Evolutionary modelling of biological structures



Michael Thomas Golden
Brasenose College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Michaelmas 2017

Acknowledgements

I would like to thank the many people who have helped me through the completion of my DPhil. The first is my supervisor, Jotun Hein, for the patient guidance and mentorship that he has provided me these past years. Jotun has a breadth of scientific curiosity and an intellect to match that has stimulated me to push my own intellectual boundaries. Jotun always has time for his students, and an unrivalled sense of humour. I'm very grateful for everything that Jotun has done for me, and that we were able to get along so well with one another.

Next I would like to thank my collaborators who have been essential in the completion of this work. Thanks goes to Eduardo García-Portugués, Thomas Hamelryck, Kanti Mardia, Michael Sørensen, and Ben Murrell.

I would also like to especially thank James Anderson, who started me off on my journey at Oxford – first supervising me and two other students on a very enjoyable and successful summer project held at Jotun's excellent Oxford Summer School in Computational Biology and then for helping me to apply for my DPhil.

Thanks also goes to the many researchers, some of whom are colleagues, who have helped me on my journey, namely: Darren Martin, Jeff Thorne, Elspeth Garman, Patrick Gemmell, Farah Colchester, Joe Herman, Mathias Cronjäger, William Larsen, Jimi Cullen, Owen Thomas, Jovana Mitrovic, Tammo Rukat, Eli Karin, and Haim Ashkenazy. I also would like to thank Madeline Mitchell, who made the OCGF building a much brighter place, and who went beyond her administrative role in making sure everyone got to know one another.

Finally, I would like to thank Jamie who has been my largest source of support, laughter, and happiness these past three years. I wouldn't have made it to the end without you. A enormous thanks to my parents (Mike and Mary), my sister (Melanie), and my brother (Robin), for their support and for cultivating an environment that encouraged me to pursue an interest in science.

Abstract

Many naturally occurring biological molecules, such as nucleic acids or proteins, are evolutionary related, having evolved from a common ancestor via a process of sequence mutation. Mutations undergo a process of selection, with combinations of mutations that preserve or improve a molecule's fitness having a higher probability of being maintained. Mutations that modify structure are expected to particularly impact fitness, and often detectably influence observed patterns of mutations.

Nucleotide coevolution is one such example, whereby pairs of nucleotides within biologically functional nucleic acid secondary structures exhibit evidence of coevolution that is consistent with the maintenance of canonical base-pairing. PINNACLE, the first of two models developed in this dissertation, is a sequence evolution model that assesses the rates of mutation associated with base-paired sites in alignments of DNA or RNA sequences. PINNACLE is able to fully account for an unknown secondary structure, and in doing so can be used to predict a secondary structure shared amongst an alignment of sequences. PINNACLE was used to infer rates of coevolution associated with GC, AU (AT in DNA), and GU (GT in DNA) dinucleotides in non-coding RNA alignments, and single-stranded RNA and DNA virus alignments. Strong evidence was found for GU dinucleotides being selectively favoured at base-paired sites in non-coding RNA and RNA virus alignments, with slight evidence for GT dinucleotides being selectively favoured at base-paired sites in DNA virus alignments. The strength of coevolution at base-paired sites in a SHAPE-MaP-determined HIV-1 NL4-3 RNA secondary structure and a corresponding alignment containing large numbers of HIV group 1M sequences was also measured, finding statistically significant correlations between the experimentally-determined SHAPE-MaP pairing scores and the inferred degrees of coevolution.

The second model developed in this dissertation, ETDBN, differs from typical evolutionary models in that it models the structural evolution of proteins in addition to their sequence evolution. ETDBN uses a dihedral angle representation of protein structure, and models the evolutionary trajectory between a pair of protein structures using an angular diffusion process on the two-dimensional torus. ETDBN is trained on a large database of proteins, and is parameterised such that it can learn the dependencies between sequence and structural evolution. The model has interpretable parameters, and is comparatively more realistic than previous stochastic models of protein structure evolution. Using the trained model, we were able to identify apparent sequence-structure evolutionary motifs present in numerous homologous protein pairs. The generative nature of our model enabled us to evaluate its validity and to infer various quantities, such as protein structures or alignments from sequences of amino acids, dihedral angles, secondary structure labels, or any combination thereof.

Author contributions

All work in this thesis has been composed by myself, except where otherwise indicated here, or acknowledge via citation. Chapter 2 is expected to be published as a co-authored paper with me as first author, whereas Chapter 3 has already been published as a co-authored paper (Golden *et al.*, 2017). The paragraph 'Dihedral angle evolution' in Section 3.2.4 was written by Eduardo García-Portugués and published in García-Portugués *et al.* (2017), and has been included for the sake of completeness. Figure 3.8 was produced by Eduardo García-Portugués with data provided by myself. Section 3.5.5 has been adapted from unpublished text written by Thomas Hamelryck. The HIV alignment datasets used in Chapter 2 were provided by Ben Murrell. All code implementations are my own.

Contents

1 Introduction **1**
1.1 Overview of evolutionary modelling 1
1.1.1 Homology 1
1.1.2 Sequence models of evolution 2
1.1.3 Stochastic processes 5
1.1.4 Pairwise to phylogeny: Felsenstein's algorithm 8
1.1.5 Practical applications 10

2 PINNACLE: Probabilistic INference of Nucleic Acid CoevoLution **12**
2.1 Introduction 13
2.2 Methods 16
2.2.1 The Muse 1995 model 16
2.2.2 Differentiating between dinucleotide substitutions 18
2.2.3 Stationarity and time-reversibility 19
2.2.4 Modelling site-to-site rate variation 19
2.2.5 Modelling variable degrees of coevolution 21
2.2.6 Testing neutrality of coevolution 22
2.2.7 Priors 22
2.2.8 Computer representations of secondary structure 25
2.2.9 Likelihood 27
2.2.10 Prior over RNA secondary structures 27
2.2.10.1 The KH99 grammar 28
2.2.10.2 The inside-outside algorithm 29
2.2.10.3 Iterative implementation and parallelisation schemes 30
2.2.11 Dinucleotide column likelihoods 31
2.2.12 Sampling secondary structure configurations 32
2.2.13 Bayesian posterior inference 36
2.2.14 Maximum likelihood inference 38
2.2.15 Posterior base-pairing and unpairing probabilities 38
2.2.16 Consensus structure prediction 38
2.2.17 Proposal variances 39
2.2.18 Partial likelihood caching 39
2.2.19 Datasets 40
2.2.19.1 Representative subset selection 41
2.2.19.2 Linearisation of circular DNA viruses 42
2.2.20 Measures of predictive accuracy 42
2.2.20.1 Precision, recall, and F-score 43
2.2.20.2 Matthews correlation coefficient 44

2.2.20.3 Mountain metric . . . 44
2.2.21 Combinatorics of RNA secondary structure . . . 46
2.2.22 Information Entropy . . . 47
2.2.23 Column permutations . . . 50
2.2.24 Spearman's rank correlation coefficient and confidence intervals 50
2.2.25 Substructure ranking . . . 50
2.2.26 Visualisation of secondary structures . . . 51
2.3 Results and Discussion . . . 53
2.3.1 Column permutation benchmarks . . . 53
2.3.2 Partial site caching yields computational reductions . . . 55
2.3.3 CPU and GPU timing benchmarks . . . 55
2.3.4 Benchmarks of RNA secondary structure prediction . . . 58
2.3.5 The role of GU and GT base-pairs in single-stranded RNA and DNA . . . 60
2.3.6 Relative coevolution rates . . . 62
2.3.7 Degrees of coevolution are correlated with experimental SHAPE-MaP quantities . . . 78
2.3.8 Ranking of substructures . . . 84
2.4 Concluding remarks . . . 93
2.5 Future challenges . . . 96
2.5.1 Dual codon and RNA constraints . . . 96
2.5.2 Evolution on secondary structure . . . 97

3 A generative angular model of protein structure evolution 99
3.1 Introduction . . . 100
3.2 Methods . . . 105
3.2.1 Overview of evolutionary model . . . 105
3.2.2 Observation types . . . 106
3.2.3 Model structure . . . 106
3.2.4 Stochastic processes: modelling evolutionary dependencies . . 107
3.2.5 Site-classes: constant evolution and jump events . . . 112
3.2.6 Pairwise statistical alignment: modelling insertions and deletions with neighbouring dependencies . . . 116
3.2.7 Calculation of angular distances . . . 120
3.2.8 Identification of evolutionary motifs encoding jump events . . 120
3.2.9 Training and test datasets . . . 121
3.2.10 Model training . . . 121
3.2.11 Parallelisation of model training . . . 122
3.2.12 Model selection . . . 122
3.3 Results and Discussion . . . 124
3.3.1 Model selection . . . 124
3.3.2 Stationary distributions over dihedral angles capture the empirical distribution . . . 124
3.3.3 Estimates of evolutionary time from dihedral angles are consistent with estimates from sequence . . . 126
3.3.4 Using sequence and structure reduces the uncertainty associated with estimates of evolutionary time . . . 127

3.3.5 The relationship between evolutionary time and angular distance is adequately modelled . . . 128
3.3.6 Structural inference . . . 130
3.3.7 Evolutionary hidden states reveal common evolutionary motifs 132
3.3.8 Dihedral angle transition densities are informed by amino acids changes . . . 136
3.3.9 Using dihedral angles for alignment . . . 140
3.4 Concluding remarks . . . 145
3.5 Future challenges . . . 147
3.5.1 Pairwise to phylogeny . . . 147
3.5.2 Site-class switching and context-dependence . . . 149
3.5.3 Error modelling . . . 149
3.5.4 Selection analyses . . . 150
3.5.5 Global structural dependencies . . . 150

4 Perspectives **153**

Bibliography **157**

A PINNACLE **171**

B ETDBN **194**

Chapter 1

Introduction

1.1 Overview of evolutionary modelling

1.1.1 Homology

A set of biological entities, such as a set of DNA sequences or a set of protein structures, are termed *homologous* if they descend from a common ancestor. The presence of shared ancestry will typically lead to evolutionary dependencies (similarities) amongst the entities in question. For example, a pair of proteins that have evolved from a common ancestor via a process of mutation and selection will likely share a degree of sequence and structural similarity (Figure 1.1). The strength of the evolutionary dependence, reflected in the degree of similarity, is assumed to be due to two major factors: (i) the time since the common ancestor and (ii) the rate of evolution.

