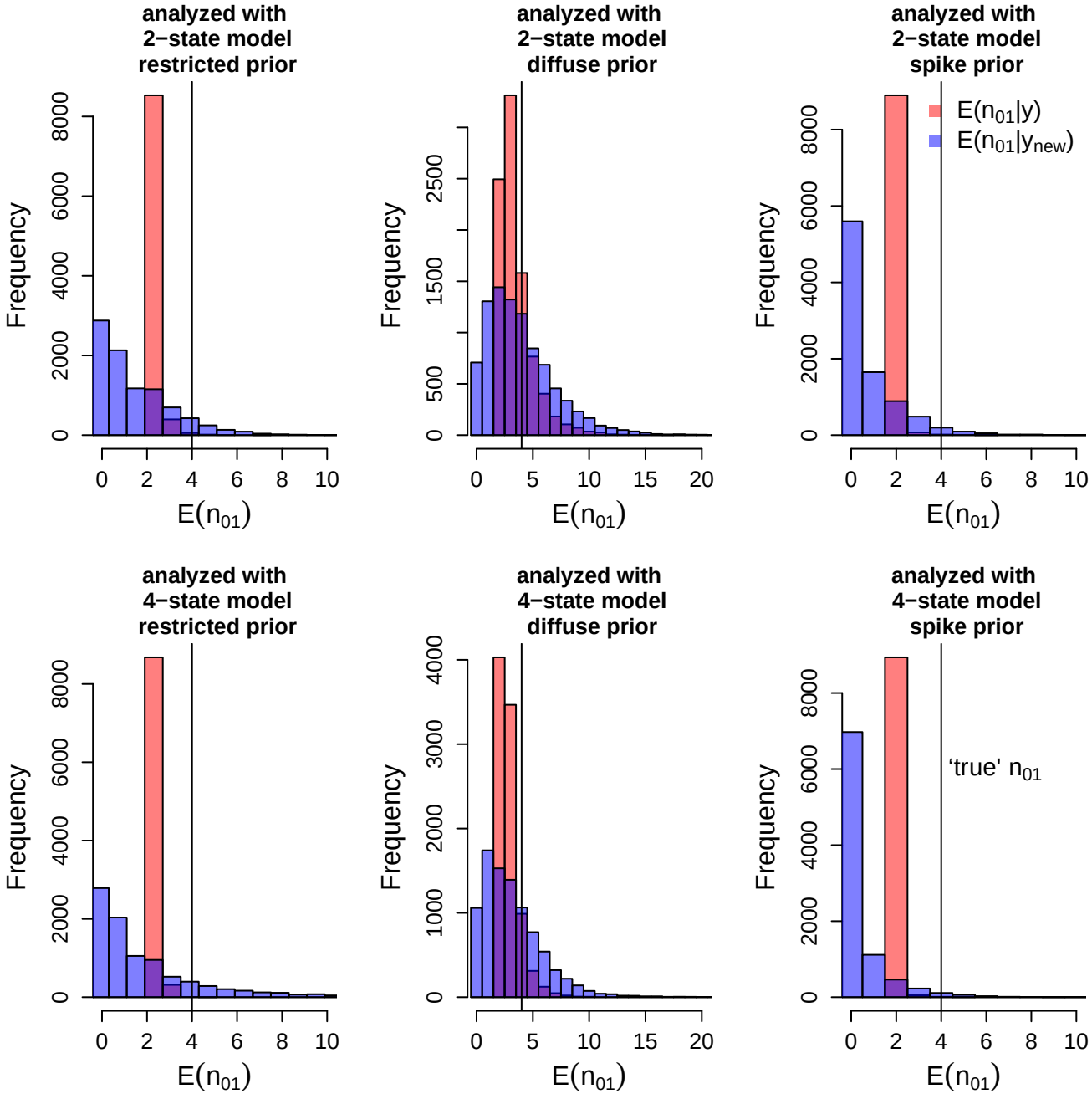

Figure B.4: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

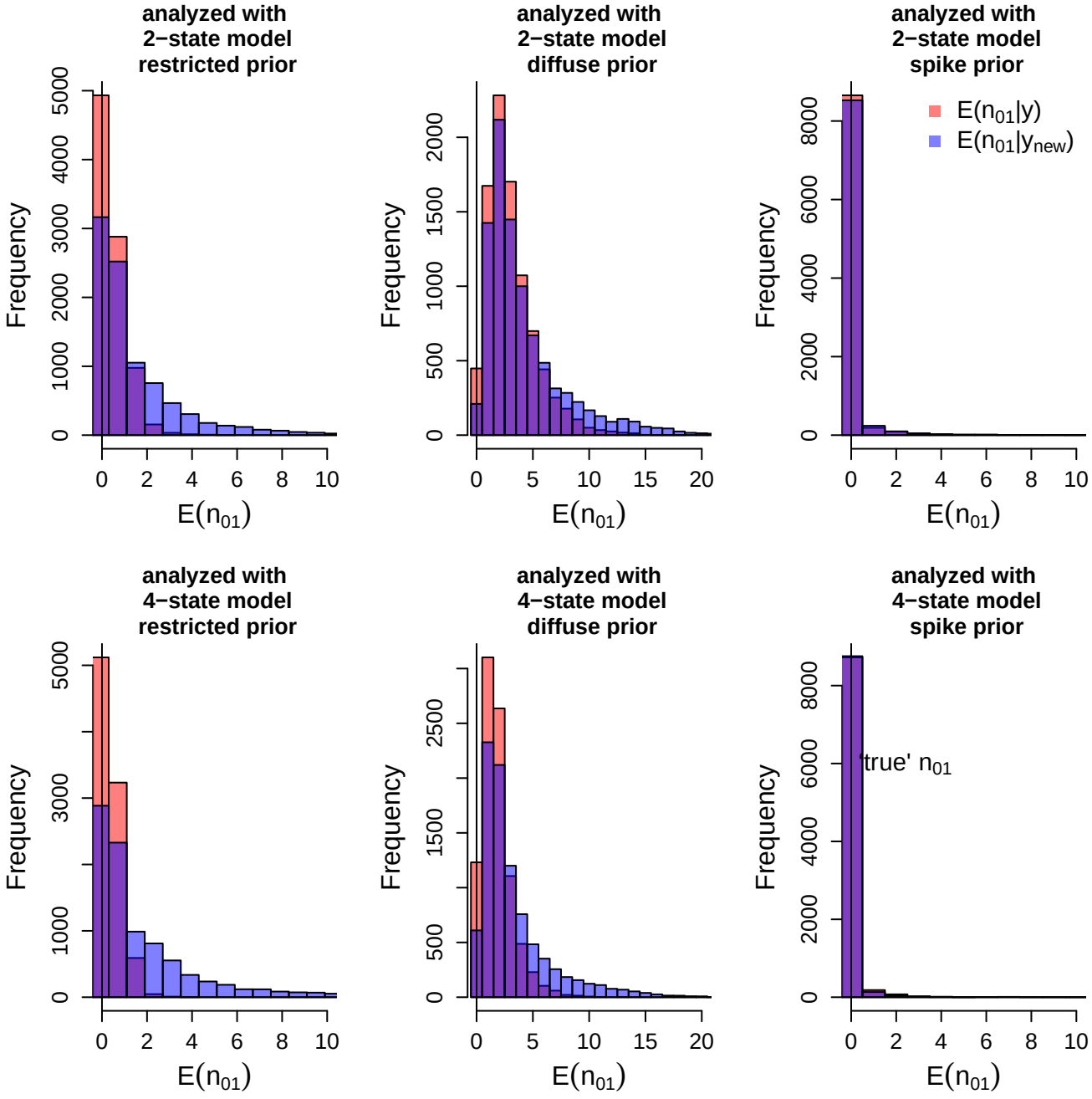

Figure B.5: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

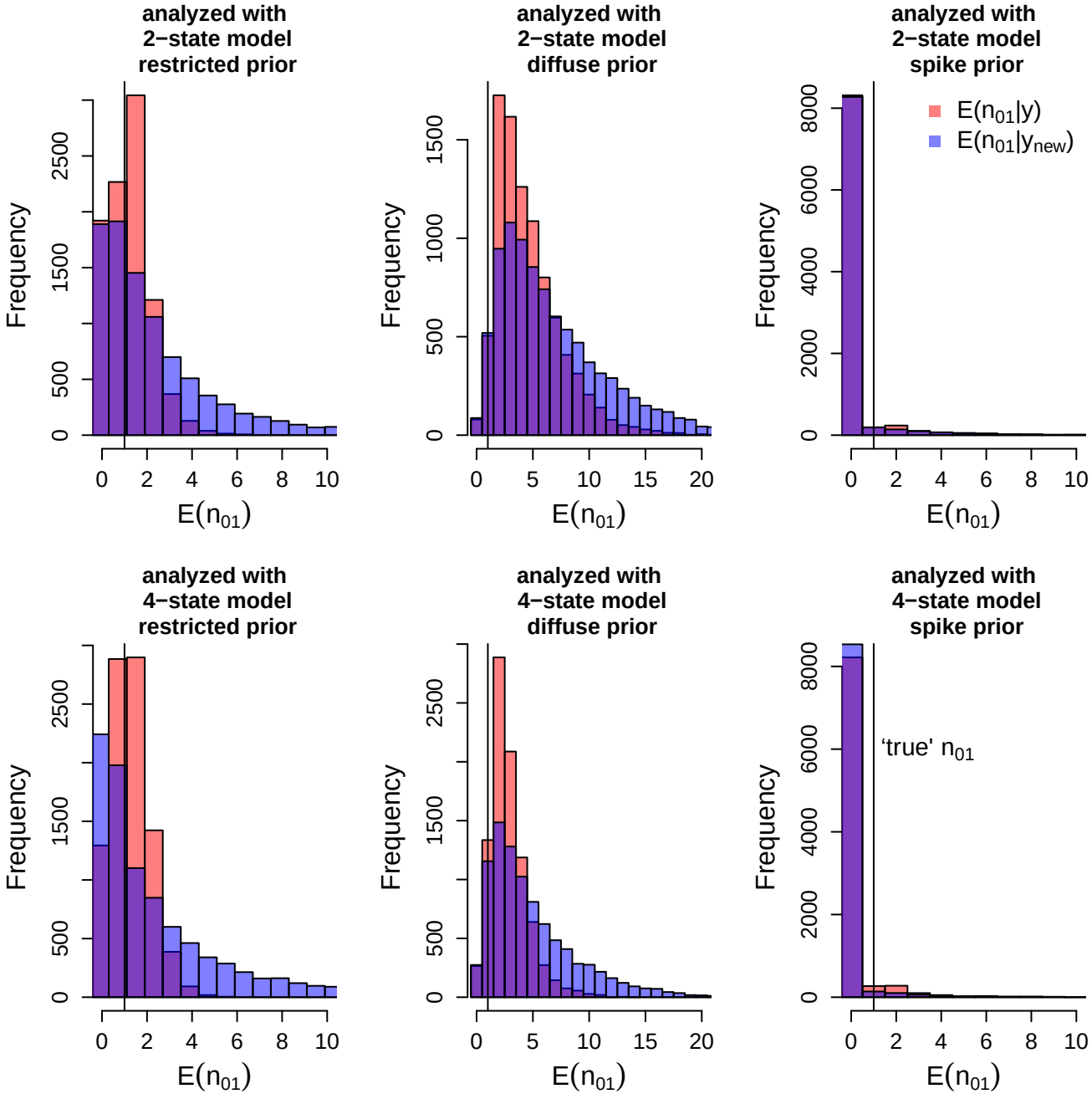

Figure B.6: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

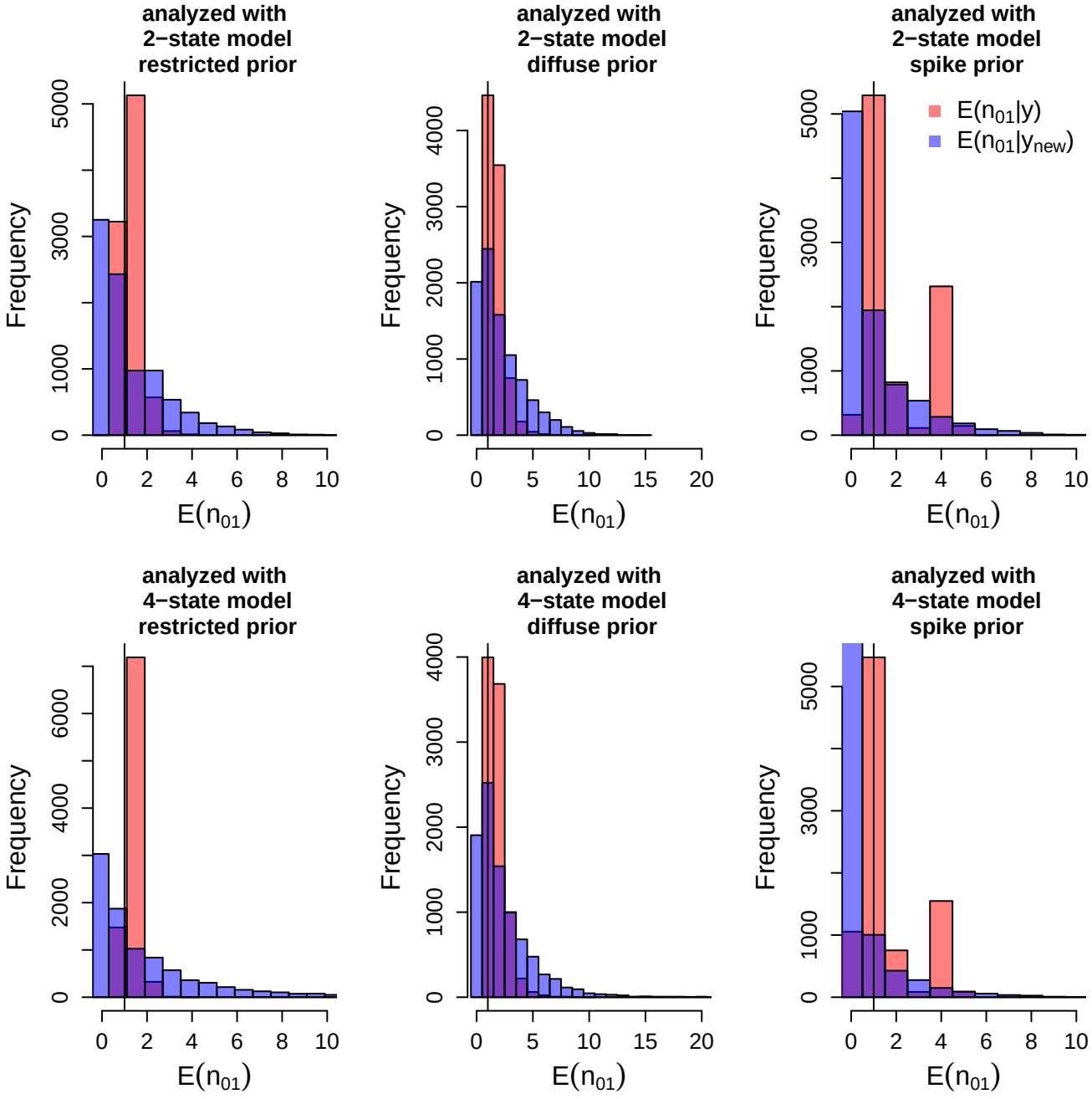

Figure B.7: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

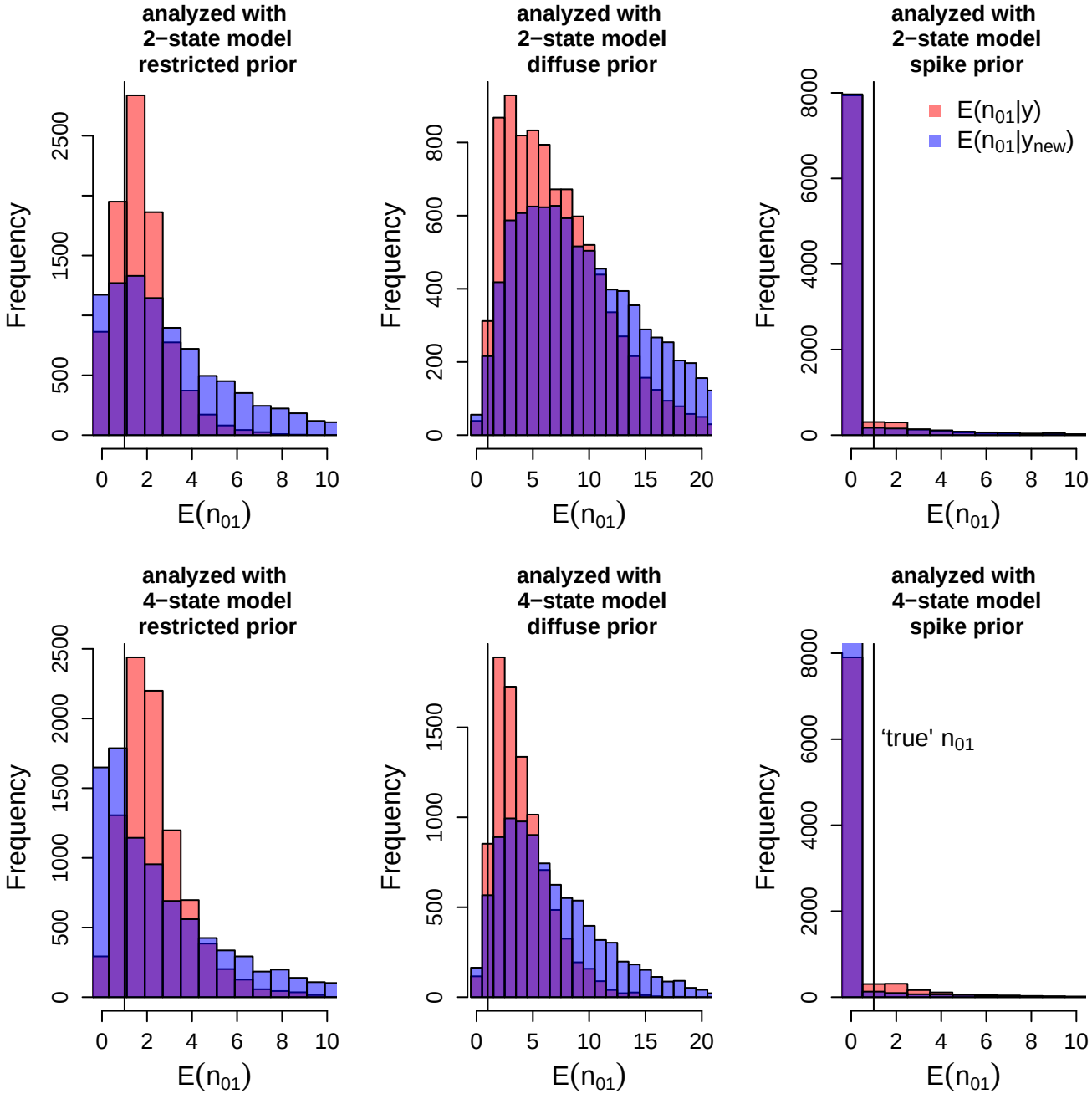

Figure B.8: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

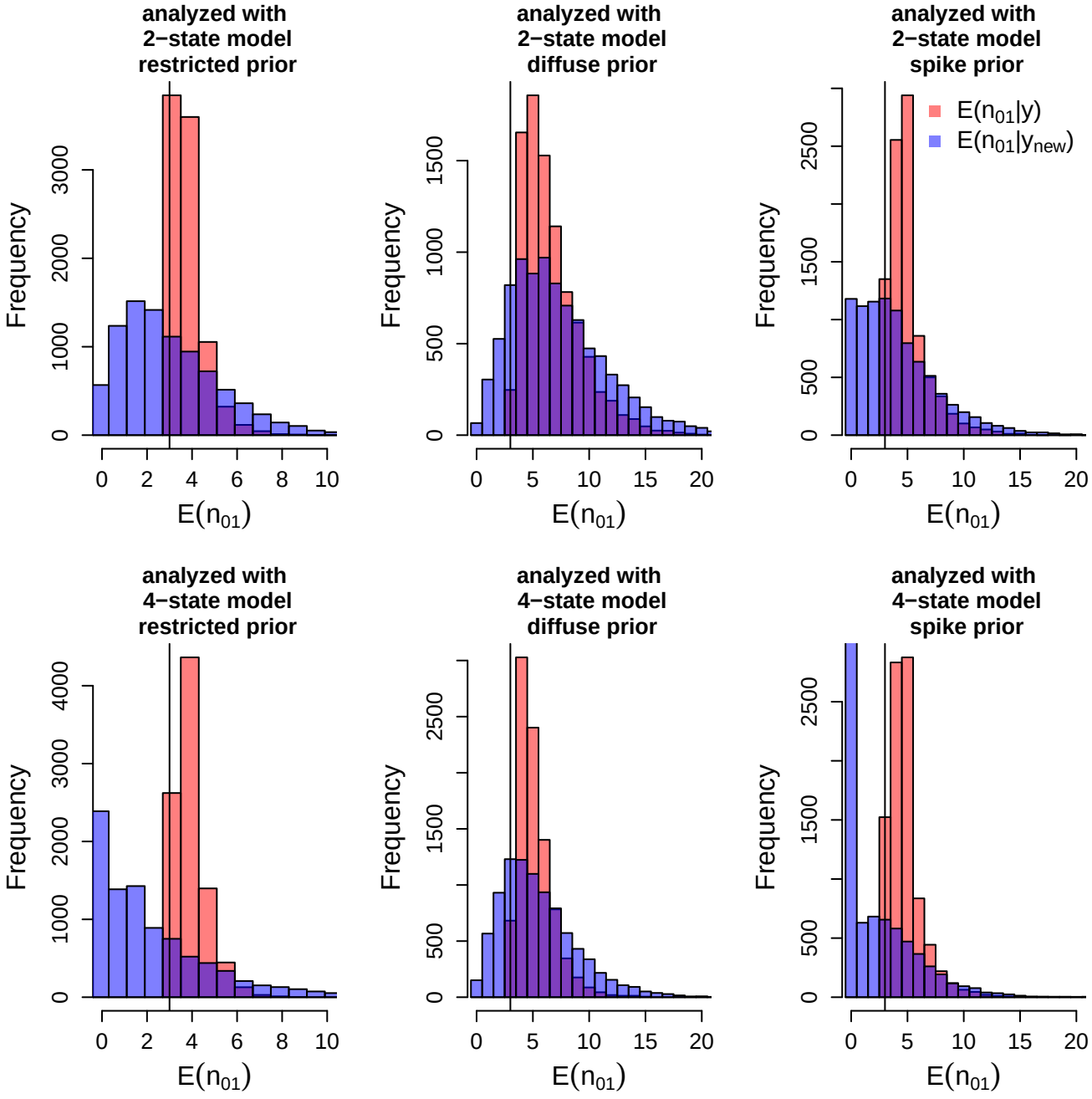

Figure B.9: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

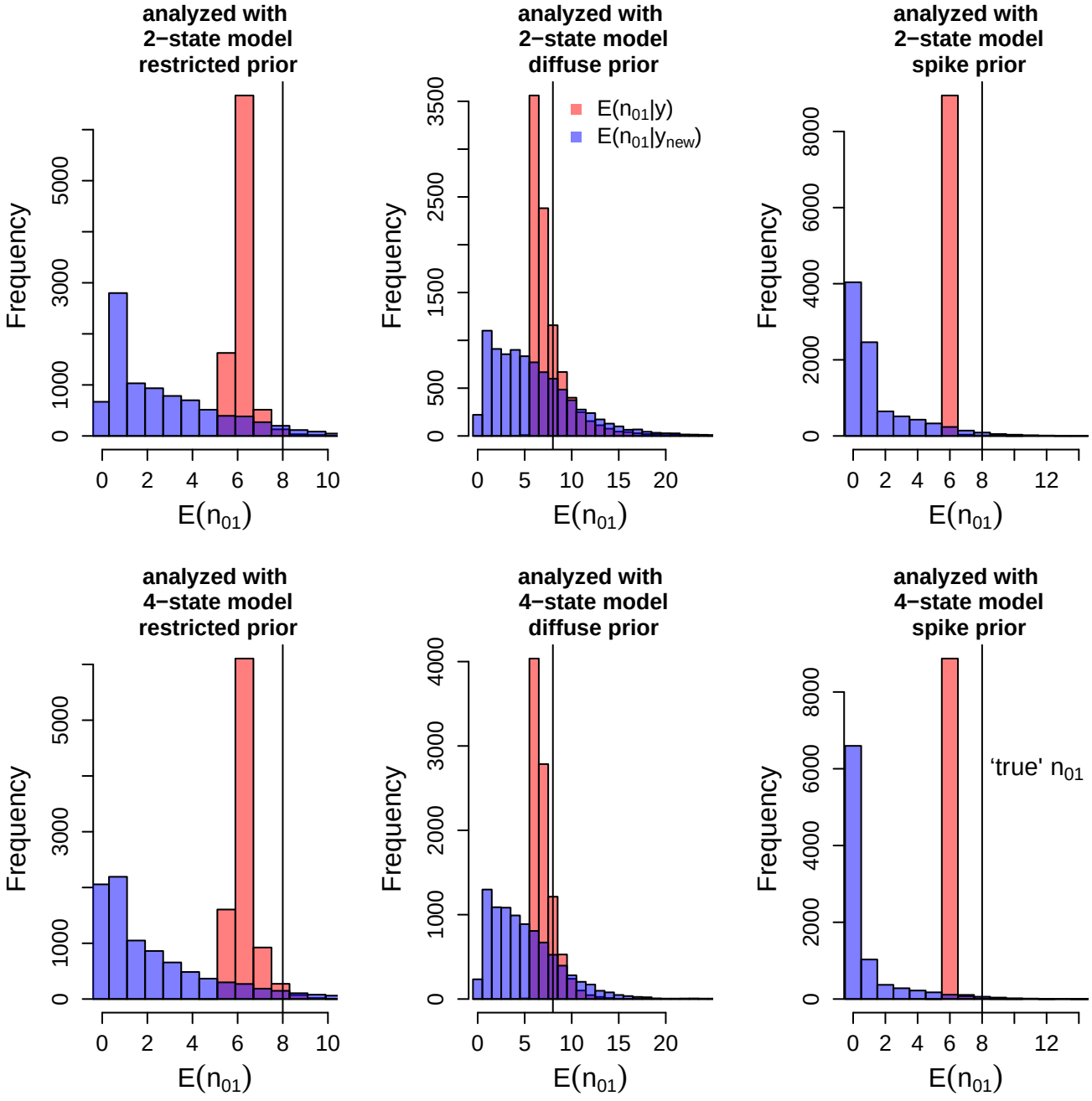

Figure B.10: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 4-state model and the 'true' number of trait gains was 8. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

Data simulated from a 2-state model on a 70 tip tree

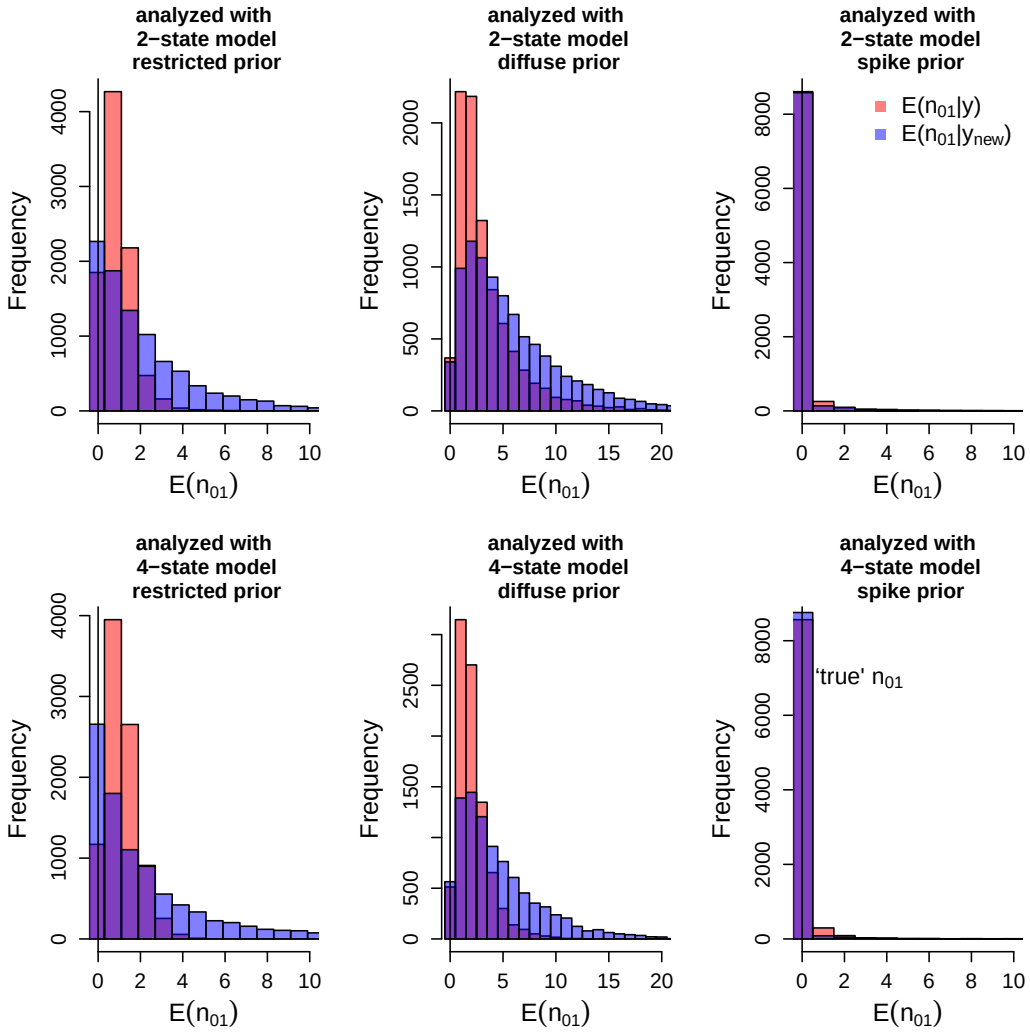

Figure B.11: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

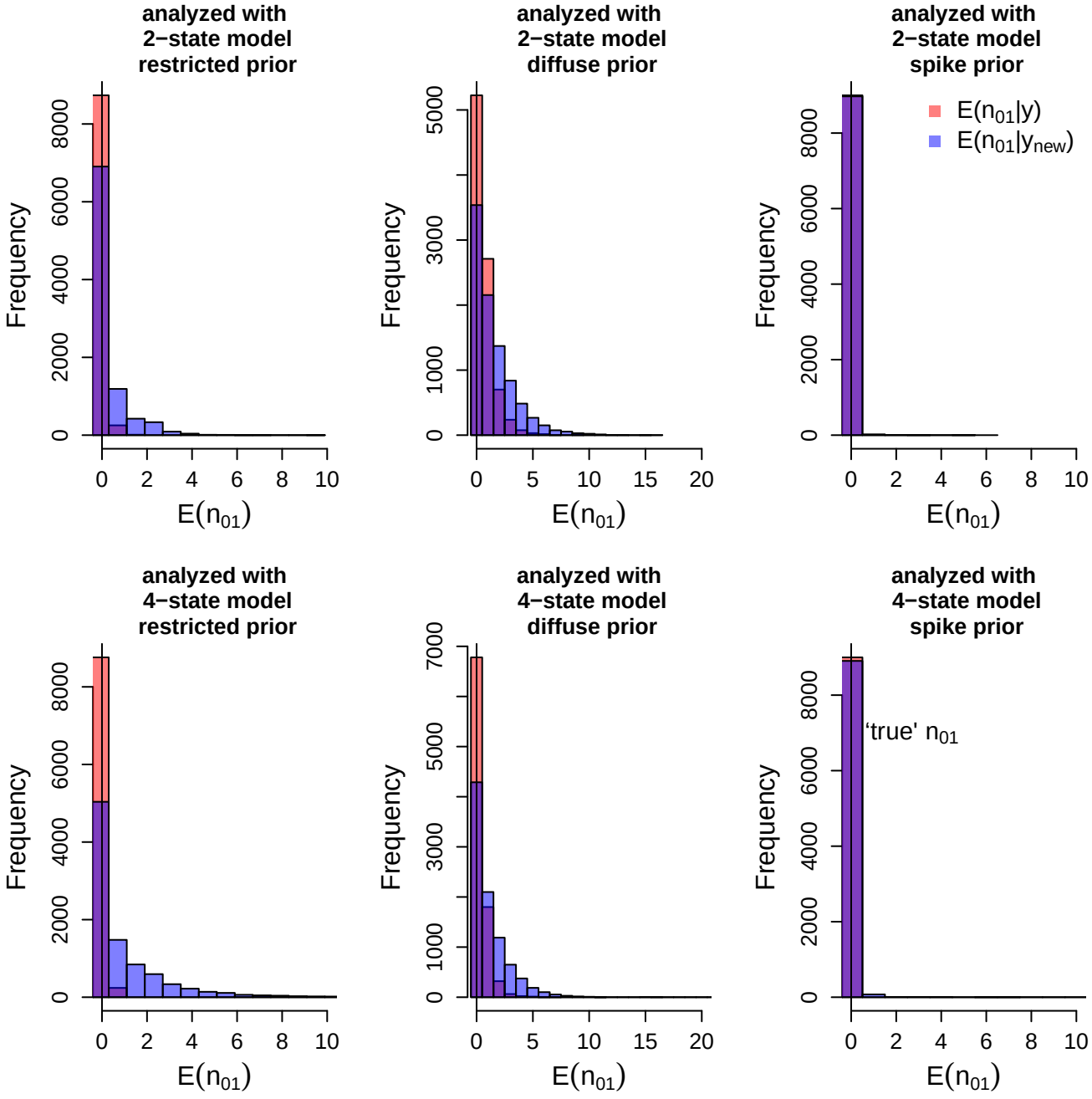

Figure B.12: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

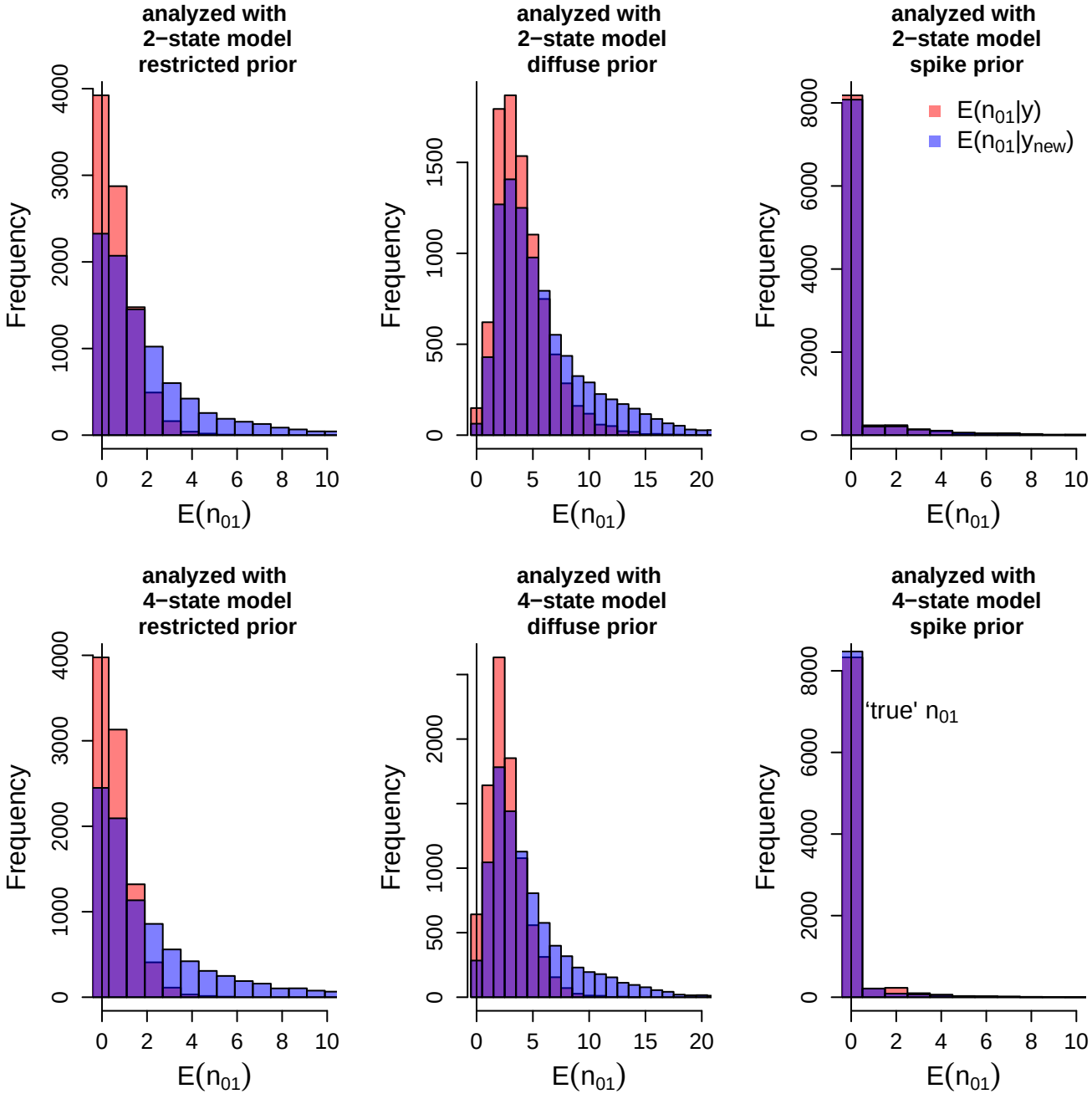

Figure B.13: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

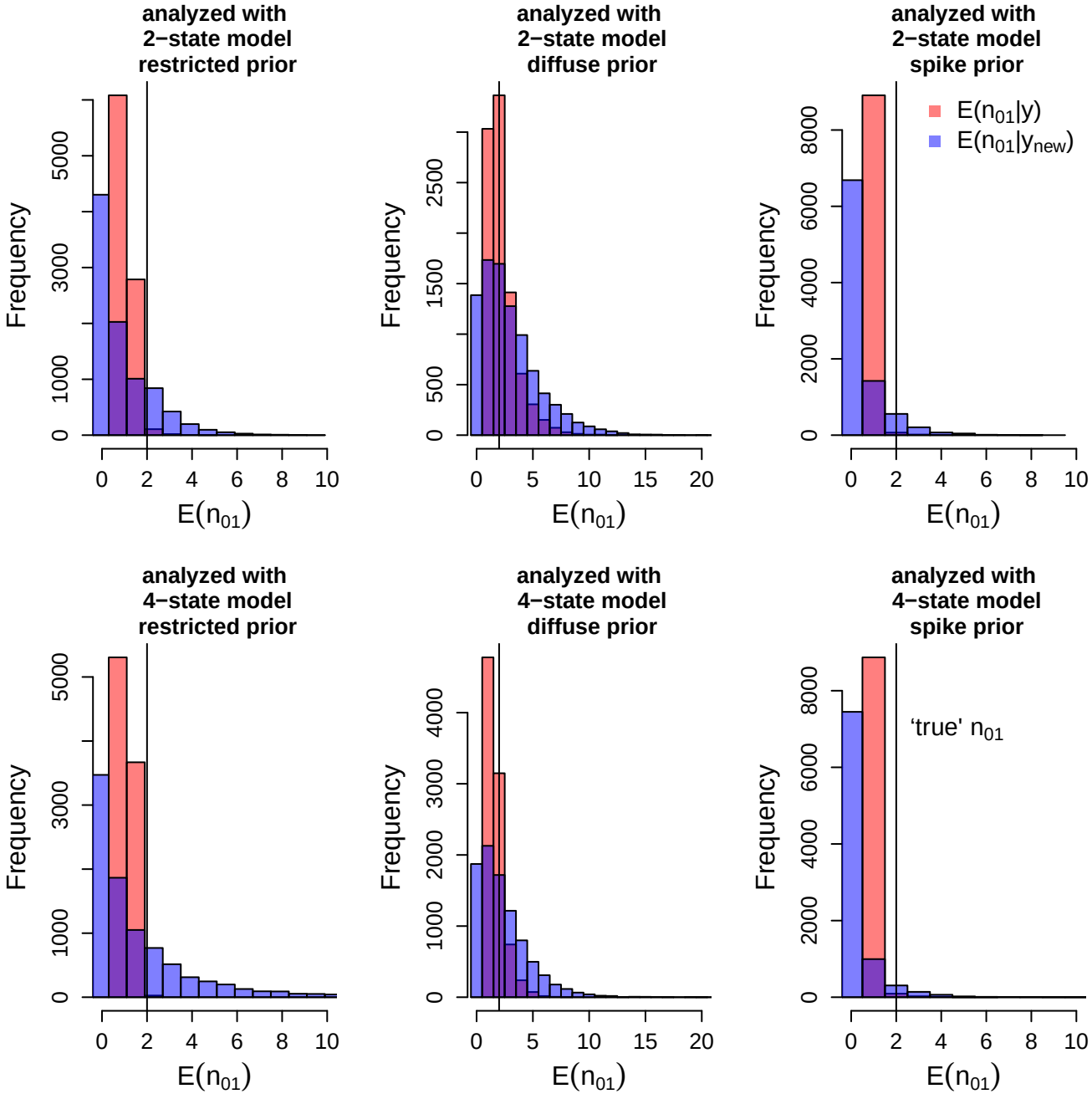

Figure B.14: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

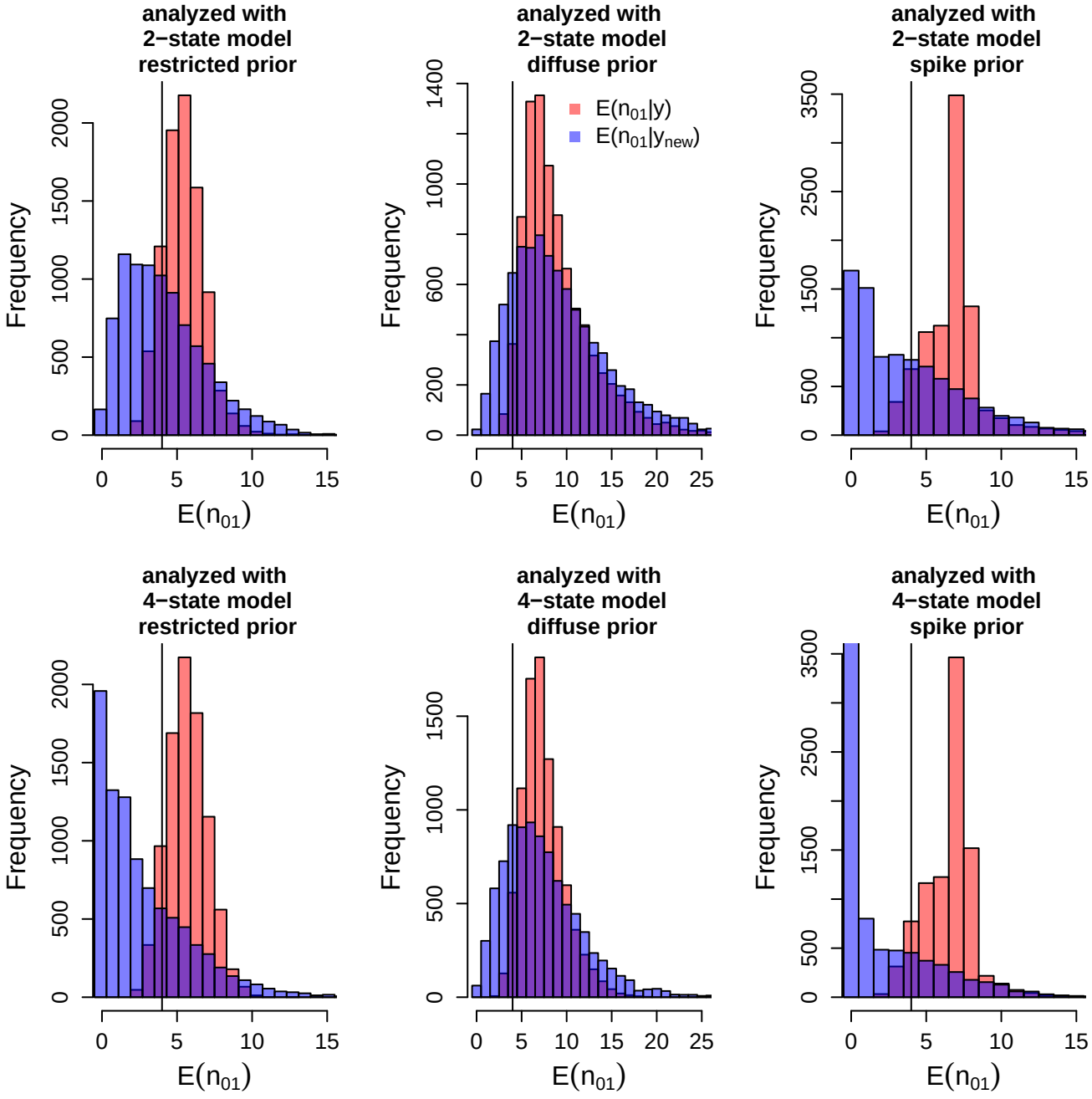

Figure B.15: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 4. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