Failing to account for evolutionary dependencies can lead to misleading inferences (Felsenstein, 1985). For example, strong signals of structural conservation may be wrongly attributed to selective maintenance of structural features due to their supposed functional importance, whereas in reality the structural similarities may simply be due to the close evolutionary relatedness of the proteins being analysed. On the other hand, accounting for evolutionary dependencies allows information from homologous proteins to be incorporated in a statistically principled manner. This can lead to more accurate inferences, such as the prediction of a protein structure from

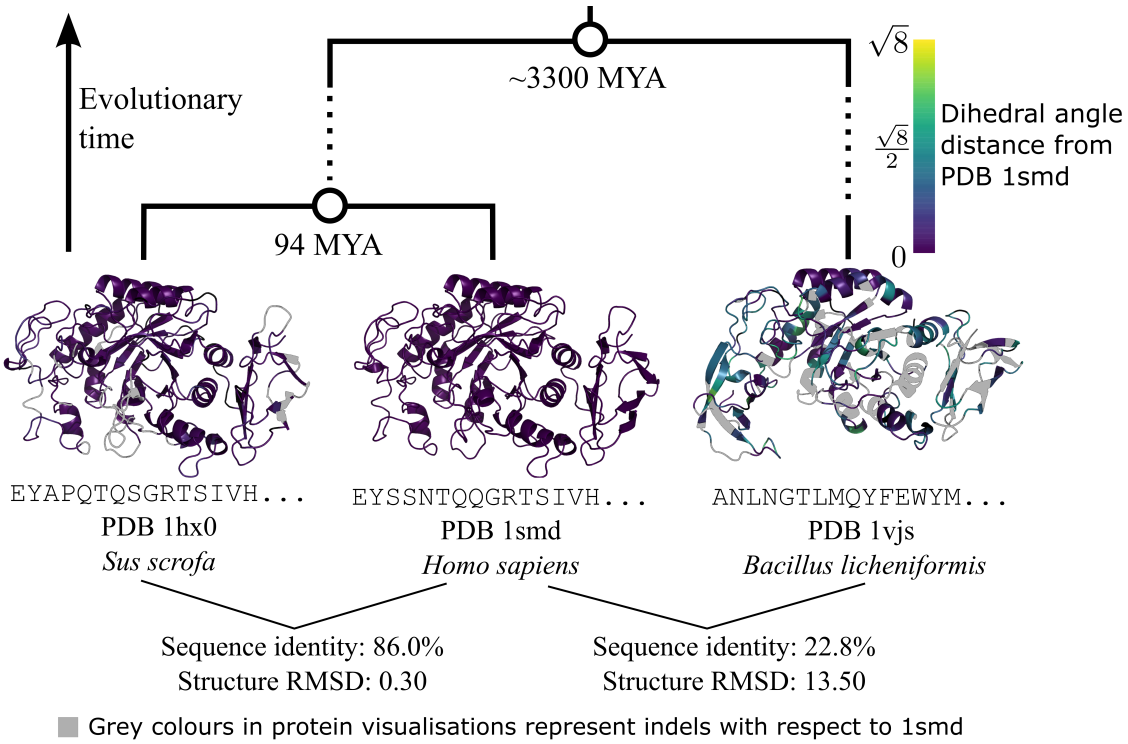


Figure 1.1: A phylogenetic tree relating three homologous amylase proteins, comparing their sequence identity and structural divergence. Overlaid on the cartoon representations of PDB 1hx0 (left) and PDB 1vjs (right) are the dihedral angle distances (as defined in Section 3.2.7) compared to PDB 1smd (centre) at each aligned amino acid position. The degree of structural divergence (reported as 'Structure RMSD') is measured using the Root Mean Squared Deviation (RMSD) between the aligned three-dimensional atomic coordinates of the corresponding PDB (Protein Data Bank; Joosten *et al.* (2011)) files. The wild boar (*Sus scrofa*) and human (*Homo sapiens*) amylases share a common ancestor 94 Million Years Ago (MYA). The third amylase from *Bacillus licheniformis* shares a common ancestor with the other two species approximately 3300 MYA. The common ancestors are unobserved and represented by white nodes. Notice that the dependencies are weaker the further back in time two proteins share a common ancestor.

a homologous protein sequence and structure, known as *homology modelling* (Arnold *et al.*, 2006).

1.1.2 Sequence models of evolution

The most commonly compared homologous biological entities are sequences. This is in part due to the large number of sequences available in public databases compared to structures. Figure 1.2 gives a comparison of the number of protein sequences versus structures available in public databases – in 2017 there were approximately

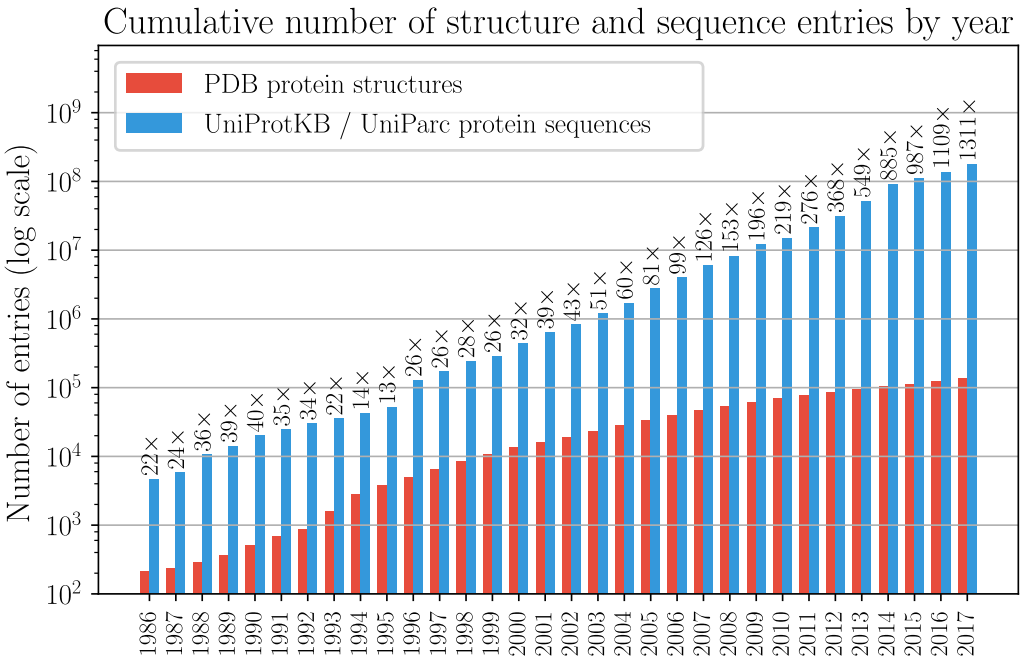


Figure 1.2: Cumulative number of protein structure and sequence entries by year in the Protein Data Bank (Joosten *et al.*, 2011) and UniProtKB / UniParc database (UniProt Consortium and others, 2017), respectively. Indicated above each year is the ratio of the cumulative number of sequences entries compared to the cumulative number of structure entries. Note the log scale on the y-axis.

1,311 times more protein sequences available than protein structures. Another reason may be that sequences are easier to compare than structures, with many more models of sequence evolution available than models of structure evolution.

Sequences consist of ordered strings of characters such as: amino acids in the case of protein sequences, nucleotides in DNA or RNA sequences, or groups of three nucleotides (known as *codons*) in the case of messenger RNAs (mRNAs).

One way to compare two sequences is to calculate a distance between them (Yang, 2014). A distance can be as simple as counting the number of character differences (known as the *Hamming distance*); however, this particular distance has the shortcoming that it underestimates the number of substitutions when multiple substitutions occur at the same site. Most distances attempt to approximate the mean number of substitutions per site that have occurred since the common ancestor of two sequences. Such distances are widely used, and have been generalised upon in order to account for various biological factors, such as the occurrence of multiple sub-

stitutions per site, and differing rates of mutation. For example, A to G mutations (an example of a transition) tend to occur more frequently in DNA than A to C mutations (an example of a transversion).

There also exist probabilistic models for comparing sites within two sequences. Typically these take the form of a *continuous-time Markov chains* (CTMC). A CTMC is a stochastic process giving a probability distribution over all possible trajectories of intermediate character states and the times at which they occur linking two *homologous characters* ending in each sequence at a particular site. The term "homologous characters" refers to two or more characters that have evolved along a common trajectory solely via substitutions, and not via the the insertion or deletion of characters – such homologous characters are identified using an alignment algorithm (more on this in Section 3.2.6).

By considering all possible trajectories between homologous characters, a CTMC accounts for multiple substitutions. CTMCs are flexible in that differing rates of substitution can be specified, and the probabilistic nature of CTMCs allows these rates to be inferred if not known *apriori*. Whilst CTMCs have attractive modelling properties, they can be computationally expensive. Many non-probabilistic methods of sequence comparison which preceded probabilistic models remain in use, for example: maximum parsimony (which attempts to explain the data using the minimal number of substitutions; Cavalli-Sforza and Edwards (1967)) or minimum evolution (which attempts to minimise the sum of the distances using least squares; Rzhetsky and Nei (1993)).

Non-probabilistic methods tend to be much more computationally efficient than CTMCs; however, improvements in computational hardware and software in recent decades means this is no longer a limiting factor. Unlike CTMCs, non-probabilistic methods do not model the underlying generative process (*i.e.* the evolutionary trajectories), and therefore arguably capture much less realism. The next section gives a general mathematical description of stochastic processes, including CTMCs, and some important properties of them.

1.1.3 Stochastic processes

Stochastic processes are widely-used in modelling the evolution of biological entities. A stochastic process describes a sequence of random variables $\{X(t) : t \in T\}$, where T is an index set most often chosen to represent time and taking values amongst the real numbers. The times in T can be discrete-valued, but in the case of evolutionary models used in this dissertation are non-negative real numbers. The random variables can take values in an arbitrary state space S; however, this dissertation considers only random variables that are discrete-valued (e.g. nucleotides, dinucleotides, secondary structure labels, and amino acids) and continuous vector-valued (pairs of angles).

Markov chains An example of a stochastic process is a *Markov chain*. A discrete-time Markov chain describes a sequence of discrete random variables $X_1, X_2, \ldots, X_t$ occurring at discrete times $1, \ldots, t$ and taking on values in a discrete state space S. The *Markov property* implies that each X_i only depends on the previous X_{i-1}, *i.e.* $p(X_i|X_{i-1}, \ldots, X_1) = p(X_i|X_{i-1})$. If a discrete-time Markov chain is *time-homogeneous* (the parameters do not depend on the time index i) and has a finite state space of size N the probability, $p(X_i|X_{i-1})$, of transitioning from one state, X_{i-1}, to another, X_i, can be represented by a $N \times N$ transition probability matrix P. The probability of starting in the initial state, $p(X_1)$ is given by an initial distribution vector, π, of length N.

Continuous-time Markov chains A CTMC is a Markov chain specified by an initial distribution vector, π_0, and a transition rate matrix Q; however, unlike a discrete-time Markov chain, events occur after exponentially distributed random times along the real line $\mathbb{R}$. The off-diagonal entries of Q, q_{ij} $(i \neq j)$, specify the rates of transition from state i to j. The diagonal entries of Q, q_{ii}, specify the holding times for each state i, which are exponentially distributed with rate parameter $-q_{ii}$. The diagonal entries, q_{ii}, are chosen such that each row of Q sums to zero, *i.e.* $q_{ii} = -\sum_{j \neq i} q_{ij}$. CTMCs are widely-used in sequence models of evolution and are referred to as *substitution models*. A transition probability matrix, $P(t)$ can be computed from the

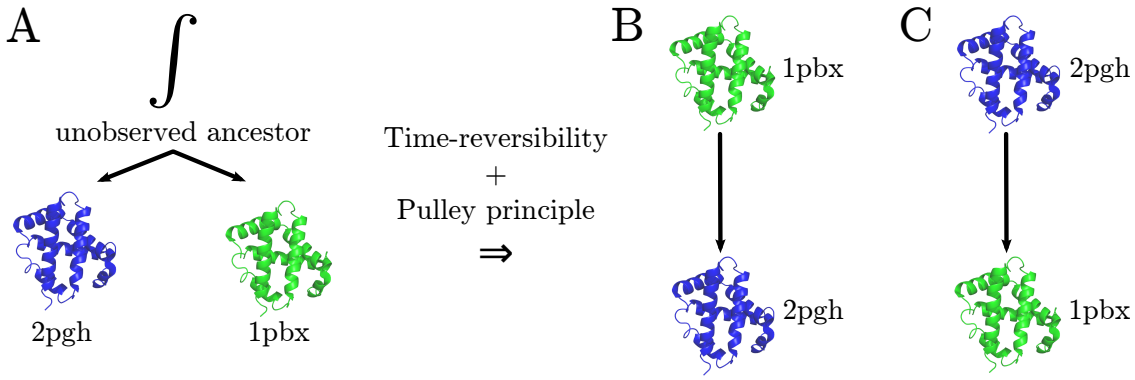


Figure 1.3: Illustration of the pulley principle in a pairwise phylogeny. To calculate the likelihood of two homologous proteins it is usually necessary to integrate out the sequence and/or structure of the unobserved ancestor at the root of the tree (A). However, under a time-reversible stochastic process, the pulley principle (*i.e.*, assuming that the protein at the root of the tree is drawn from the stationary distribution of the process) allows any of the two observed proteins to be set as the root (B or C) without altering the likelihood. This has the advantage of avoiding a costly integration.

rate matrix Q using the matrix exponential: $P(t) = e^{Qt}$ (Moler and Van Loan, 2003). Most models of sequence evolution make use of the matrix exponential to compute transition probabilities in order to perform inference.

Stationary distribution The stationary distribution of a Markov chain, typically denoted π, assuming it exists and is unique, is a distribution over states that if the Markov chain starts at t_0 in the stationary distribution, then at any time later $t \geq t_0$ the random variable, X_t, will be distributed according to π.

Time-reversibility A common feature of many stochastic processes used to model molecular evolution is *time-reversibility*. Time-reversibility is a mathematical property that, roughly speaking, implies that the statistical properties of a stochastic process are the same even when the stochastic process is reversed. Stated mathematically a Markov process with stationary distribution, π, and transition rates, q_{ij}, is time-reversible if:

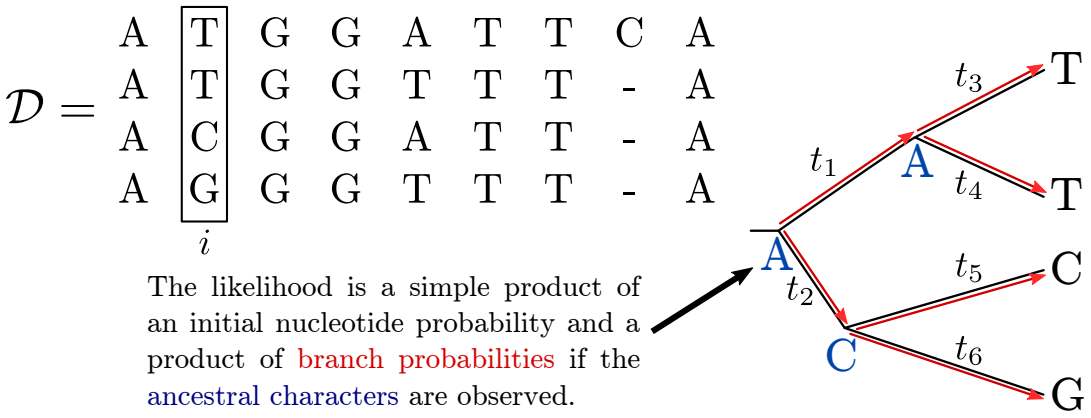
$$\pi_i q_{ij} = \pi_j q_{ji} \tag{1.1}$$

Time-reversibility provides computational advantages. For example, in a pairwise evolutionary model, time-reversibility together with the *pulley principle* (Felsenstein, 1981) allows the phylogenetic tree to be re-rooted on an observed protein (where the root is assumed to be drawn from the stationary distribution of the stochastic process), avoiding a potentially costly integration over an unobserved ancestral protein (see Figure 1.3 for an illustration).

Exchangeability terms for CTMCs Time-reversibility of a CTMC provides a useful symmetry that, relative to a non-reversible model, simplifies parameterisation by reducing the number of parameters to a set of symmetric terms and stationary probabilities. Let $s_{ij} = q_{ij}/\pi_j$, given time-reversibility in (1.1) note that:

$$s_{ij} = s_{ji} \tag{1.2}$$

These symmetric terms, s_{ij} and s_{ji}, are referred to as the *exchangeability* between i and j. Exchangeability is later used to simplify the parameterisation of one our models (Section 3.2.4).



$$p(\mathcal{D}_i|\mathcal{T}) = \pi_{\mathrm{A}} \cdot p(\mathrm{A}\to\mathrm{A}|t_1)p(\mathrm{A}\to\mathrm{T}|t_3)p(\mathrm{A}\to\mathrm{T}|t_4)p(\mathrm{A}\to\mathrm{C}|t_2)p(\mathrm{C}\to\mathrm{C}|t_5)p(\mathrm{C}\to\mathrm{G}|t_6)$$

Figure 1.4: An example dataset showing the likelihood calculation for an alignment column of homologous characters, $\mathcal{D}_i$ conditioned on having observed topology and branch lengths of a phylogenetic tree, $\mathcal{T}$, and the ancestral characters up to the root of the tree.