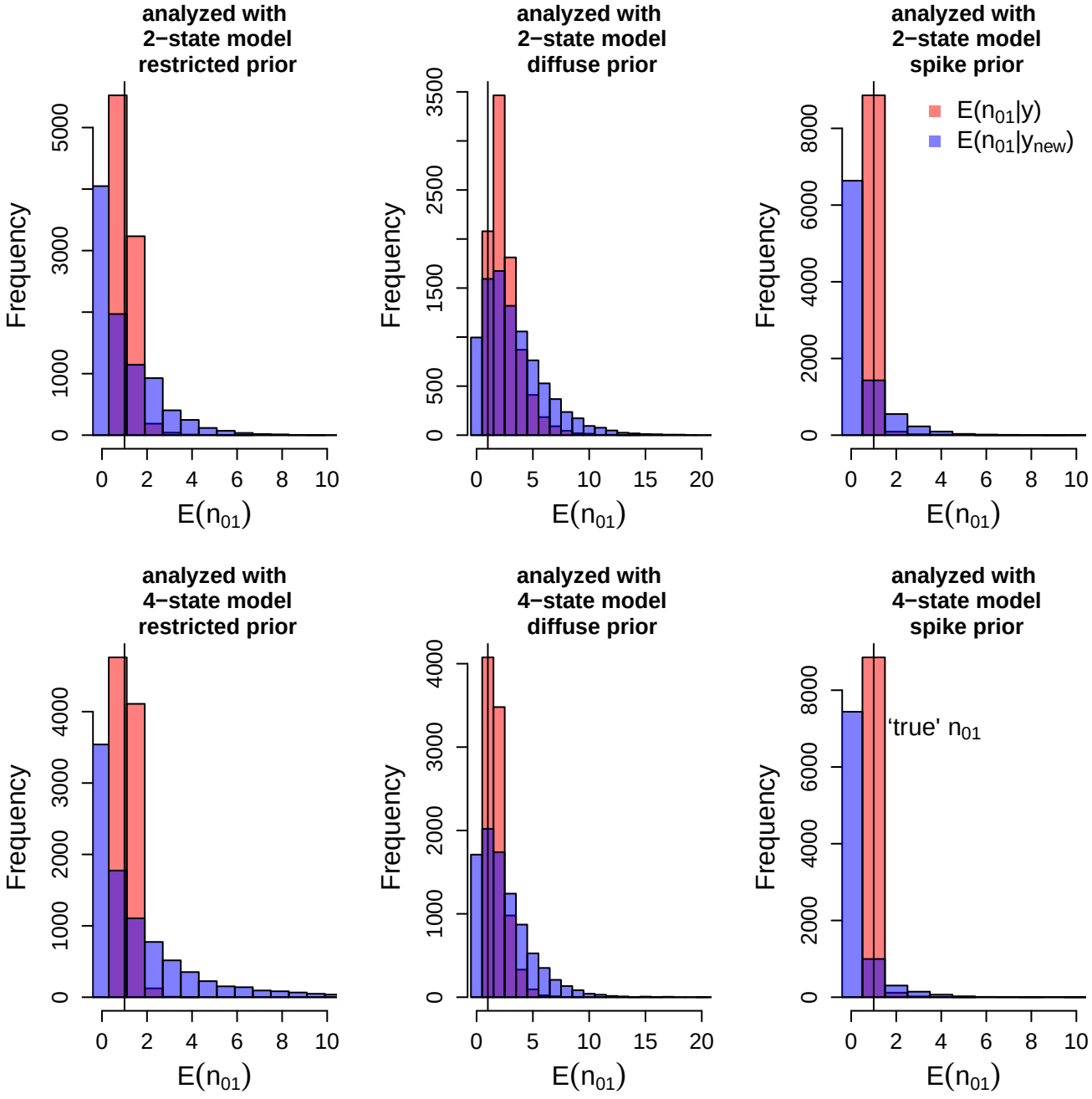

Figure B.16: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

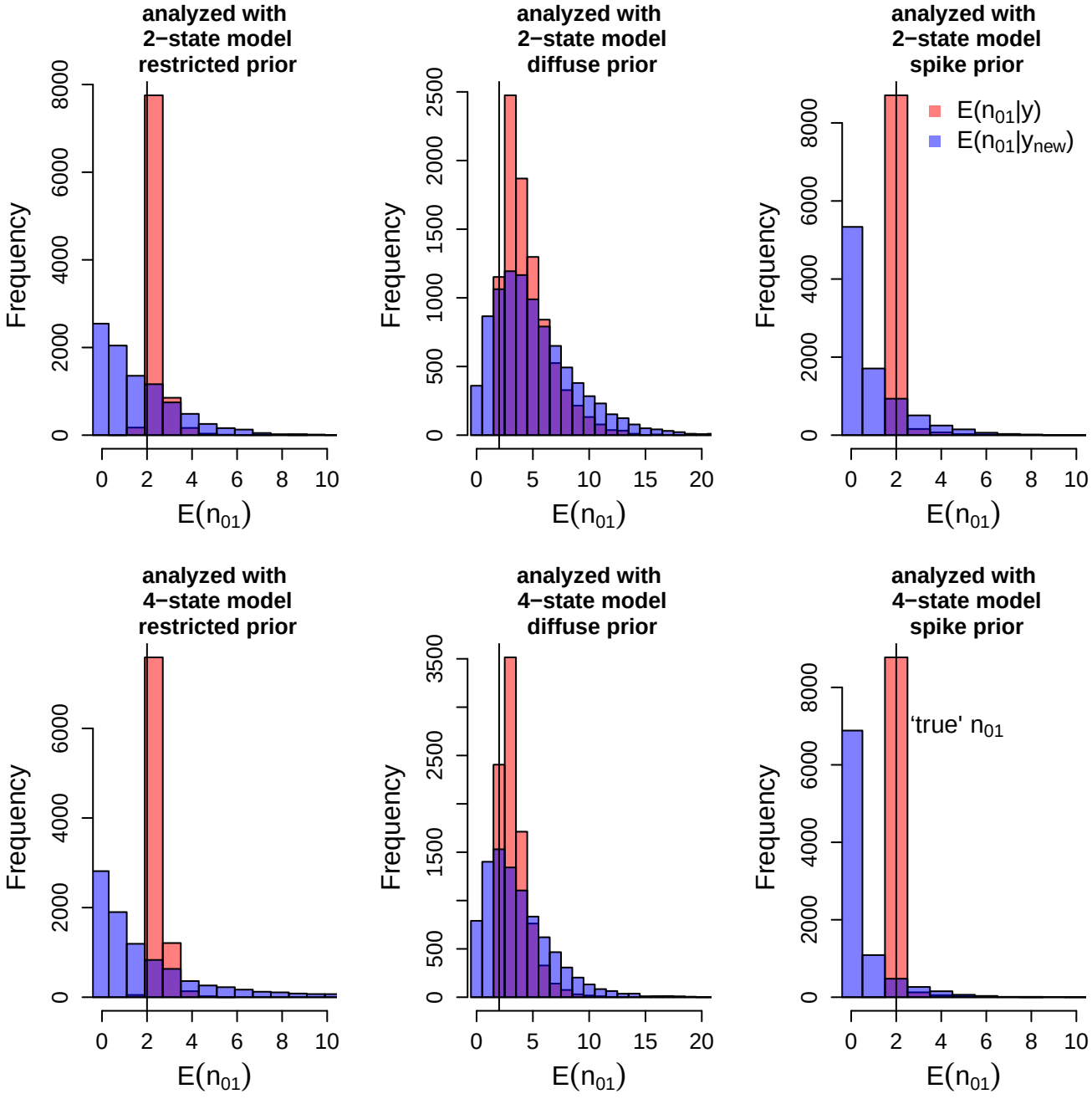

Figure B.17: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

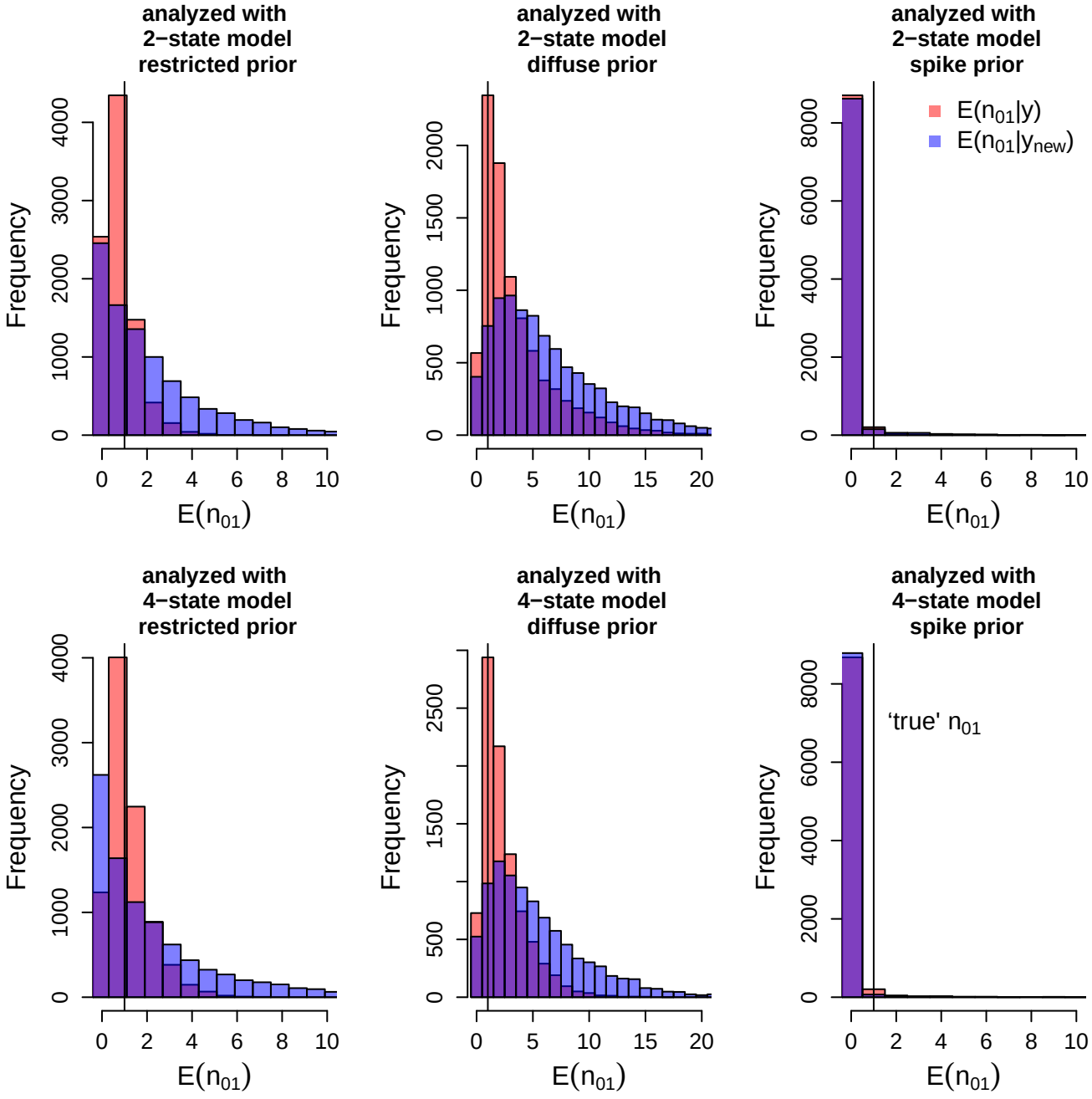

Figure B.18: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

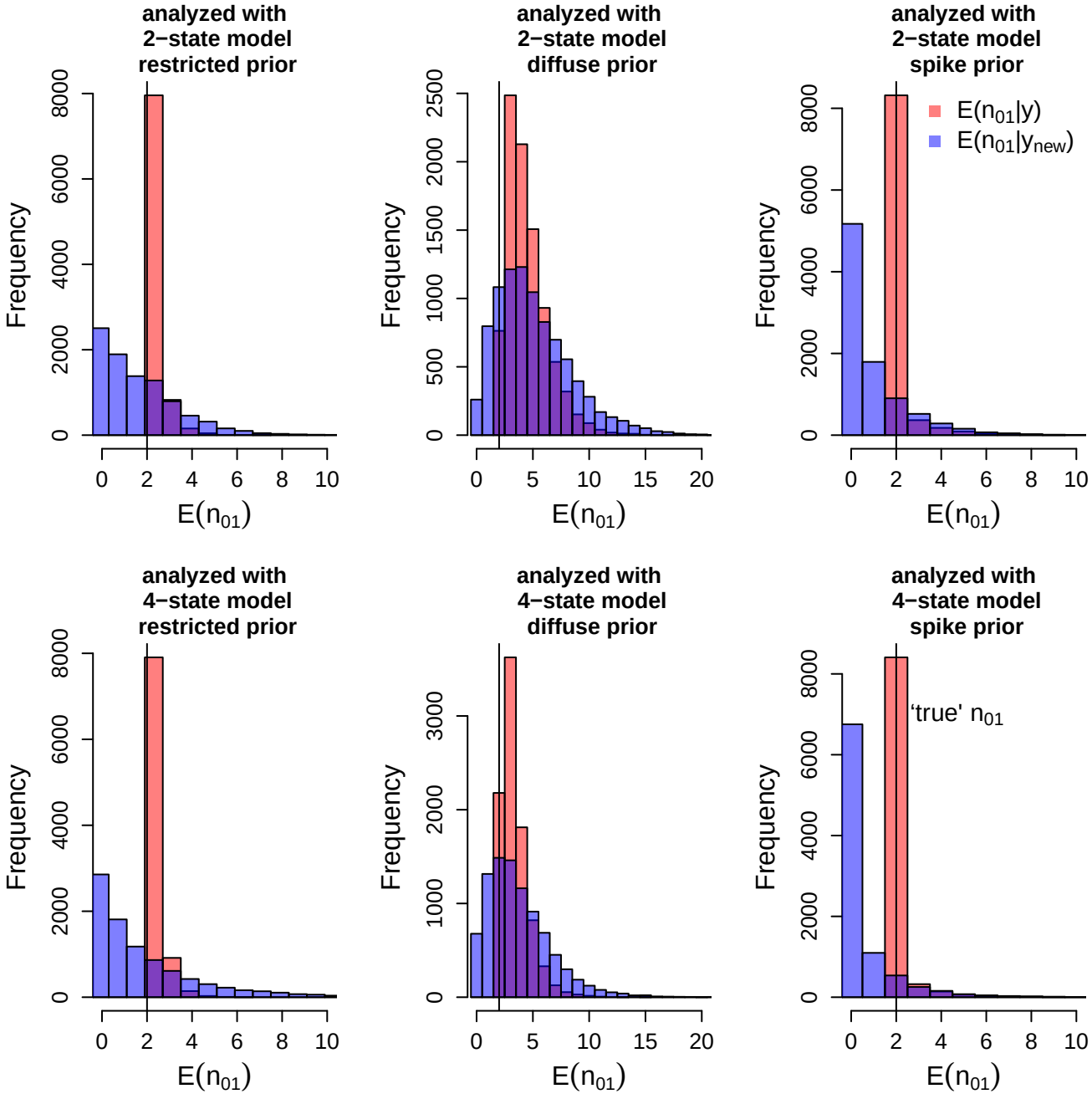

Figure B.19: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

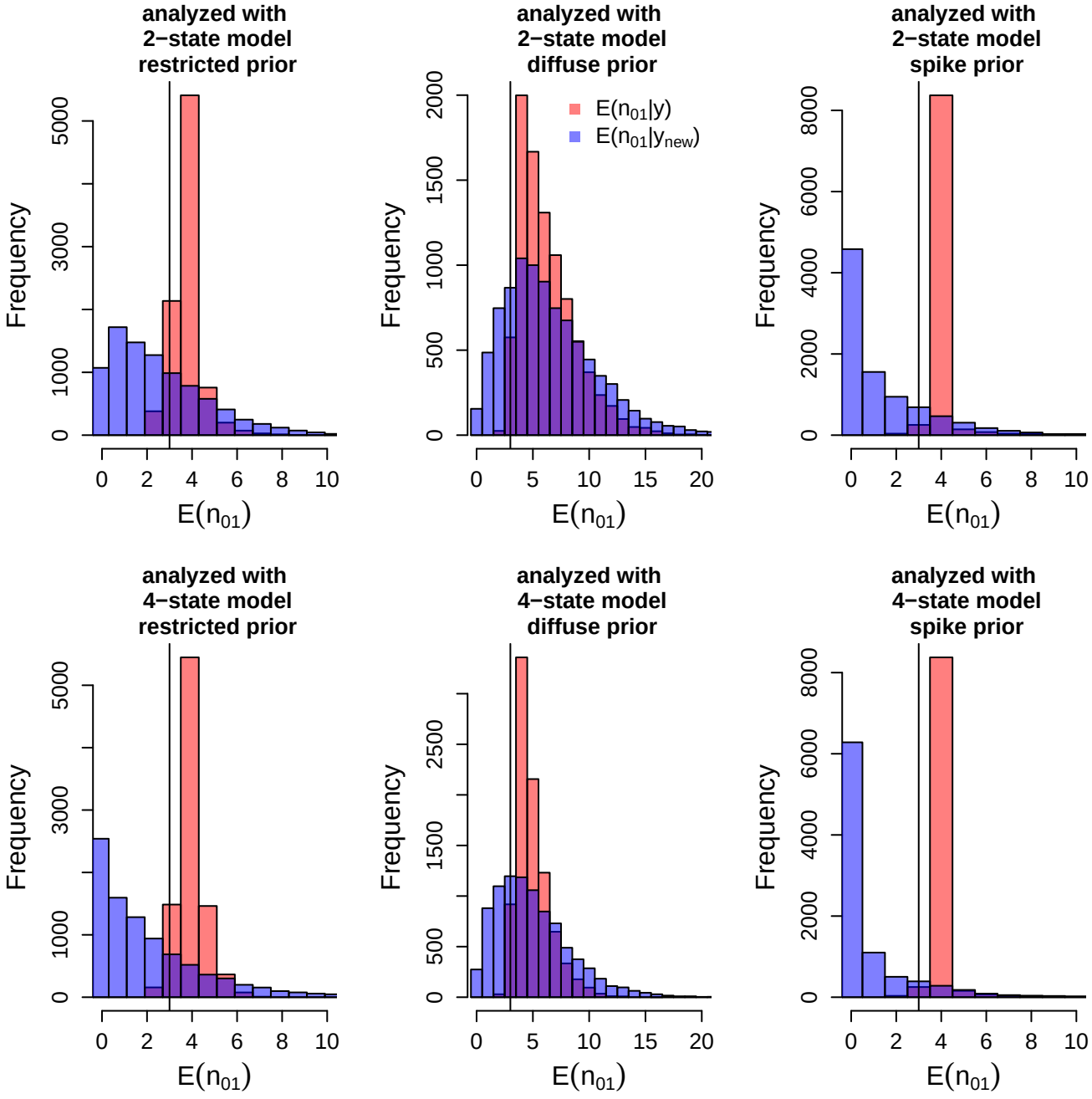

Figure B.20: Posterior predictive plots for the expected number of trait gains using a 70 tip tree. The data was generated from a 2-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution.

'true' n_{01}	analysis model	restricted	diffuse	spike
0	2-state	0.60	0.79	0.04
	4-state	0.53	0.78	0.03
0	2-state	0.51	0.74	0.01
	4-state	0.70	0.75	0.02
0	2-state	0.74	0.74	0.09
	4-state	0.68	0.78	0.05
2	2-state	0.30	0.59	0.12
	4-state	0.41	0.57	0.07
4	2-state	0.25	0.50	0.22
	4-state	0.15	0.43	0.11
1	2-state	0.32	0.61	0.13
	4-state	0.40	0.57	0.08
2	2-state	0.23	0.55	0.12
	4-state	0.27	0.47	0.06
1	2-state	0.67	0.88	0.05
	4-state	0.55	0.88	0.03
2	2-state	0.26	0.56	0.14
	4-state	0.29	0.49	0.08
3	2-state	0.22	0.48	0.11
	4-state	0.17	0.43	0.06

Table B.1: Tail probabilities of our discrepancy for ten simulated data sets on a 70 tip tree. The data sets were simulated from a 2-state model and the 'true' number of trait gains associated with each simulated data set can be found in the first column. Our discrepancy is $D(y, \boldsymbol{\theta}) = E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta})$ and the tail probability is $p_b(y) = p_A(D(y^{\mathrm{rep}}, \theta) \geq D(y, \theta)|H, y)$. We computed tail probabilities for each data set six times, using three different prior sets (restricted, diffuse, and spike) for the 2-state model and for the 4-state model.

Data simulated from a 4-state model on a 70 tip tree with a fixed root state

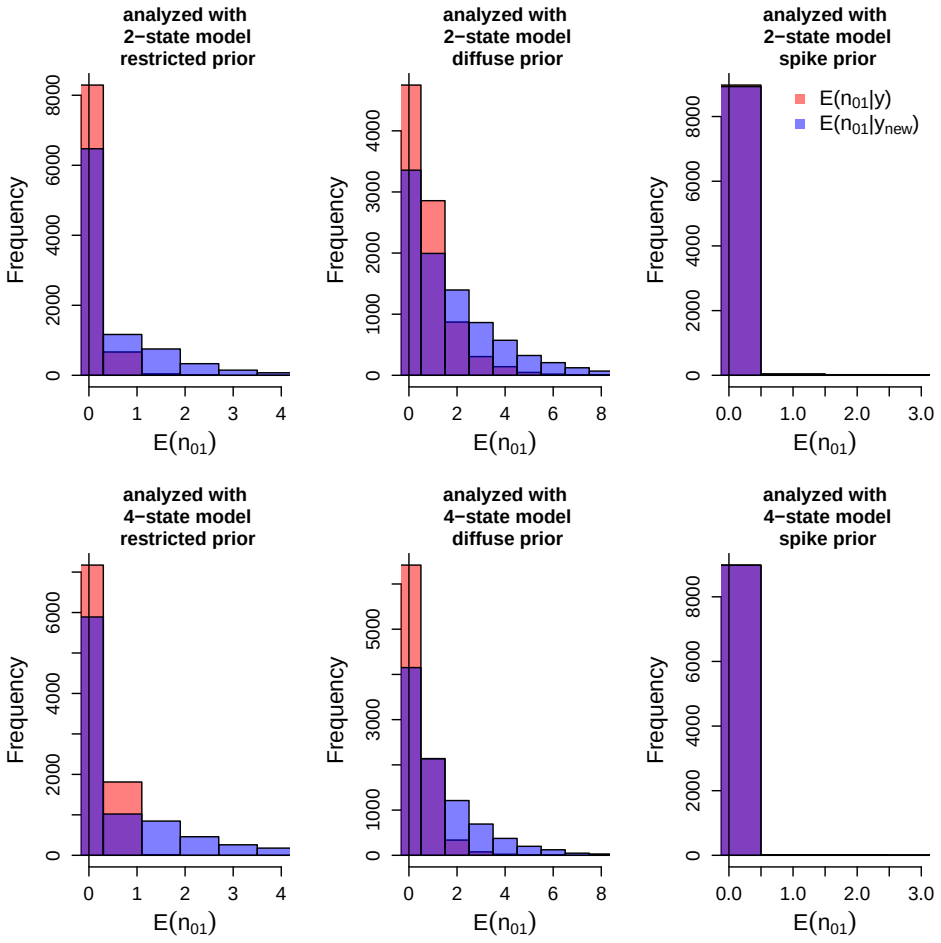

Figure B.21: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

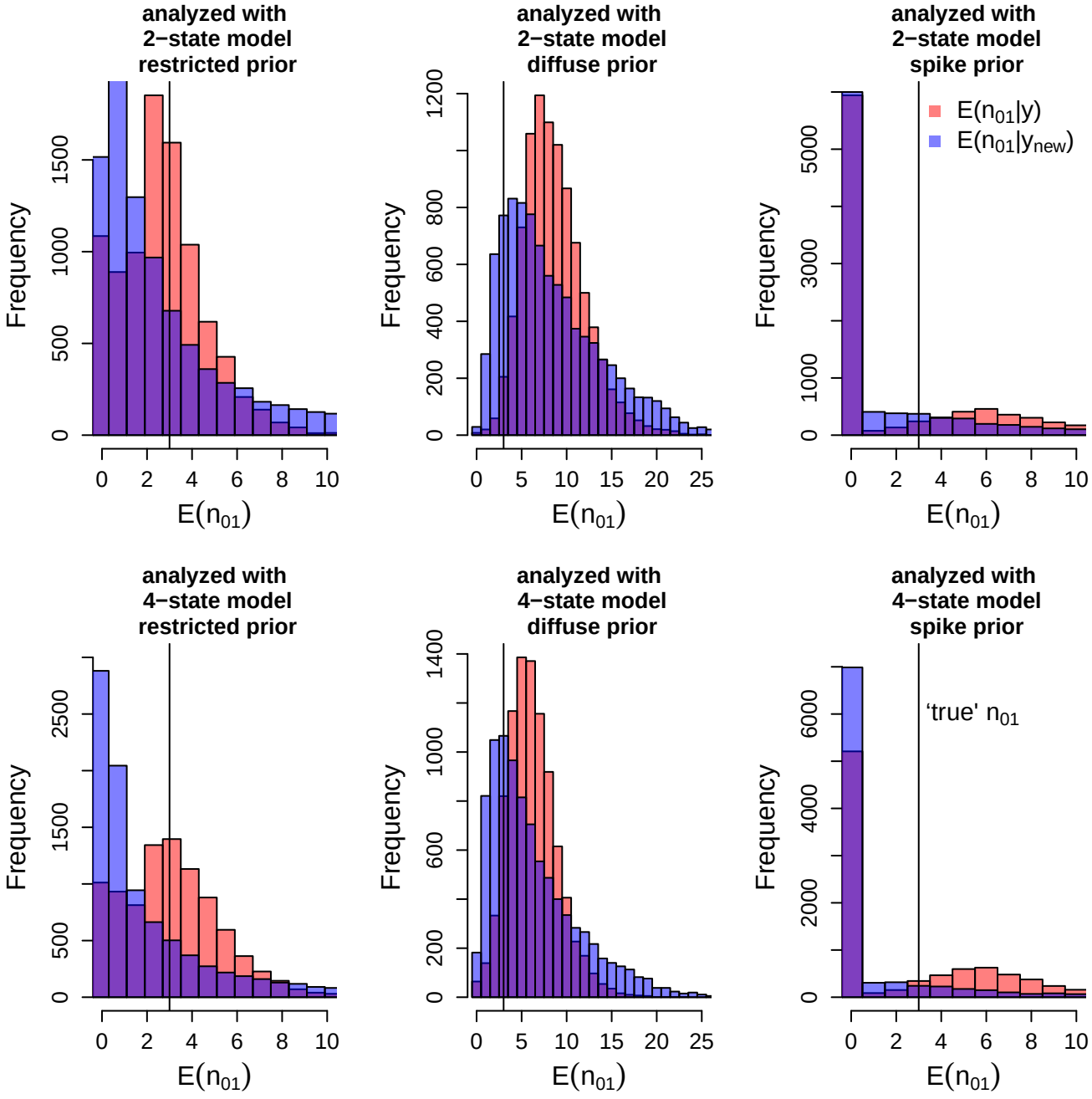

Figure B.22: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 3. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