1.1.4 Pairwise to phylogeny: Felsenstein's algorithm

A stochastic process, such as a CTMC, gives a probability distribution over a pair of homologous observations (typically two sequences). In practice, we often have multiple homologous sequences related by a phylogenetic tree. A key algorithm used for performing inference under models of sequence evolution when multiple homologous sequences have been observed is Felsenstein's algorithm (Algorithm 1; Felsenstein (1981)). Felsenstein's algorithm provides an efficient way of calculating the probability of observing a set of homologous character states related by a phylogenetic tree under a particular substitution model.

Typically only characters at the leaves of a phylogenetic tree are observed and not the ancestral characters at the internal nodes, which implies that a sum over all combinations of unobserved characters at internal nodes is required. Doing so exhaustively scales exponentially: $\mathcal{O}|S|^{|I|}$, where $|S|$ is the size of the state space (four in the case of a nucleotide substitution models) and $|I|$ is the number of internal nodes. Figure 1.4 shows the likelihood calculation conditioned on a single combination of ancestral states, where each branch can be treated as conditionally independent and transition probabilities multiplied along the tree – clearly this is inefficient when having to exhaustively sum over all possible combinations of character states at the internal nodes. Fortunately, Felsenstein's algorithm provides an efficient way of doing it in $\mathcal{O}(|I||S|^2)$ computational steps. Note that Felsenstein's algorithm can not be used for continuous state evolution, only discrete state evolution.

Algorithm 1 Felsenstein's algorithm

ITERATION

Assuming a rooted binary tree $\mathcal{T}$, with root node: N_{root}. Start at the root node and recursively compute the likelihood, $p(N_k = a|\mathcal{T})$, for every character, a, in the nucleotide alphabet $\Lambda = \{\text{A,C,G,T}\}$ as follows:

If the likelihood for node N_k and character a has previously been calculated, return the stored value: $p(N_k = a|\mathcal{T})$.

Else if N_k is a leaf node, calculate, store, and return:

$$p(N_k = a|\mathcal{T}) = \begin{cases} 1 & \text{if } N_k = a \\ 0 & \text{otherwise} \end{cases}$$

Otherwise if N_k is an ancestral node, calculate, store, and return:

$$p(N_k = a|\mathcal{T}) = \overbrace{\sum_{b \in \Lambda} \overbrace{p(a \to b|t_{k,\text{left}})}^{\text{trans. prob.}} \times \overbrace{p(N_{\text{k,left}} = b|\mathcal{T})}^{\text{recursive}}}^{\text{left child}} \times \overbrace{\sum_{c \in \Lambda} \overbrace{p(a \to c|t_{k,\text{right}})}^{\text{trans. prob.}} \times \overbrace{p(N_{\text{k,right}} = c|\mathcal{T})}^{\text{recursive}}}^{\text{right child}}$$

where $p(a \to b|t_{k,\text{left}})$ and $p(a \to c|t_{k,\text{right}})$ are the probabilities of transitioning between the designated pairs of nucleotides ($a \to b$ and $a \to c$, respectively) in the specified evolutionary times ($t_{k,\text{left}}$ and $t_{k,\text{right}}$, respectively).

TERMINATION

The character at the root of the tree is assumed to be drawn from an initial distribution π (typically the stationary distribution of the process), hence the likelihood of a particular alignment column, $\mathcal{D}_i$, is:

$$p(\mathcal{D}_i|\mathcal{T}) = \sum_{a \in \Lambda} p(N_{root} = a|\mathcal{T})\pi(a)$$

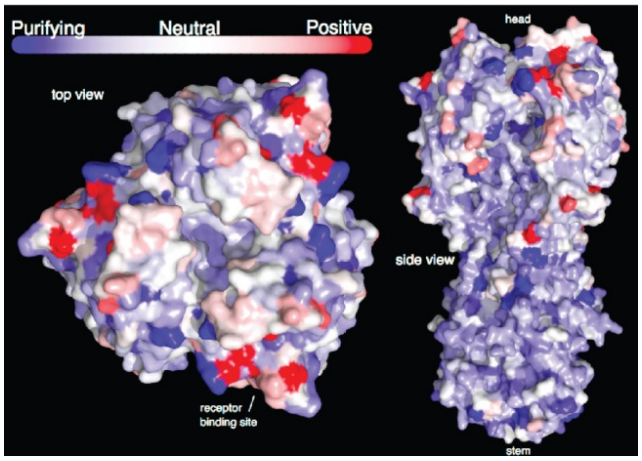


Figure 1.5: Adapted [reprinted] from Murrell *et al.* (2013). Selection values inferred using the FUBAR method overlaid on a influenza virus hemagglutinin protein (PDB 3ZTJ), displayed from two viewpoints. Red regions with stronger diversifying selection are likely involved in immune escape. These primarily occur on the "head" of the protein, with mostly purifying selection on the membrane proximal stem.

1.1.5 Practical applications

The methodologies described in this overview have led to numerous practical applications in the field of evolutionary modelling. The most prominent application that leaps to mind is the use of codon substitution models that differentiate between nucleotide substitutions within codons that do not change the encoded amino acid (synonymous changes), and those that do change the encoded amino acid (non-synonymous changes) (Muse and Gaut, 1994). Because changes at the amino acid level are likely to modify the structure of a protein, and hence its function, selection mainly operates at the level of proteins, allowing synonymous substitutions to act as baseline or *neutral* rate of evolution that can then be compared to the rates of non-synonymous changes. This enables one to make statements about whether selection is acting to preserve the amino acid sequence (purifying selection) or acting to diversify the amino acid sequence (positive selection). Figure 1.5 gives an example from a popular method, FUBAR (Murrell *et al.*, 2013), that shows how positive selection detected in regions of an Influenza virus protein structure may be evidence of viral immune escape.

Other applications include, but are not limited to: (i) the inference of phylogenetic

trees (Woese and Fox, 1977), (ii) ancestral state reconstruction (which, for example, can allow the reconstruction of ancient proteins from modern drug target proteins, helping elucidate a drug's mechanism; Wilson *et al.* (2015)), (iii) phylogeography (which can reveal the origin and predict the spread of pathogens; Monjane *et al.* (2011)), (iv) motif finding (used for the detection of common binding motifs within DNA, but where underlying phylogenetic correlations may inflate the reported significance of those motifs, if unaccounted for; Siddharthan *et al.* (2005); Satija *et al.* (2009)), and (v) detection of episodic directional selection (periods of evolution where amino acid substitutions have a significant tendency to occur in a directional manner. This is particularly relevant in cases where a patient has received a drug treatment following infection by a virus, and the viral quasispecies accumulates drug resistance mutations; Murrell *et al.* (2012)).

Chapter 2

PINNACLE: Probabilistic INference of Nucleic Acid CoevoLution

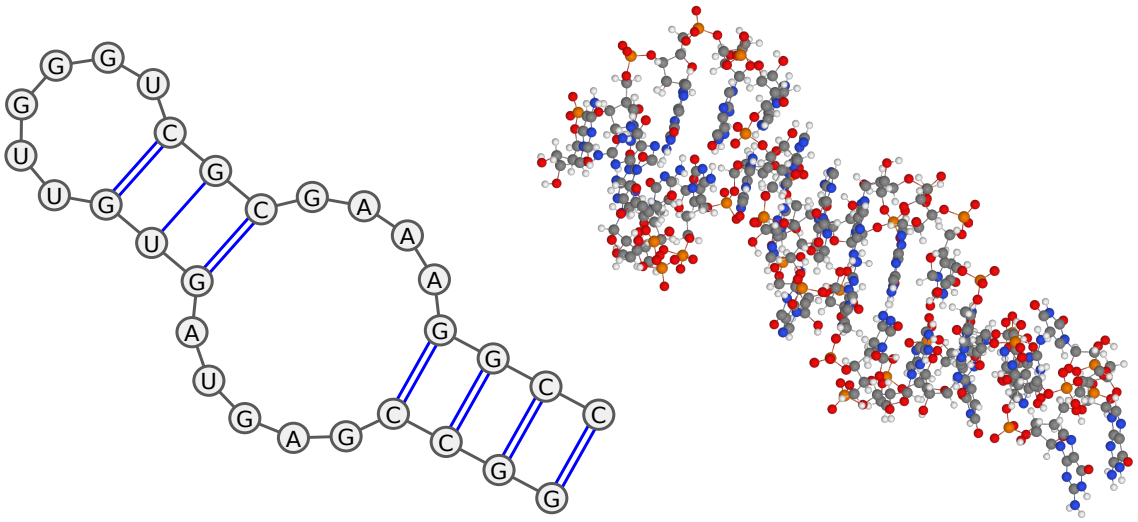


Figure 2.1: On the left is a graphical representation of the HCV-1B IRES domain IIID minimum free energy secondary structure predicted using UNAfold (Markham and Zuker, 2008) and visualised using VARNA (Darty *et al.*, 2009). Hydrogen bonding interactions are indicated using blue lines joining pairs of nucleotides. On the right is a tertiary structure representation of the same RNA molecule obtained from the Protein Data Bank (PDB ID 1F84; Lukavsky *et al.* (2000)) and visualised using Jmol (The Jmol team, 2013), providing a graphical representation of the three-dimensional co-ordinates of each atom.

2.1 Introduction

The primary role of nucleic acid molecules, such as DNA (deoxyribonucleic acid) and RNA (ribonucleic acid), is encoding genetic information for storage and transfer; however, both types of molecules can form structures with additional functions (Mattick, 2003). DNA is ordinarily thought of as a double-stranded molecule forming the now iconic double helical configuration (Watson and Crick, 1953), although many viral genomes consist entirely of single-stranded DNA or RNA molecules. Such single-stranded nucleic acid molecules are far less constrained than double-stranded nucleic acid molecules in the variety of structures that they can form, and hence can have diverse functionality. For example, the Rev response element (RRE) within the single-stranded HIV RNA genome plays a crucial role in the regulation of HIV virion expression by binding the HIV Rev protein, allowing transfer of HIV genomes from the nucleus to the cytoplasm where translation and virion packaging occur (Heaphy *et al.*, 1990; Daugherty *et al.*, 2010).

The structures that nucleic acid molecules form are commonly referred to as their secondary or tertiary structures. Whereas secondary structure is defined as the set of hydrogen bonding interactions between the constituent bases of a nucleic acid molecule; tertiary structure is defined as the arrangement of the constituent atoms in three-dimensional space (Figure 2.1). Despite being less informative than tertiary structure, the formation of base-pairings comprising the secondary structure of an RNA molecule is thought to guide the formation of the final tertiary structure configuration (Banerjee *et al.*, 1993; Mathews *et al.*, 1999; Tinoco Jr and Bustamante, 1999).

Both computational (Markham and Zuker, 2008; Sükösd *et al.*, 2012; Bernhart *et al.*, 2006) and hybrid experimental-computational techniques (Wilkinson *et al.*, 2006) for secondary structure prediction exist. However, even if the secondary structure of an RNA sequence can be accurately determined, this does not immediately say anything about the potential functional or biological importance of the identified structure. Given that many RNA secondary structures are known to have specific biological functions, it is expected that these structures might detectably impact the patterns of evolution in the genomes in which they occur.

One evolutionary signal that can identify selectively maintained genomic secondary structures is nucleotide coevolution. Nucleotide coevolution is expected to occur between base-paired nucleotides within these structures (Eddy and Durbin, 1994; Tuplin *et al.*, 2002; Cheng *et al.*, 2012). Many pairs of nucleotides within RNA molecules exhibit evidence of coevolution, such that whenever a substitution occurs in one partner of the pair, a complementary substitution occurs in the other partner in a manner that is consistent with the selective maintenance of canonical base-pairing (Cheng *et al.*, 2012). For the purposes of this study, we consider the canonical RNA base-pairs to be the two Watson-Crick base-pairs, GC and AU, and the weaker GU wobble base-pair (GC, AT, and GT base-pairs in DNA, respectively).

Methods for detecting coevolution, such as mutual information (Eddy and Durbin, 1994; Lindgreen *et al.*, 2006), can be used to aid the computational inference of sec-

ondary structures. Accordingly, some RNA comparative secondary structure prediction approaches, such as PPfold (Sükösd *et al.*, 2012), use information about coevolving nucleotides inferred from sequence alignments to more accurately predict secondary structures. Conversely, within a given secondary structural element, evidence that paired bases are coevolving is evidence of the functional importance of that element (Tuplin *et al.*, 2002; Cheng *et al.*, 2012; Muhire *et al.*, 2014).

Standard approaches for measuring coevolution (or more accurately: covariation), such as mutual information (Lindgreen *et al.*, 2006), are non-evolutionary in that they do not take into account the phylogenetic relationships of the sequences being analysed. Founder substitutions can, by chance, induce correlations between bases in a large number of observed species, which may be mistaken for strong evidence of coevolution if the phylogeny is not accounted for. Continuous-time Markov chains (CTMCs) provide a statistical framework for modelling of both phylogenetic relationships and underlying substitution processes; however, such approaches typically come at a computational cost.

In this study, we extend the M95 (Muse, 1995) dinucleotide CTMC model of RNA base-pairing evolution and implement a new software tool called PINNACLE (Probabilistic INference of Nucleic Acid CoevoLution). In what follows, the underlying models and methodologies are detailed. Relative rates of coevolution amongst different base-pairs are measured, site-to-site variation is accounted for, site-specific rates of coevolution are measured, secondary structure is predicted, and substructures expected to have particular functional significance are identified.

2.2 Methods

2.2.1 The Muse 1995 model

Let $\mathbf{Q}$ be a 4×4 General Time Reversible (GTR; Tavaré (1986)) nucleotide matrix:

$$\mathbf{Q} = \begin{pmatrix} -(r_1\pi_{\mathrm{C}} + r_2\pi_{\mathrm{G}} + r_3\pi_{\mathrm{U}}) & r_1\pi_{\mathrm{C}} & r_2\pi_{\mathrm{G}} & r_3\pi_{\mathrm{U}} \\ r_1\pi_{\mathrm{A}} & -(r_1\pi_{\mathrm{A}} + r_4\pi_{\mathrm{G}} + r_5\pi_{\mathrm{U}}) & r_4\pi_{\mathrm{G}} & r_5\pi_{\mathrm{U}} \\ r_2\pi_{\mathrm{A}} & r_4\pi_{\mathrm{C}} & -(r_2\pi_{\mathrm{A}} + r_4\pi_{\mathrm{C}} + r_6\pi_{\mathrm{U}}) & r_6\pi_{\mathrm{U}} \\ r_3\pi_{\mathrm{A}} & r_5\pi_{\mathrm{C}} & r_6\pi_{\mathrm{G}} & -(r_3\pi_{\mathrm{A}} + r_5\pi_{\mathrm{C}} + r_6\pi_{\mathrm{G}}) \end{pmatrix} \tag{2.1}$$

Muse (1995) developed a pioneering model, henceforth referred to as the M95 model, that takes into account RNA base-pairing constraints by modelling pairs of nucleotides (dinucleotides) using a 16×16 matrix. The model generalises upon standard 4×4 nucleotide models, such as the GTR model, by introducing a coevolution parameter, λ, that is intended to capture substitutions at dinucleotide positions that are consistent with the maintenance of canonical RNA base-pairing. We define the set of canonical base-pairs as follows:

$$\mathcal{C} = \{\text{GC, CG, AU, UA, GU, UG}\} \tag{2.2}$$

Here we present a version of the original M95 dinucleotide model based on the GTR model Q and canonical base-pairs $\mathcal{C}$ defined above:

$$M_{ab} = \begin{cases} q_{ab^*}\lambda & \text{pairing gained } (a \notin \mathcal{C} \text{ and } b \in \mathcal{C}, \text{ e.g. a=AC} \rightarrow \text{b=AU}) \\ q_{ab^*} & \text{pairing unchanged } (a, b \notin \mathcal{C} \text{ or } a, b \in \mathcal{C}, \text{ e.g. a=AU} \rightarrow \text{b=GU}) \\ q_{ab^*}/\lambda & \text{pairing lost } (a \in \mathcal{C} \text{ and } b \notin \mathcal{C}, \text{ e.g. a=AU} \rightarrow \text{b=AC}) \\ 0 & \text{2 differences} \\ & \text{(e.g. a=AC} \rightarrow \text{b=GG; a=AU} \rightarrow \text{b=GC)} \end{cases} \tag{2.3}$$

Where q_{ab^*} is the entry of the GTR matrix Q corresponding to the nucleotide position that underwent a substitution, ab represents a dinucleotide substitution, and λ is a parameter capturing the degree of RNA coevolution, *i.e.* the degree to which canonical RNA base-pairing is evolutionary maintained ($\lambda > 1$) or disrupted ($\lambda < 1$). Note that $\lambda = 1$ represents the *neutral* case where each of the two positions in a dinucleotide is treated as evolving independently under the GTR model specified by Q.

Let π^{dinuc} denote a length 16 vector of dinucleotide frequencies. π^{dinuc} is the union of two mutually exclusive cases: $\pi^{\text{dinuc}} = \pi^{\text{unpaired}} \bigcup \pi^{\text{paired}}$, π^{unpaired} represents the case where the target dinucleotide d_{ij} is not in the set of canonical base-pairs ($d_{ij} \notin \mathcal{C}$), and π^{paired} represents the case where the target dinucleotide d_{ij} is in the set of canonical base-pairs ($d_{ij} \in \mathcal{C}$), respectively:

$$\pi^{\text{unpaired}}_{d_{ij}} = \kappa_1^{-1}\pi_i\pi_j \qquad \text{and} \qquad \pi^{\text{paired}}_{d_{ij}} = \kappa_1^{-1}\pi_i\pi_j\lambda^2,$$

Note that i and j correspond to the first and second positions of the target dinucleotide, respectively. Where π_i is the equilibrium frequency under the GTR model, Q, of the nucleotide in the first position of the target dinucleotide d_{ij}, and similarly π_j is the equilibrium frequency of the nucleotide in the second position. $\kappa_1 = 1 + 2(\pi_{\text{G}}\pi_{\text{C}} + \pi_{\text{A}}\pi_{\text{U}} + \pi_{\text{G}}\pi_{\text{U}})(\lambda^2 - 1)$ is a normalising constant that ensures the entries of π^d sum to one.