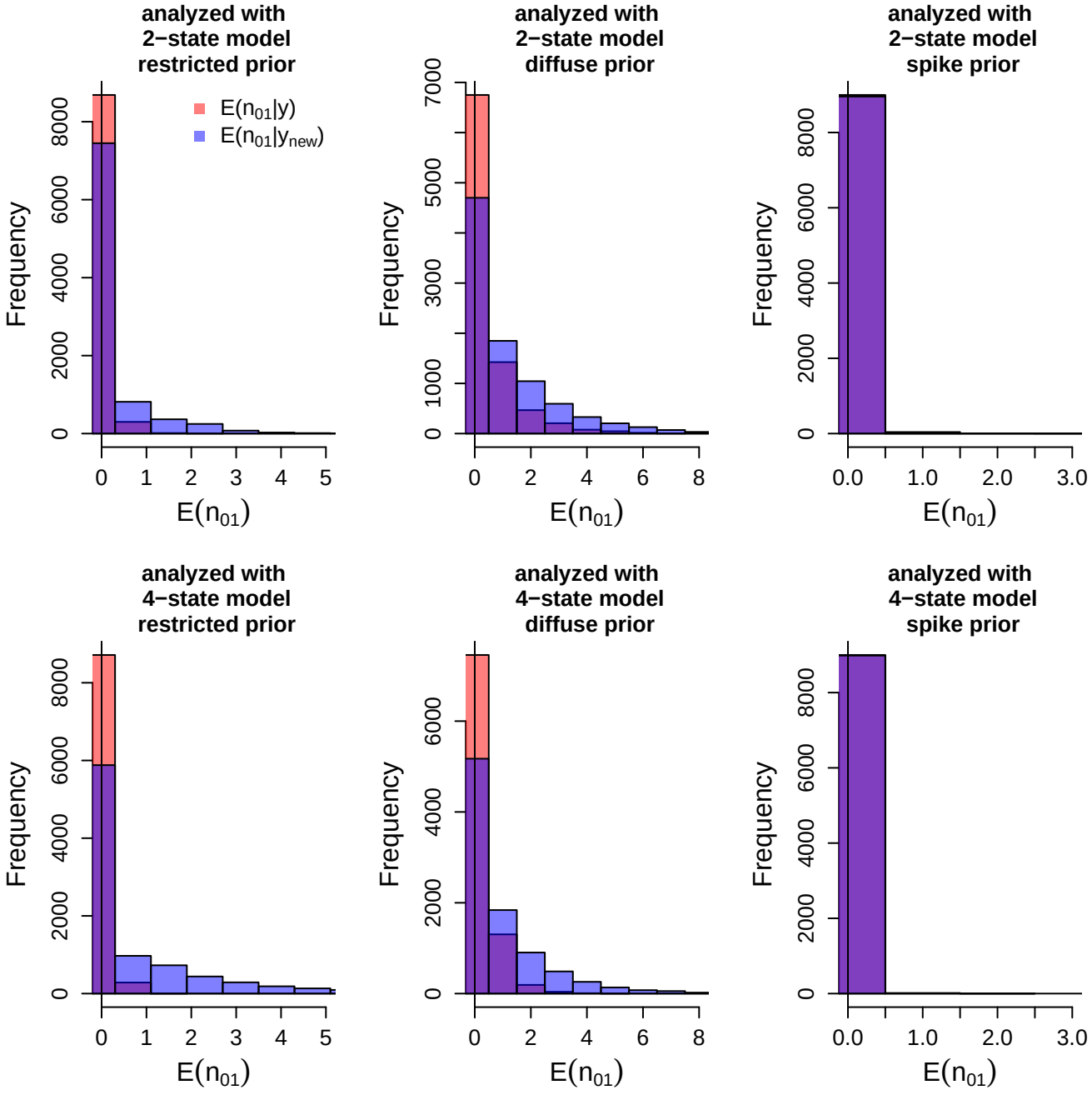

Figure B.23: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

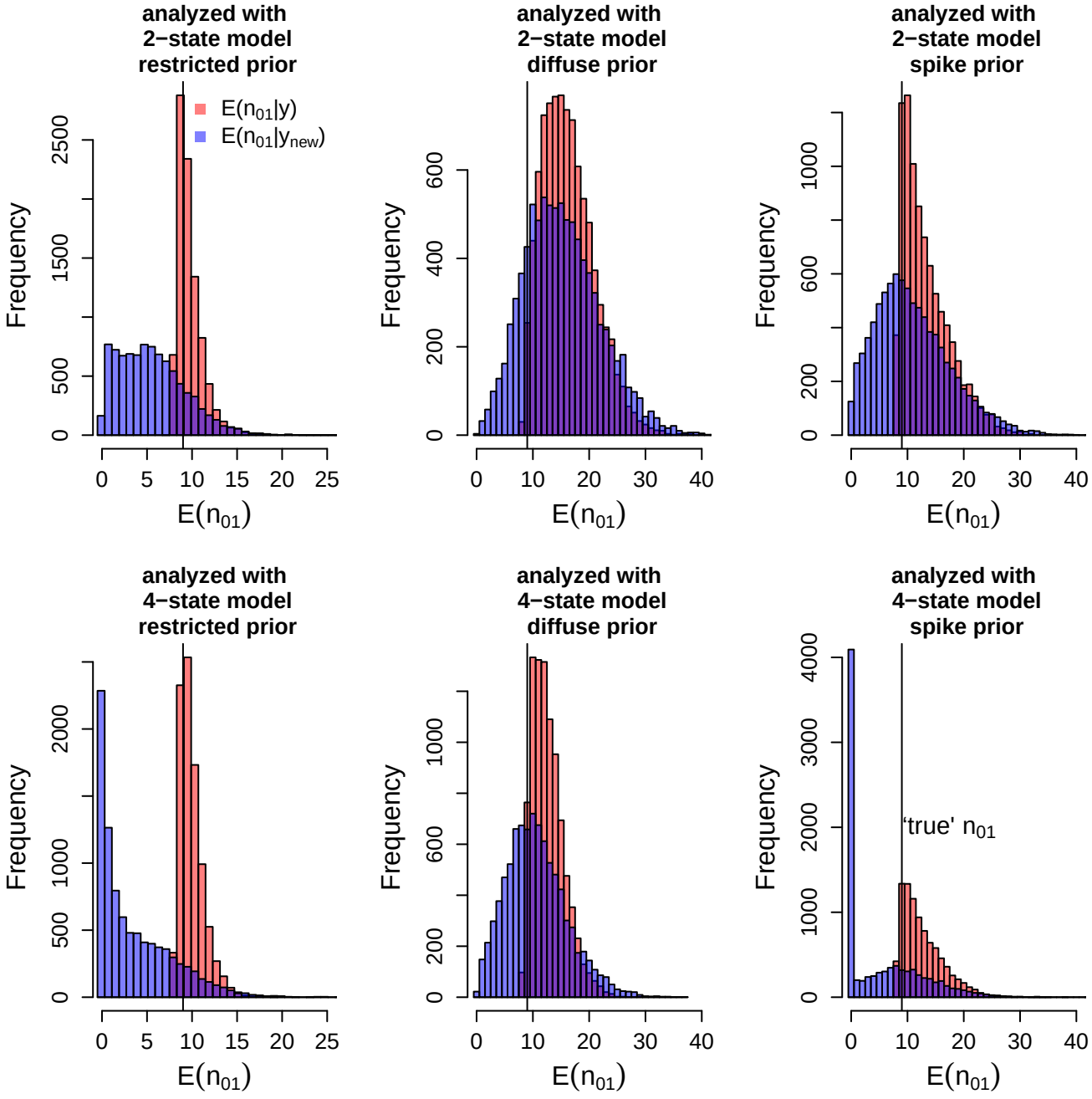

Figure B.24: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 9. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

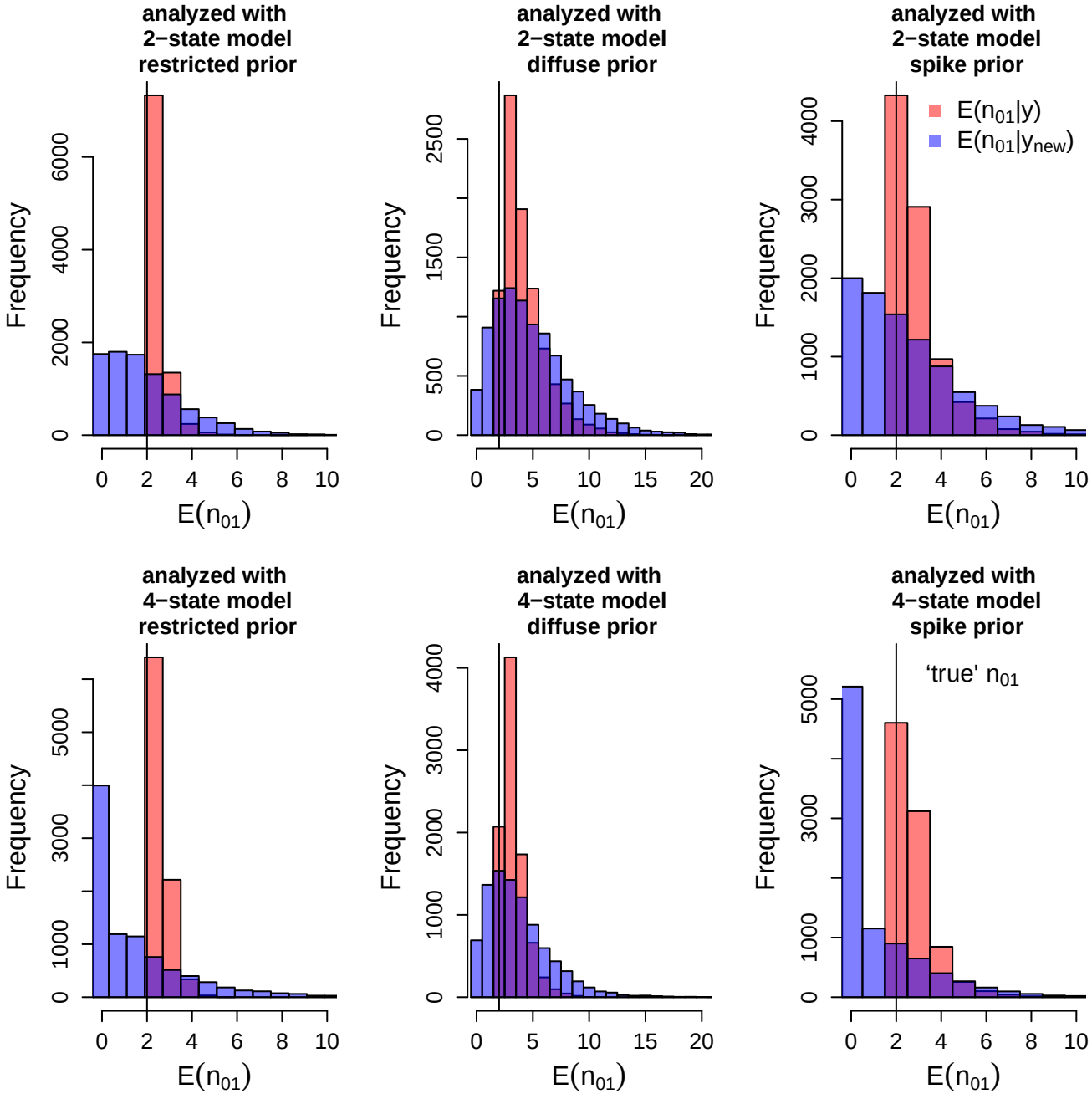

Figure B.25: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

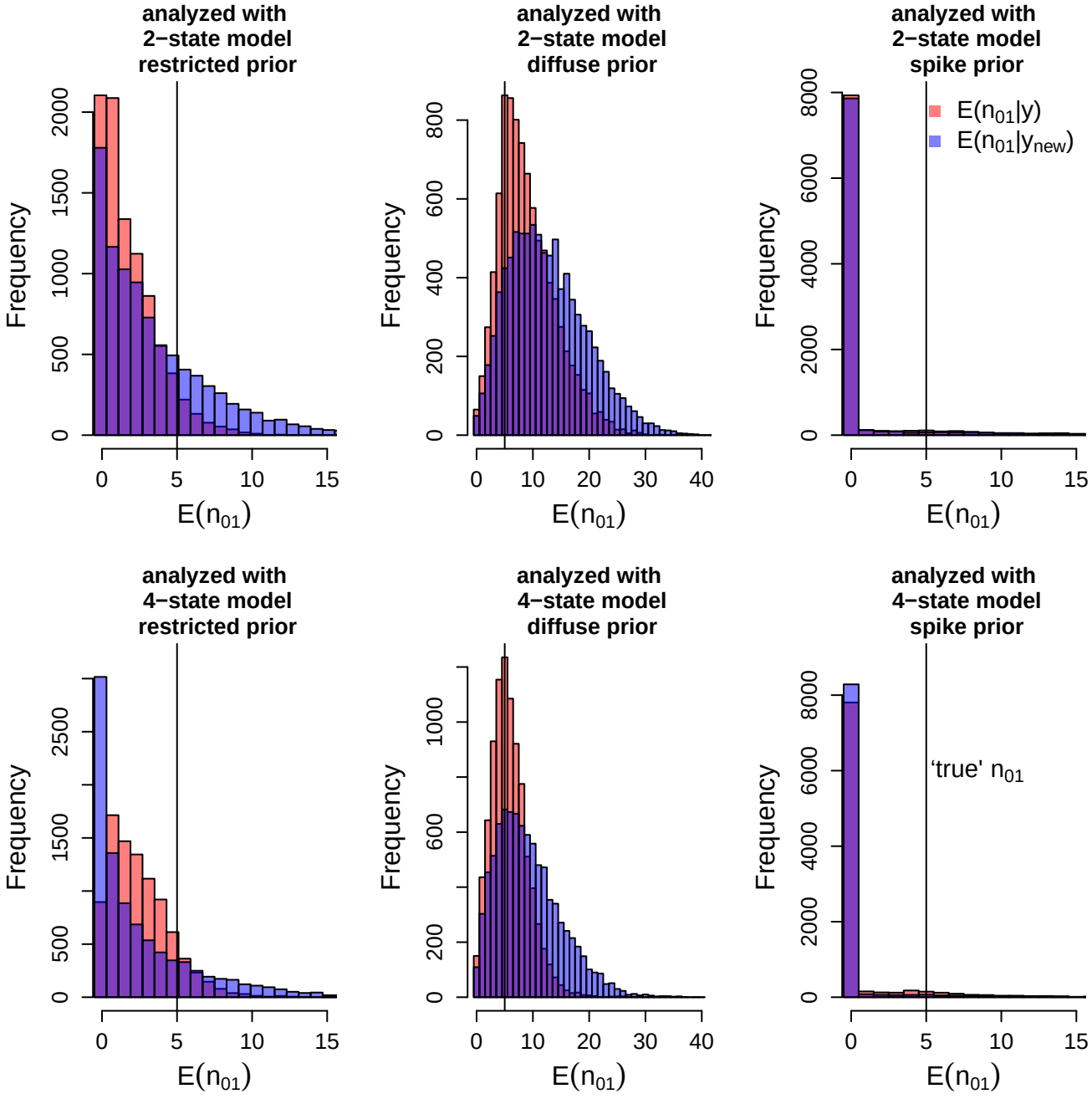

Figure B.26: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 5. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

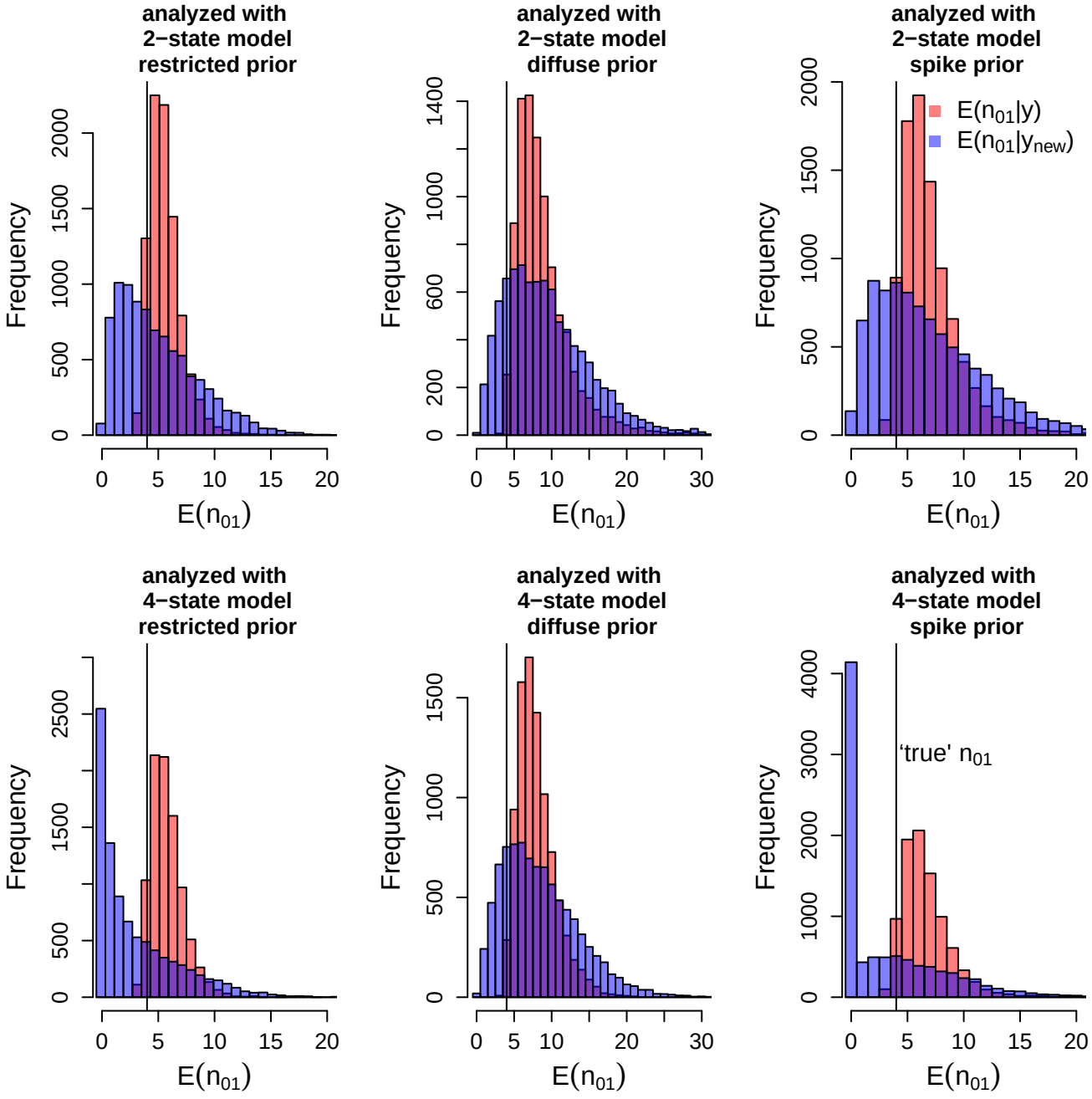

Figure B.27: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 4. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

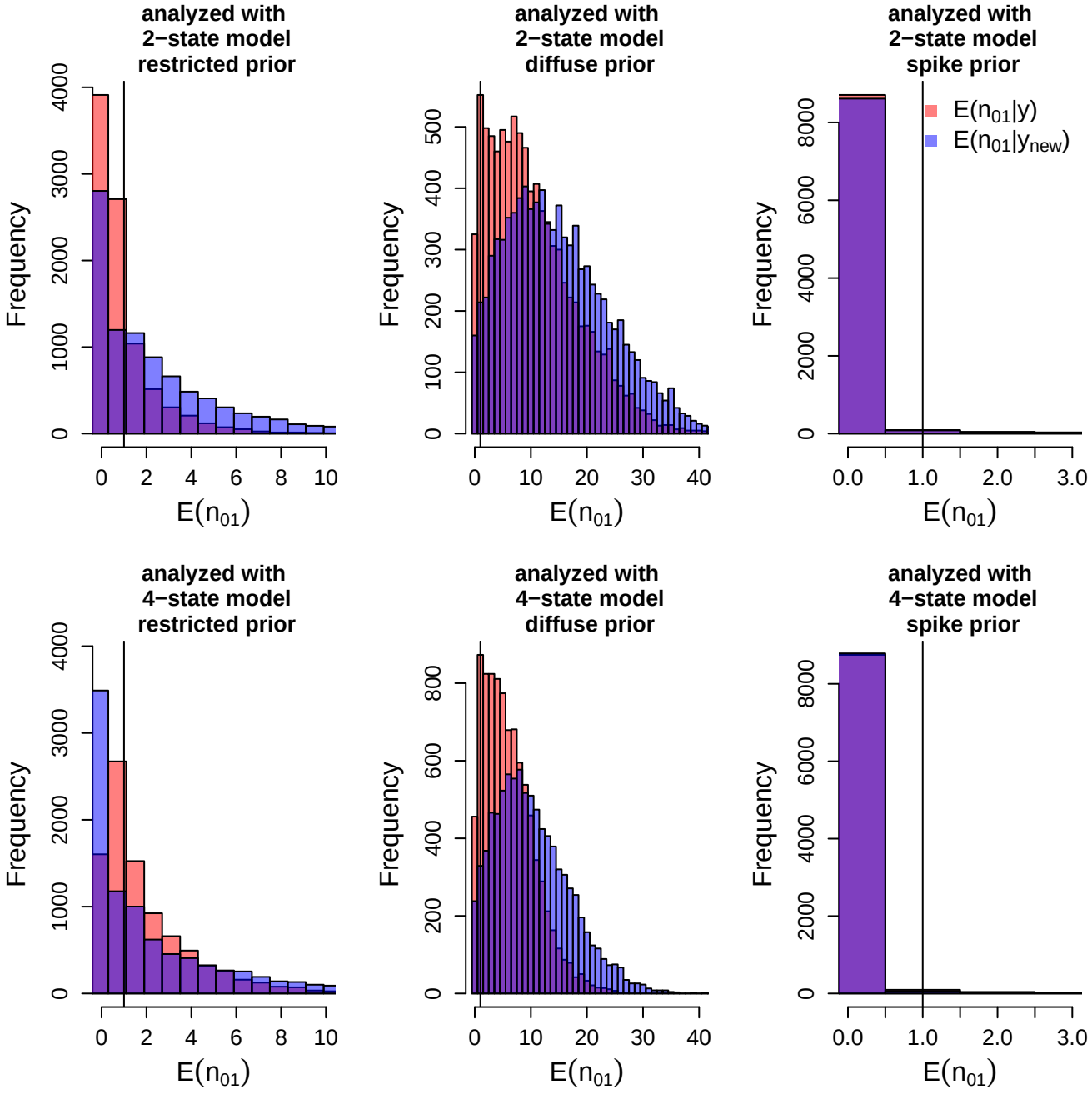

Figure B.28: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

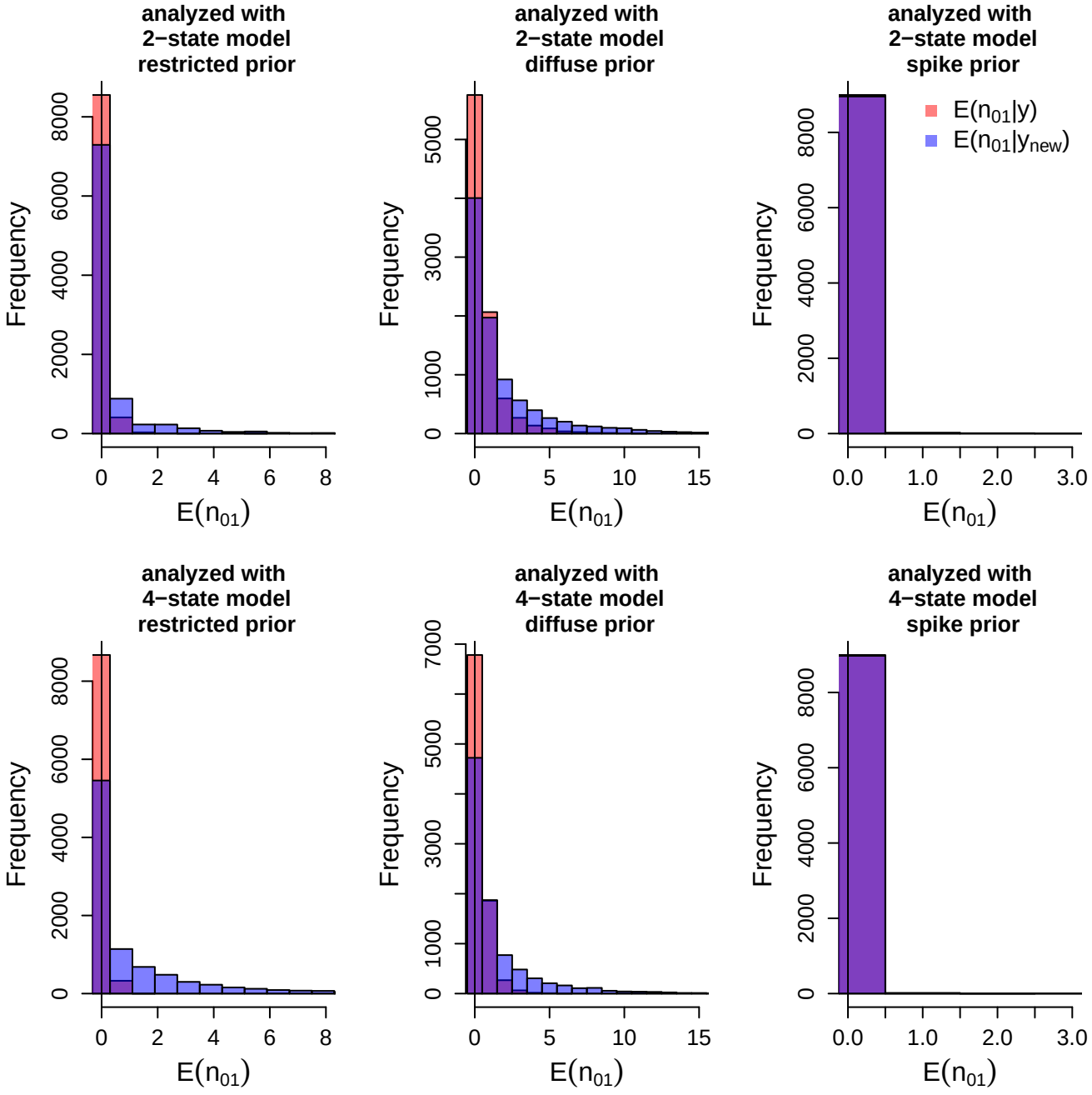

Figure B.29: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

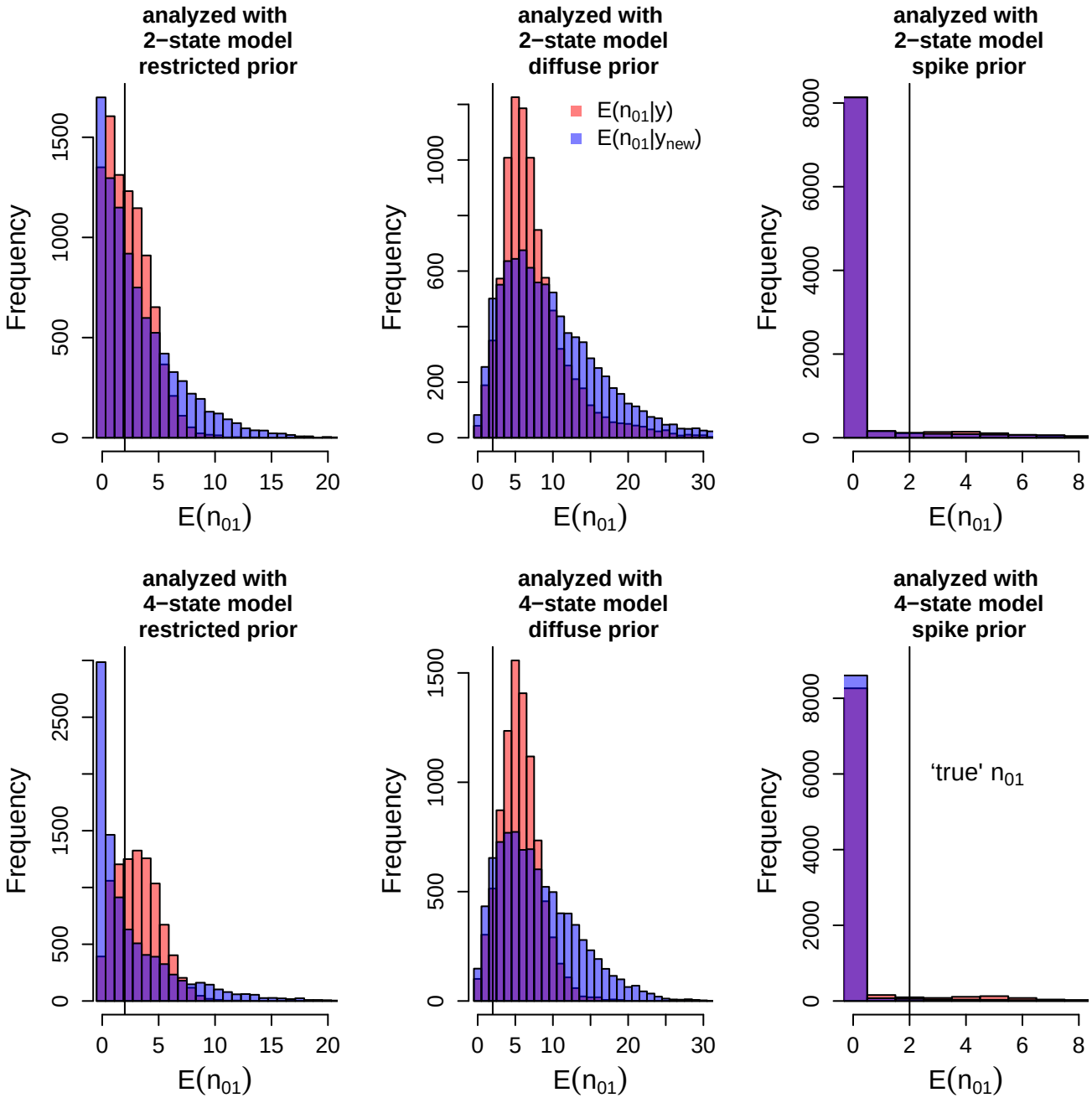

Figure B.30: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node forced to be in state 1 or state 3. The data was generated from a 4-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we forced the state of the root to be state 1 or state 3.

'true' n_{01}	analysis model	restricted	diffuse	spike
0	2-state	0.67	0.75	0.03
	4-state	0.50	0.75	0.01
3	2-state	0.36	0.31	0.10
	4-state	0.28	0.35	0.07
0	2-state	0.50	0.70	0.02
	4-state	0.57	0.70	0.01
9	2-state	0.08	0.36	0.22
	4-state	0.06	0.24	0.11
2	2-state	0.29	0.56	0.32
	4-state	0.22	0.49	0.16
5	2-state	0.67	0.83	0.12
	4-state	0.41	0.78	0.07
4	2-state	0.36	0.49	0.42
	4-state	0.20	0.49	0.22
1	2-state	0.76	0.93	0.06
	4-state	0.47	0.91	0.03
0	2-state	0.54	0.81	0.02
	4-state	0.67	0.77	0.01
2	2-state	0.50	0.65	0.07
	4-state	0.32	0.64	0.03

Table B.2: Tail probabilities of our discrepancy for ten simulated data sets on a 70 tip tree with the root node forced to be in state 1 or state 3. The data sets were simulated from a 4-state model and the 'true' number of trait gains associated with each simulated data set can be found in the first column. Our discrepancy is $D(y, \boldsymbol{\theta}) = E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta})$ and the tail probability is $p_b(y) = p_A(D(y^{\mathrm{rep}}, \theta) \geq D(y, \theta)|H, y)$. We computed tail probabilities for each data set six times, using three different prior sets (restricted, diffuse, and spike) for the 2-state model and for the 4-state model. In these analyses we forced the state of the root to be state 1 or state 3.