Note that within the set of canonical base-pairs, $\mathcal{C}$ (defined in Equation 2.2), there are three pairs of symmetrical base-pairs: {GC, CG}, {AU, UA}, and {GU, UG}. It is assumed that each base-pair within a symmetrical pair has the same fitness. This is a reasonable assumption as it treats the evolution of nucleotides towards the 5' end of the sequence the same as nucleotides towards the 3' end. From this point forward we assume this symmetry and refer to the three pairs of symmetrical base-pairs as the *three canonical base-pairs*.

In the formulation of the original M95 model in Equation 2.3 all three canonical base-pairs in the set $\mathcal{C}$ are treated as having equal fitness, this is not a realistic

assumption as there is good evidence that GU base-pairings in RNA, for example, are deleterious evolutionary intermediates relative to GC and AU (Rousset *et al.*, 1991). In light of this, in the next section we extend the M95 model such that the three canonical base-pairs are not constrained to have the same rate.

2.2.2 Differentiating between dinucleotide substitutions

In the original formulation of the M95 model, the same coevolution parameter, λ, was used to model the rate of substitutions effecting dinucleotides corresponding to canonical base-pairs. We extend the M95 model to differentiate between the three different canonical base-pairs, by introducing distinct coevolution rates (λ_{GC}, λ_{AU}, and λ_{GU}) for each of three different base-pairs (GC, AU, and GU, respectively). The extended rate matrix is given as follows:

$$M_{ab} = \begin{cases} q_{ab^*}\lambda_{\mathrm{GC}} & \text{GC pairing gained (e.g. a=AC} \rightarrow \text{b=GC)} \\ q_{ab^*}\lambda_{\mathrm{AU}} & \text{AU pairing gained (e.g. a=AC} \rightarrow \text{b=AU)} \\ q_{ab^*}\lambda_{\mathrm{GU}} & \text{GU pairing gained (e.g. a=GA} \rightarrow \text{b=GU)} \\ q_{ab^*}/\lambda_{\mathrm{GC}} & \text{GC pairing lost (e.g. a=GC} \rightarrow \text{b=GA)} \\ q_{ab^*}/\lambda_{\mathrm{AU}} & \text{AU pairing lost (e.g. a=AU} \rightarrow \text{b=AG)} \\ q_{ab^*}/\lambda_{\mathrm{GU}} & \text{GU pairing lost (e.g. a=GU} \rightarrow \text{b=GA)} \\ q_{ab^*}\lambda_{\mathrm{GU}}/\lambda_{\mathrm{GC}} & \text{GC to GU (e.g. a=GC} \rightarrow \text{b=GU)} \\ q_{ab^*}\lambda_{\mathrm{GC}}/\lambda_{\mathrm{GU}} & \text{GU to GC (e.g. a=GU} \rightarrow \text{b=GC)} \\ q_{ab^*}\lambda_{\mathrm{GU}}/\lambda_{\mathrm{AU}} & \text{AU to GU (e.g. a=AU} \rightarrow \text{b=GU)} \\ q_{ab^*}\lambda_{\mathrm{AU}}/\lambda_{\mathrm{GU}} & \text{GU to AU (e.g. a=GU} \rightarrow \text{b=AU)} \\ q_{ab^*} & \text{pairing unchanged (e.g. a=AC} \rightarrow \text{b=UC)} \\ 0 & \text{2 differences} \\ & \text{(e.g. a=AC} \rightarrow \text{b=GG; a=AU} \rightarrow \text{b=GC)} \end{cases} \tag{2.4}$$

Where q_{ab^*} is once again the entry of the GTR matrix Q corresponding to the nucleotide position that underwent a substitution. The corresponding stationary frequencies for this model are:

$$\pi_{d_{ij}}^{\text{unpaired}} = \kappa_2^{-1}\pi_i\pi_j \tag{2.5}$$

$$\pi_{\text{GC}}^{\text{dinuc}} = \kappa_2^{-1}\pi_{\text{G}}\pi_{\text{C}}\lambda_{\text{GC}}^2 \qquad \pi_{\text{AU}}^{\text{dinuc}} = \kappa_2^{-1}\pi_{\text{A}}\pi_{\text{U}}\lambda_{\text{AU}}^2 \qquad \pi_{\text{GU}}^{\text{dinuc}} = \kappa_2^{-1}\pi_{\text{G}}\pi_{\text{U}}\lambda_{\text{GU}}^2 \tag{2.6}$$

Where $\kappa_2 = 1 + 2[\pi_{\text{G}}\pi_{\text{C}}(\lambda_{\text{GC}}^2 - 1) + \pi_{\text{A}}\pi_{\text{U}}(\lambda_{\text{AU}}^2 - 1) + \pi_{\text{G}}\pi_{\text{U}}(\lambda_{\text{GU}}^2 - 1)]$

2.2.3 Stationarity and time-reversibility

We show that π^{dinuc} is the stationary distribution of $\mathbf{M}$ by verifying that:

$$\pi^{\text{dinuc}}\mathbf{M} = \mathbf{0}, \tag{2.7}$$

and that time-reversibility of $\mathbf{M}$ holds:

$$\pi_a^{\text{dinuc}} M_{ab} = \pi_b^{\text{dinuc}} M_{ba} \quad \forall_{ab} \tag{2.8}$$

where a and b represent dinucleotides. The conditions in (2.7) and (2.8) were verified using the symbolic math package, SymPy (Joyner *et al.*, 2012), as implemented in the `musesymbolic.py` script in Listing A.1.

2.2.4 Modelling site-to-site rate variation

In the above descriptions of the M95 and extended M95 models, the substitution rate was assumed to be the same for each of the two nucleotide positions within a dinucleotide, as well as across all dinucleotide sites. However, it is well-established that the rate of substitution can vary across sites and that failing to account for rate variability can lead to biased parameter estimates (Yang, 1996). Additionally, many of the datasets analysed in this thesis have coding regions, where it is expected that the third nucleotide position in each codon (the so-called 'codon wobble position')

will have relatively higher substitution rates associated with it, due to there being a lower chance of nucleotide substitutions modifying the encoded amino acid.

To model variable substitution rates across sites, the gamma distributed sites rate approach of (Yang, 1993, 1994) was implemented. The GTR model with gamma distributed sites, denoted GTR + Γ, is well-understood for single site substitution models; however, an extension to dinucleotide models had not been previously described. The modified M95 dinucleotide model with gamma distributed site rates, denoted M95 + Γ, is given below:

$$M_{ab}^{+\Gamma} = \begin{cases} \rho_k M_{ab} & \text{if dinucleotides } a \text{ and } b \text{ differ in the 1st position,} \\ \rho_l M_{ab} & \text{if dinucleotides } a \text{ and } b \text{ differ in the 2nd position,} \\ 0 & \text{2 differences} \end{cases} \tag{2.9}$$

ρ_k and ρ_l are independently and identically distributed according to a gamma distribution $f(\rho) = \frac{\beta^\alpha \rho^{\alpha-1}}{\Gamma(\alpha)\exp(\beta\rho)}$ with $\mathbb{E}_f[\rho] = \frac{\alpha}{\beta}$. Because β is a scale parameter the number of free parameters can be reduced by setting $\alpha = \beta$, which also restricts the mean of the gamma distribution to 1. A small α suggests rates differ significantly across sites, whereas a large α implies roughly equal rates amongst sites.

The conditional probability for a site pair $\mathcal{D}_{q,r}$ with rates ρ_k and ρ_l at sites $\mathcal{D}_q$ and $\mathcal{D}_r$, respectively, is given by $P(\mathcal{D}_{q,r}|\widehat{q,r}, \mathcal{T}, \boldsymbol{\theta}, \rho_k, \rho_l)$ and the marginal probability for a site pair $\mathcal{D}_{q,r}$ is given by:

$$P(\mathcal{D}_{q,r}|\widehat{q,r}, \mathcal{T}, \boldsymbol{\theta}) = \int_0^\infty \int_0^\infty f(\rho_k) f(\rho_l) P(\mathcal{D}_{q,r}|\widehat{q,r}, \mathcal{T}, \boldsymbol{\theta}, \rho_k, \rho_l) dk dl \tag{2.10}$$

Where $\widehat{q,r}$ indicates that nucleotide positions q and r are base-paired. Note that it is necessary to perform a double integration as it is assumed that substitution rates at each of two positions within a dinucleotide may differ. As is it not computationally feasible to evaluate the double integral in Equation 2.10 it is approximated using a discrete distribution (Yang, 1993, 1994). Let $F(\rho)$ represent the cumulative distribution function (CDF) of $f(\rho)$ and $F^{-1}(\rho)$ be the inverse CDF, if $p = F(\rho)$ then

$\rho = F^{-1}(p)$. The K rate classes are assumed to have equal proportions, i.e. $p_k = \frac{1}{K}$ for all k, and the ρ_k's are given by $\rho_k = \frac{F^{-1}(\frac{k}{K+1})}{K^{-1}\sum_{k_2} F^{-1}(\frac{k_2}{K+1})}$ (the denominator ensures that the discretised ρ_k's have a mean of one). Hence, the probability of the data at a particular dinucleotide site are approximated by:

$$P(\mathcal{D}_{q,r}|\widehat{q,r}, \lambda, \mathcal{T}, \boldsymbol{\theta}) \approx \sum_{k=1}^{K} \left[\sum_{l=1}^{K} p_k p_l P(\mathcal{D}_{q,r}|\widehat{q,r}, \lambda, \mathcal{T}, \boldsymbol{\theta}, \rho_k, \rho_l) \right] \tag{2.11}$$

Note that the double summation in Equation 2.11 increases the computational time by a factor of $\mathcal{O}(K^2)$ over a dinucleotide model assuming equal substitution rates across sites. For all analyses a discretisation of $K = 3$ was used.