Data simulated from a 2-state model on a 70 tip tree with a fixed root state

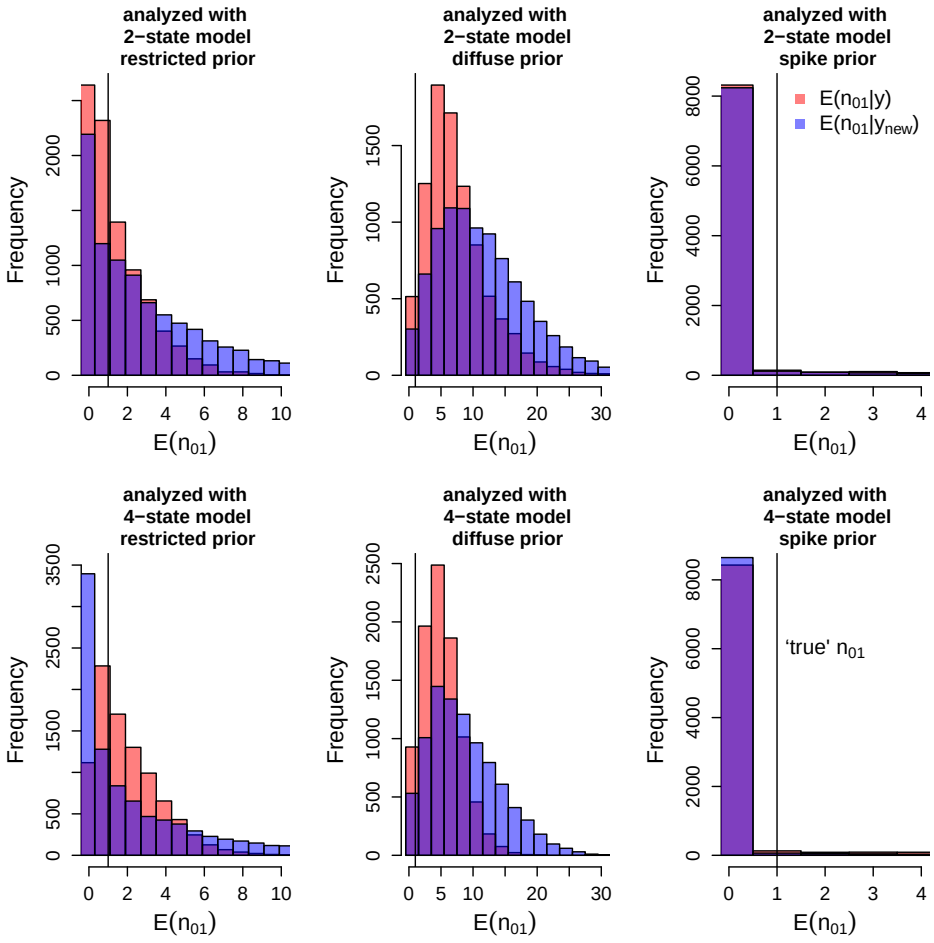

Figure B.31: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

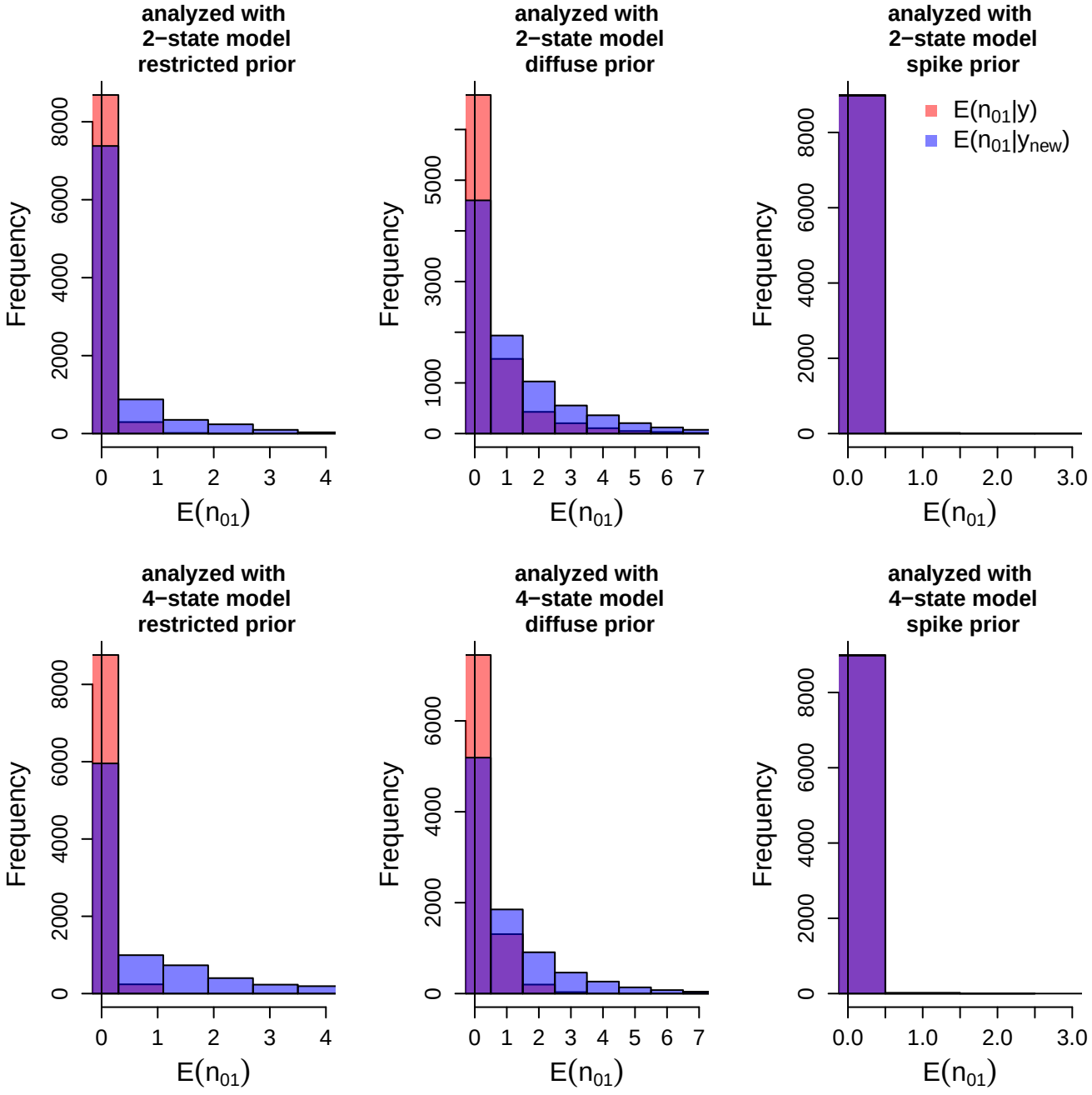

Figure B.32: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

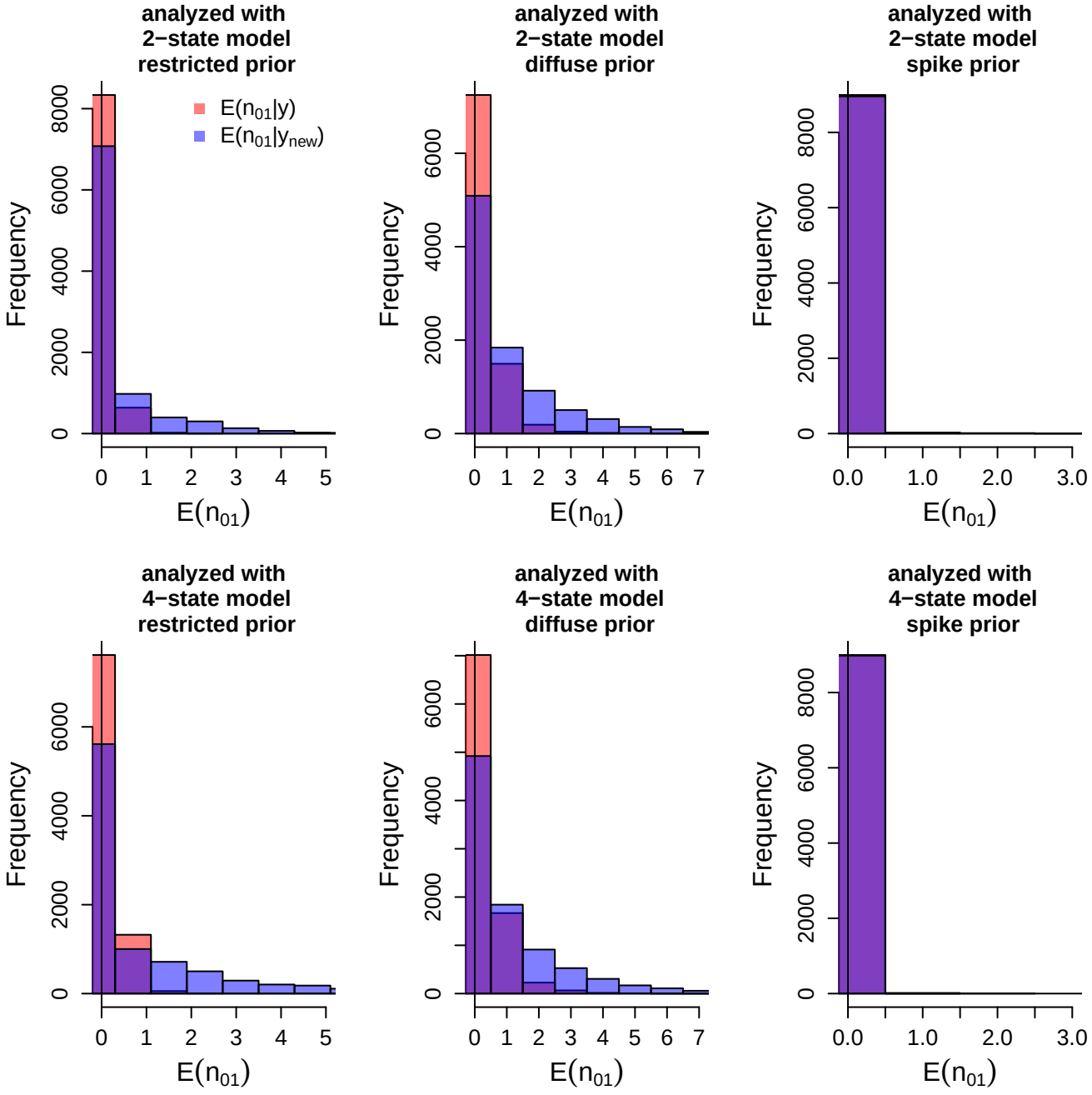

Figure B.33: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

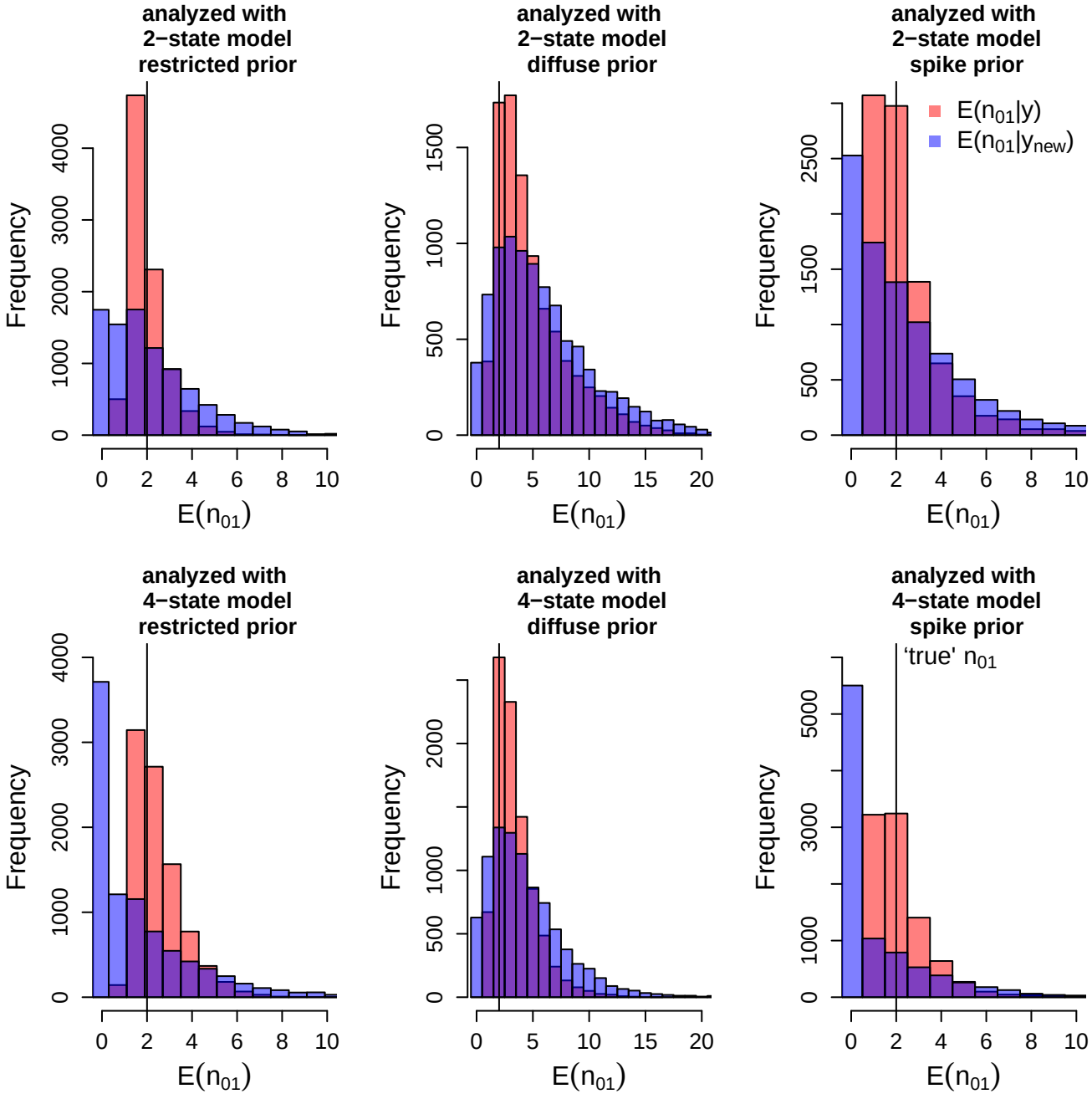

Figure B.34: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

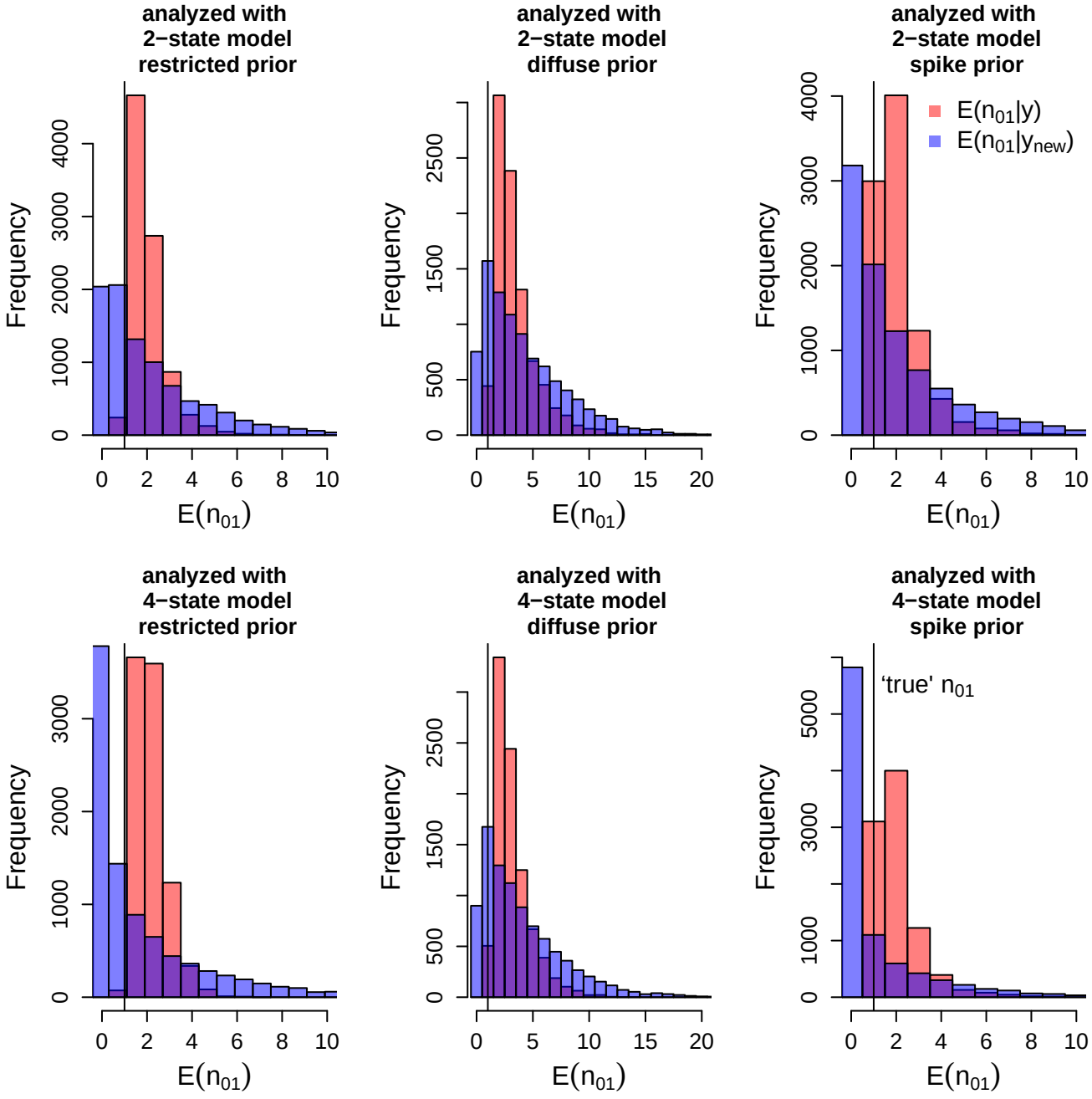

Figure B.35: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 1. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

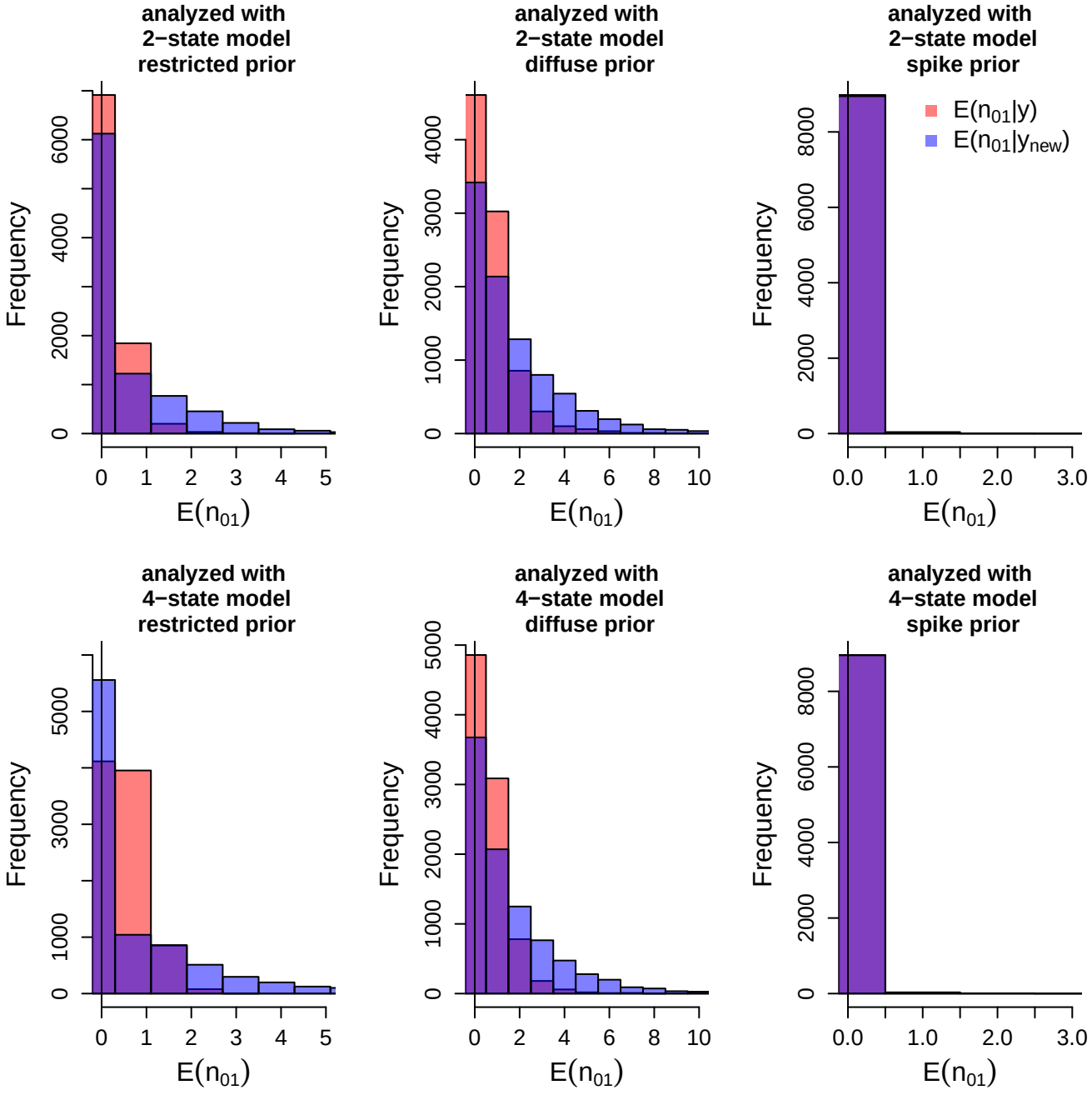

Figure B.36: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 0. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

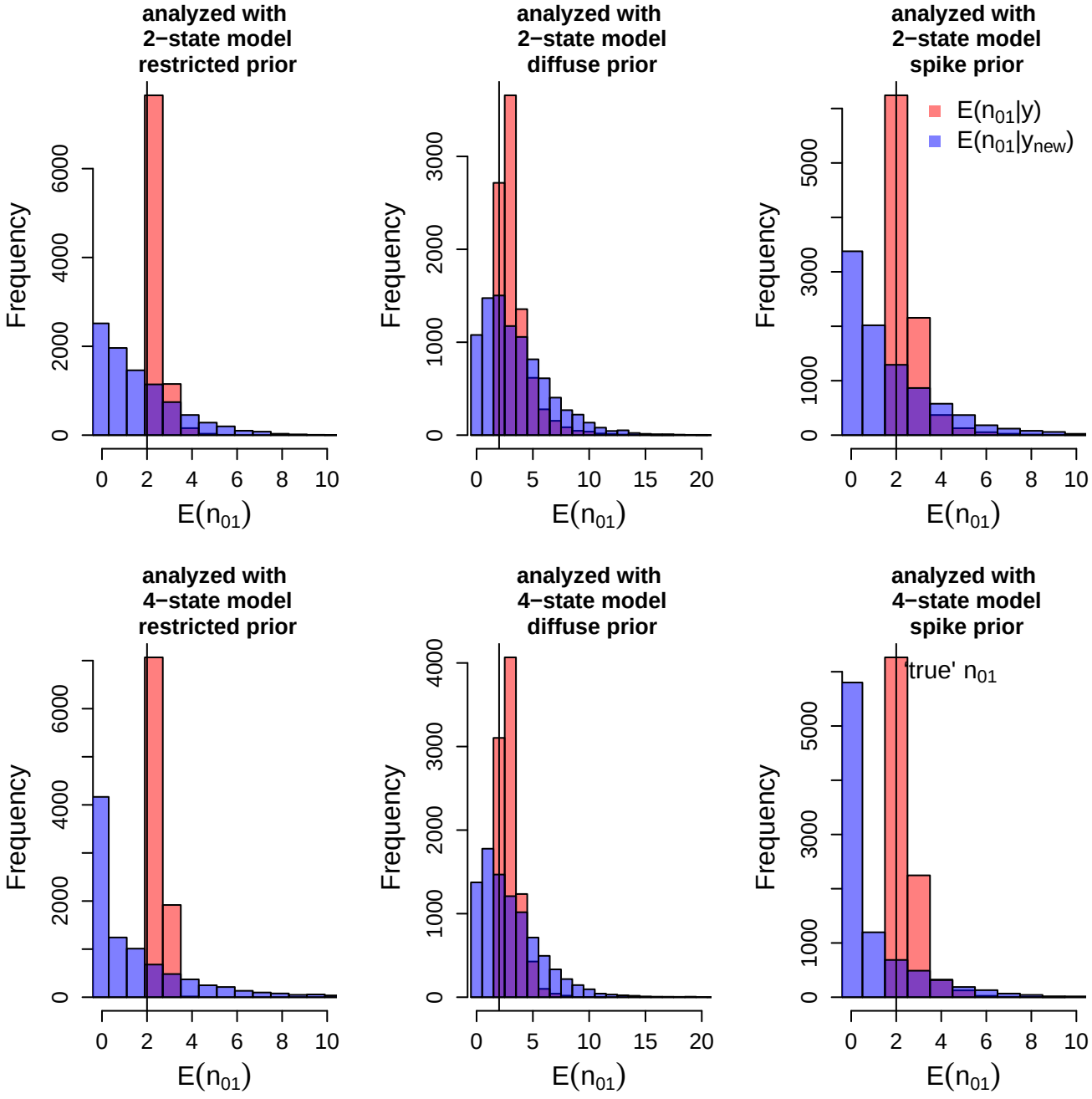

Figure B.37: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

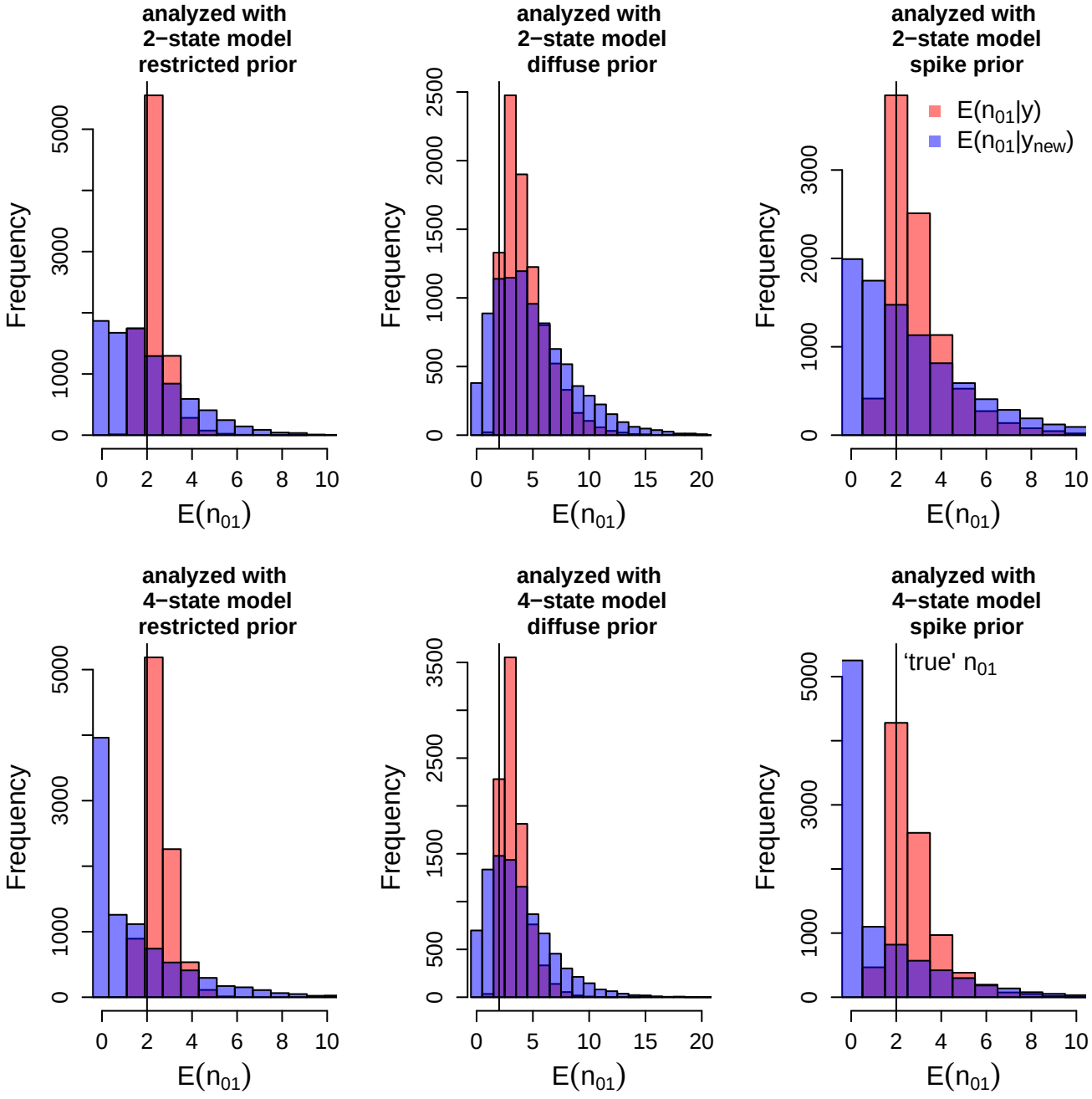

Figure B.38: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

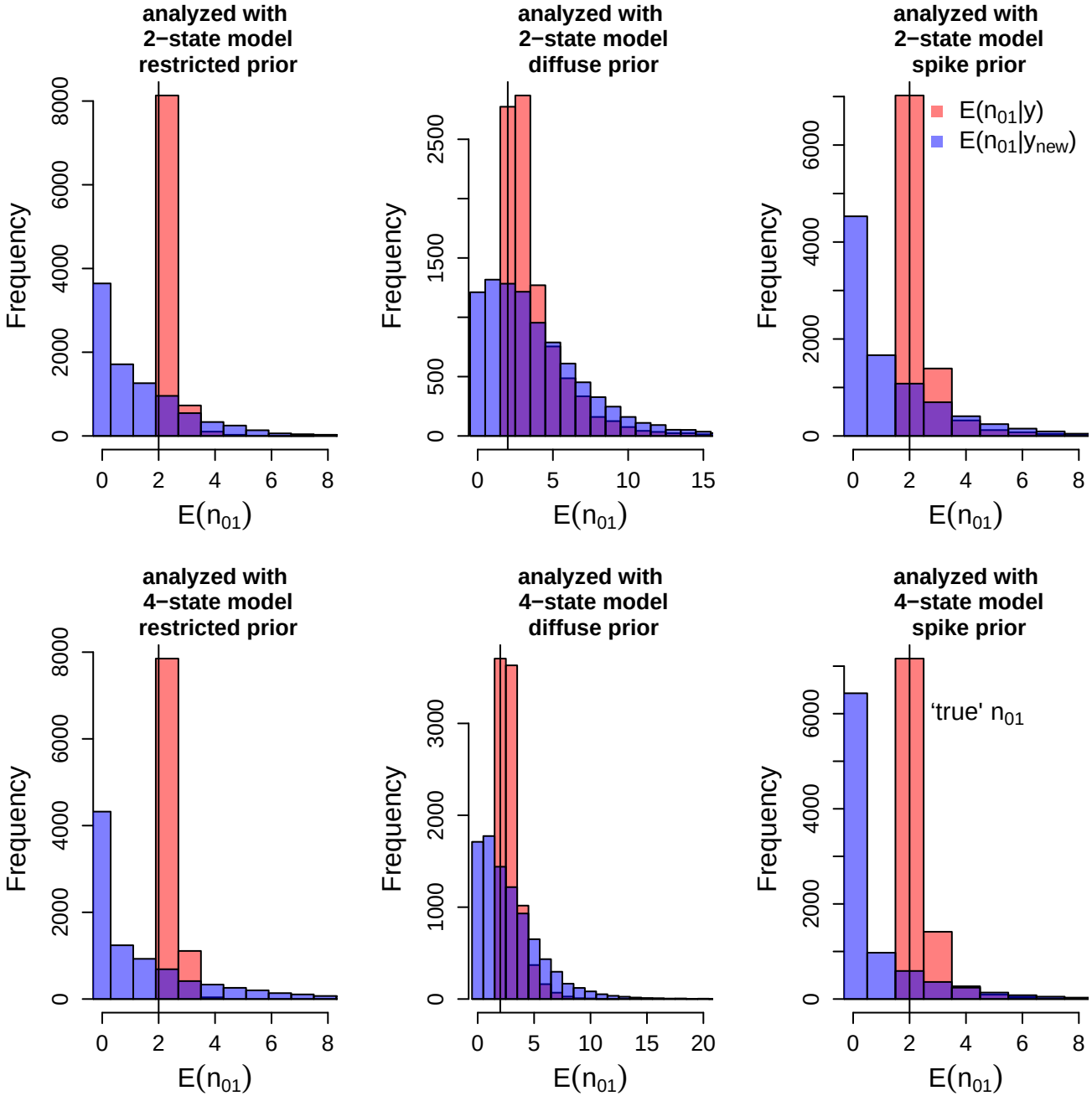

Figure B.39: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 2. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

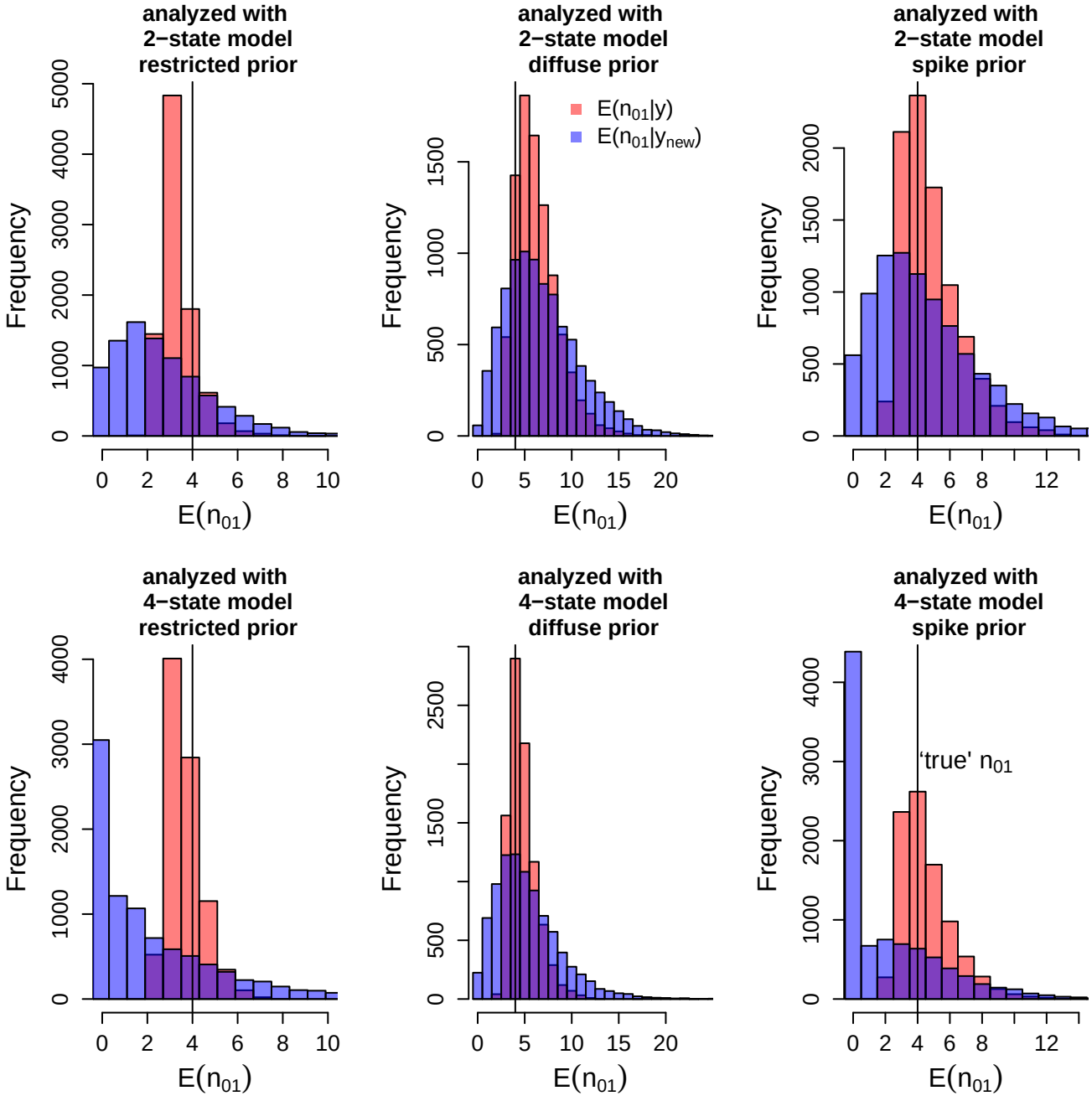

Figure B.40: Posterior predictive plots for the expected number of trait gains using a 70 tip tree with the root node fixed in state 1. The data was generated from a 2-state model and the 'true' number of trait gains was 4. The top plots show results for the 2-state analysis. The bottom plots show results for the 4-state analysis. The first column shows results for the restricted priors, the second column shows results for the diffuse priors and the third column shows results for the spike priors. The pink bars show the expected number of trait gains conditional on the 'true' tip data. The blue bars show the corresponding reference distribution. In these analyses we fixed the state of the root to be in state 1.

'true' n_{01}	analysis model	restricted	diffuse	spike
1	2-state	0.68	0.89	0.09
	4-state	0.43	0.85	0.04
0	2-state	0.50	0.70	0.02
	4-state	0.56	0.69	0.01
0	2-state	0.44	0.63	0.02
	4-state	0.52	0.63	0.01
2	2-state	0.42	0.64	0.38
	4-state	0.27	0.61	0.21
1	2-state	0.36	0.51	0.32
	4-state	0.30	0.50	0.18
0	2-state	0.45	0.66	0.02
	4-state	0.38	0.65	0.01
2	2-state	0.24	0.45	0.24
	4-state	0.22	0.42	0.14
2	2-state	0.32	0.56	0.34
	4-state	0.22	0.50	0.18
2	2-state	0.19	0.45	0.18
	4-state	0.23	0.39	0.11
4	2-state	0.29	0.55	0.38
	4-state	0.25	0.51	0.20

Table B.3: Tail probabilities of our discrepancy for ten simulated data sets on a 70 tip tree with the root node fixed in state 1. The data sets were simulated from a 2-state model and the 'true' number of trait gains associated with each simulated data set can be found in the first column. Our discrepancy is $D(y, \boldsymbol{\theta}) = E(\mathrm{n}_{01}|\mathbf{y}, \boldsymbol{\theta})$ and the tail probability is $p_b(y) = p_A(D(y^{\mathrm{rep}}, \theta) \geq D(y, \theta)|H, y)$. We computed tail probabilities for each data set six times, using three different prior sets (restricted, diffuse, and spike) for the 2-state model and for the 4-state model. In these analyses we fixed the state of the root to be in state 1.