2.2.5 Modelling variable degrees of coevolution

In the M95 model (2.3) the rate of coevolution, as specified by λ_{GC}, λ_{AU}, and λ_{GU}, was assumed to be the same at each base-paired site within a secondary structure $\mathcal{S}$ (defined in Section 2.2.8). However, it is expected that the strength of the selective forces maintaining canonical base-pairing will vary between base-paired sites. In this section, we extend the M95 model such that the *degree of coevolution*, denoted by $\eta_{q,r}$, is able to vary from base-paired site to base-paired site. $\eta_{q,r}$ is a parameter intended to capture the degree of coevolution at each base-paired site $\widehat{q,r}$ and is drawn independently for each base-paired site (described in the next section). $\eta_{q,r}$ acts to scale the three coevolution rates as follows:

$$\lambda_{\mathrm{GC}}^{q,r} = (\lambda_{\mathrm{GC}} - 1)\eta_{q,r} + 1 \quad \lambda_{\mathrm{AU}}^{q,r} = (\lambda_{\mathrm{AU}} - 1)\eta_{q,r} + 1 \quad \lambda_{\mathrm{GU}}^{q,r} = (\lambda_{\mathrm{GU}} - 1)\eta_{q,r} + 1 \tag{2.12}$$

where $\lambda_{\mathrm{GC}} \geq 1$, $\lambda_{\mathrm{AU}} \geq 1$, and $\lambda_{\mathrm{GU}} \geq 1$ are the base coevolution rates shared across all base-paired sites. This parametrisation was chosen so that $\lambda_{\mathrm{GC}}^{q,r} = \lambda_{\mathrm{AU}}^{q,r} = \lambda_{\mathrm{GU}}^{q,r} = 1$ when $\eta_{q,r} = 0$.

2.2.6 Testing neutrality of coevolution

To test the hypothesis that two nucleotide positions within a particular base-paired site are evolving neutrally, i.e. the substitutions at each of the two sites are behaving independently rather than actively favouring canonical base-pairing, we assume that the degree of coevolution, $\eta_{q,r}$, at each base-paired site is distributed as follows: $\eta_{q,r} = 0$ with probability w_η (the neutral, independent case), or $\eta_{q,r}$ is drawn from a M-way discretised gamma distribution with probability $1 - w_\eta$ (the dependent case). Note that $\eta_{q,r} \geq 0$ and therefore the case where substitutions are acting to disrupt canonical RNA base-pairing is not considered, i.e. the case where the coevolution parameters are between 0 and 1. For all analyses a discretisation of $M = 4$ was used, resulting in five total rate categories: one neutral category with probability w_η, and four positive categories each with probability $\frac{1-w_\eta}{4}$.

2.2.7 Priors

This section lists the priors (and hyperpriors) over parameters used in the most general version of the implemented model (the unconstrained model). Note that for some analyses we perform Bayesian inference, whereas for others we perform maximum likelihood (ML) inference. The priors over the parameters specified here are those used for Bayesian inference; however, we also indicate how the parameters are treated during ML inference (either estimated whilst ignoring the prior, or using the prior and fully marginalised).

The prior probability of neutral coevolution, w_η (estimated during ML inference whilst ignoring the prior), is drawn from a beta distribution:

$$w_\eta \sim \text{Beta}(2,2) \tag{2.13}$$

For each dinucleotide position q, r a Bernoulli random variable, $X_{q,r}$ (marginalised during ML inference), indicating neutral coevolution is drawn from a Bernoulli dis-

tribution with probability w_η:

$$X_{q,r} \sim \text{Bernoulli}(w_\eta) \tag{2.14}$$

A tied shape and rate parameter, c (estimated during ML inference whilst ignoring the prior), specifying a prior over the variation in coevolution rates is drawn from an exponential distribution:

$$c \sim \text{Exponential}(\frac{1}{10}) \tag{2.15}$$

For each dinucleotide position q, r the coevolution rate, $\eta_{q,r}$, is set equal to zero if $X_{q,r} = 1$, otherwise if $X_{q,r} = 0$ the coevolution rate, $\eta_{q,r}$ (marginalised during ML inference), is drawn from a discretised gamma distribution with shape c and rate c:

$$\eta_{q,r} = \begin{cases} 0 & \text{if } X_{q,r} = 1 \\ \eta_{q,r} \sim \text{DiscretisedGamma}_M(c, c) & \text{otherwise} \end{cases} \tag{2.16}$$

A tied shape and rate parameter, d (estimated during ML inference whilst ignoring the prior), specifying a prior over the variation in site-to-site substitution rates is drawn from an exponential distribution:

$$d \sim \text{Exponential}(\frac{1}{10}) \tag{2.17}$$

For each nucleotide position q a substitution rate, ρ_q (marginalised during ML inference), is drawn from a discretised gamma distribution with shape d and rate d:

$$\rho_q \sim \text{DiscretisedGamma}_K(d, d) \tag{2.18}$$

The prior probabilities of the four nucleotides, π (estimated during ML inference),

is given by a flat Dirichlet distribution:

$$(\pi_{\mathrm{A}}, \pi_{\mathrm{C}}, \pi_G, \pi_T) \sim \mathrm{Dirichlet}(1, 1, 1, 1) \tag{2.19}$$

The six symmetric nucleotide substitution rates (estimated during ML inference whilst ignoring the prior) are each drawn from exponential distributions:

$$\begin{aligned} q_{\mathrm{AC}} &\sim \mathrm{Exponential}(\frac{1}{10}) \\ q_{\mathrm{AG}} &\sim \mathrm{Exponential}(\frac{1}{10}) \\ q_{\mathrm{AT}} &\sim \mathrm{Exponential}(\frac{1}{10}) \\ q_{\mathrm{CG}} &\sim \mathrm{Exponential}(\frac{1}{10}) \\ q_{\mathrm{CT}} &\sim \mathrm{Exponential}(\frac{1}{10}) \\ q_{\mathrm{GT}} &\sim \mathrm{Exponential}(\frac{1}{10}) \end{aligned} \tag{2.20}$$

The three coevolution rates (estimated during ML inference whilst ignoring the prior) corresponding to the three canonical base-pairs are each drawn from exponential distributions right shifted by 1:

$$\begin{aligned} \lambda_{\mathrm{GC}} &\sim \mathrm{Exponential}(\frac{1}{10}) + 1 \\ \lambda_{\mathrm{AT}} &\sim \mathrm{Exponential}(\frac{1}{10}) + 1 \\ \lambda_{\mathrm{GT}} &\sim \mathrm{Exponential}(\frac{1}{10}) + 1 \end{aligned} \tag{2.21}$$

The secondary structure, $\mathcal{S}$ (marginalised during ML inference), is drawn from the KH99 SCFG (discussed in Section 2.2.10):

$$\mathcal{S} \sim \mathrm{KH99} \tag{2.22}$$

Finally, the phylogenetic tree, $\mathcal{T}$, relating the alignment of sequences, $\mathcal{D}$, for both Bayesian and ML inference is estimated in advance and fixed *apriori* using FastTree

(Price *et al.*, 2010). FastTree performs ML inference of the tree topology and branch lengths using a GTR+CAT nucleotide substitution model, where "CAT" denotes a categorical model with 20 rate categories geometrically spaced from 0.05 to 20.

2.2.8 Computer representations of secondary structure

An important step in computational modelling of nucleic acid secondary structures is specifying a suitable definition of secondary structure. We utilise the precise mathematical definition outlined in Moulton *et al.* (2000): a secondary structure for a nucleic acid molecule consisting of N nucleotides is a simple graph $\mathcal{S}$ specified by the vertex set $[N] \coloneqq \{1, \cdots, N\}$ and an edge set $B_{\mathcal{S}}$. Where each vertex in $[N]$ corresponds to a nucleotide and each edge in the edge set $B_{\mathcal{S}}$ corresponds to a base-pair. $\mathcal{S}$ is such that if $\{i, j\}, \{k, l\} \in B_{\mathcal{S}}$ with $i < j$ and $k < l$ then:

i. $i = k$ if and only if $j = l$, and

ii. $k \leq j$ implies that $i < k < l < j$

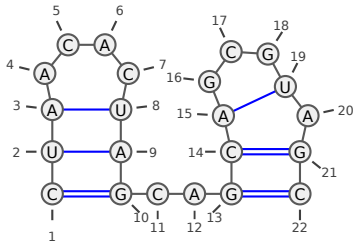
Vertices that are not contained within the edge set $B_{\mathcal{S}}$ are termed *unpaired*. Condition (i) implies that each vertex (nucleotide) belongs to at most one base-pair. Condition (ii) prevents *pseudoknotting*, *i.e.* non-nested base-pairs. Note that pseudoknotting is common in RNA structures (and DNA structures), but is excluded in many definitions of secondary structures as efficient algorithms exist for marginalising or maximising over secondary structures when assuming condition (ii).

Figure 2.2 provides two formats for representing secondary structure. The dot-bracket format (Figure 2.2A) is a natural and compact way of representing non-pseudoknotted secondary structures. Matching brackets represent base-paired nucleotide positions and dots represent unpaired (singled-stranded) nucleotide positions. On the other hand, the connect-table format (Figure 2.2D) is less compact but can represent secondary structures with arbitrary pseudoknotting. The first column represents each nucleotide position, the second column gives the nucleotide character at each position, and the fifth column specifies the position to which a nucleotide is

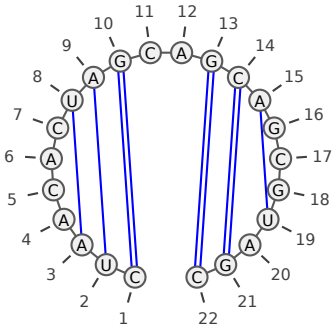
Secondary structure without pseudoknot

```
>Example
CUAACACUAGCAGCAGCGUAGC
(((....)))..(((...).))
```

A. Dot bracket notation file format



B. VARNA visualisation

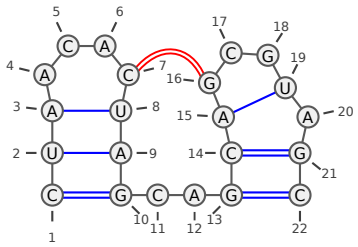


C. Circular visualisation

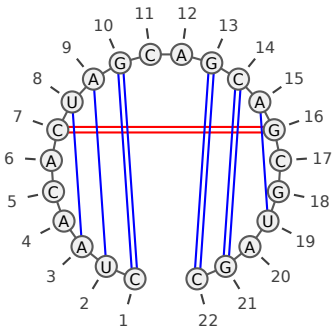
```
22 Example
1  C 0  2  10 1
2  U 1  3  9  2
3  A 2  4  8  3
4  A 3  5  0  4
5  C 4  6  0  5
6  A 5  7  0  6
7  C 6  8  16 7
8  U 7  9  3  8
9  A 8  10 2  9
10 G 9  11 1  10
11 C 10 12 0  11
12 A 11 13 0  12
13 G 12 14 22 13
14 C 13 15 21 14
15 A 14 16 19 15
16 G 15 17 7  16
17 C 16 18 0  17
18 G 17 19 0  18
19 U 18 20 15 19
20 A 19 21 0  20
21 G 20 22 14 21
22 C 21 23 13 22
```

D. Connect table (CT) file format

Secondary structure with an additional base-pair creating a pseudoknotted structure



E. VARNA visualisation



F. Circular visualisation

```
>Example
CUAACACUAGCAGCAGCGUAGC
(((...<)))..(((>..).))
```

G. Dot bracket notation file format with additional bracket type

Figure 2.2: Two examples of secondary structure representations. Above (A) is a dot bracket notation representation of a secondary structure and two corresponding visualisations (B and C). Below (D) is a connect table (CT) format representation and two corresponding visualisations (E and F) of the same secondary structure as above, but with an additional base-pairing (in red) resulting in a pseudoknotting. Also shown (G) is an extended dot bracket notation format with an additional bracket type, <>, that allows the structure to be represented unambiguously in dot bracket notation. Note how the overlapping bonds in the circular visualisation (F) demonstrate that the secondary structure is pseudoknotted.

paired, or 0 if the nucleotide is unpaired.

Note that I use the term *pseudoknotting*, as opposed to *pseudoknot*. There is common misconception, whereby a particular individual non-pseudoknotted element within a larger secondary structure is often deemed responsible for a pseudoknot, giving it unwarranted recognition. For example, the base-pairing coloured in red in the bottom of Figure 2.2. Mathematically two or more non-pseudoknotted elements within a larger secondary structure are required to create a pseudoknotting (e.g. Figure 2.2E), and it is therfore not possible to identify a particular element responsible.

2.2.9 Likelihood

Conditioned on a secondary structure, $\mathcal{S}$, unpaired nucleotide positions within $\mathcal{S}$, denoted by $\dot{q}$, and base-paired nucleotide positions within $\mathcal{S}$, denoted by $\widehat{q,r}$, are assumed to be independent. The likelihood of an alignment, $\mathcal{D}$, is given by a simple product of unpaired and paired site likelihoods:

$$p(\mathcal{D}|\mathcal{S},\mathcal{T},\boldsymbol{\theta}) = \overbrace{\prod_{\dot{q}\in\mathcal{S}} p(\mathcal{D}_q|\dot{q},\mathcal{T},\boldsymbol{\theta})}^{\text{unpaired}} \overbrace{\prod_{\widehat{q,r}\in\mathcal{S}} p(\mathcal{D}_{q,r}|\widehat{q,r},\mathcal{T},\boldsymbol{\theta})}^{\text{paired}} \tag{2.23}$$

Where $\mathcal{T}$ is a phylogenetic tree and Felsenstein's pruning algorithm (Felsenstein, 1981) was used to calculate both the single column likelihoods, $p(\mathcal{D}_q|\dot{q},\mathcal{T},\boldsymbol{\theta})$, for unpaired sites and the dinucleotide column likelihoods, $p(\mathcal{D}_{q,r}|\widehat{q,r},\mathcal{T},\boldsymbol{\theta})$, for base-paired sites. Base-paired sites were modelled using the M95 + Γ model, whereas unpaired sites were modelled using the GTR + Γ model that is nested within the M95 + Γ model.

2.2.10 Prior over RNA secondary structures

Up until now, we have assumed that a secondary structure $\mathcal{S}$ corresponding to a particular alignment, $\mathcal{D}$, is known *a priori* either through experimental or computational methods of structure prediction. However, it is also possible to treat the

secondary structure as unknown, by placing a prior probability distribution, $p(\mathcal{S})$, over secondary structures and marginalising $\mathcal{S}$.

One way of introducing a prior over secondary structures is by using a Stochastic Context Free Grammars (SCFG). A SCFG is a probabilistic extension of a context-free grammar (CFG). A CFG is a type of grammar that defines a set of rules for generating all possible strings in a given formal language. A SCFG extends this notion by assigning probabilities to each possible string in the given language. RNA SCFGs are SCFGs that give probability distributions over strings of base-paired and unpaired nucleotides representing RNA secondary structures. A typical RNA SCFG assigns probabilities to all strings of a given length that can be written in dot-bracket notation (see Section 2.2.8).

2.2.10.1 The KH99 grammar

The KH99 SCFG (Knudsen and Hein, 1999) was chosen as a prior over secondary structures for the PINNACLE model. The rules and associated probabilities are given as follows:

$$\mathcal{G}_{\text{KH99}} = \begin{array}{lcccccc} S & \rightarrow & \bullet & \text{or} & LS & \text{or} & (F) \\ & & 0.118 & & 0.869 & & 0.014 \\ \\ L & \rightarrow & \bullet & \text{or} & (F) & & \\ & & 0.895 & & 0.105 & & \\ \\ F & \rightarrow & (F) & \text{or} & LS & & \\ & & 0.788 & & 0.212 & & \end{array} \tag{2.24}$$

Note that S is the start symbol.

Double emission normal form The KH99 SCFG in Equation 2.24 is given in double emission normal form (Anderson *et al.*, 2012). Its main benefit is that it specifies not only that a nucleotide position is base-paired, but the partner nucleotide position to which it is base-paired. This allows for modular implementations of SCFG

algorithms of RNA secondary structure prediction, where a particular SCFG in double emission normal form can easily be replaced by another without having to modify the underlying inference algorithms – making it easy to replace one prior over secondary structures, such as the KH99 SCFG, with a different prior over secondary structures.

Double emission normal form captures three key features of secondary structure: branching, unpaired positions and base-paired positions. For any combination of non-terminals (T, U, V) only rules of the following form are allowed:

$$\begin{aligned} T &\rightarrow \bullet && \text{(type 1 rules: unpaired positions)} \\ T &\rightarrow (U) && \text{(type 2 rules: base-paired positions)} \\ T &\rightarrow UV && \text{(type 3 rules: branching)} \end{aligned} \tag{2.25}$$

2.2.10.2 The inside-outside algorithm

Using Bayes' rule, the probability of a secondary structure, $\mathcal{S}$, conditional on the data, $\mathcal{D}$, and phylogenetic parameters, $\boldsymbol{\theta}$, is given by:

$$p(\mathcal{S}|\mathcal{D}, \boldsymbol{\theta}) = \frac{p(\mathcal{D}|\mathcal{S}, \boldsymbol{\theta}) p_{\mathcal{G}}(\mathcal{S})}{p_{\mathcal{S}}(\mathcal{D}|\boldsymbol{\theta})} = \frac{p(\mathcal{D}|\mathcal{S}, \boldsymbol{\theta}) p_{\mathcal{G}}(\mathcal{S})}{\sum_{\mathcal{S}} p(\mathcal{D}|\mathcal{S}, \boldsymbol{\theta}) p_{\mathcal{G}}(\mathcal{S})} \tag{2.26}$$

The *structure-integrated likelihood* term in the denominator, $p_{\mathcal{S}}(\mathcal{D}|\boldsymbol{\theta}) = \sum_{\mathcal{S}} p(\mathcal{D}|\mathcal{S}, \boldsymbol{\theta}) p_{\mathcal{G}}(\mathcal{S})$, requires a sum over secondary structures, which as we will see in Section 2.2.21 grows exponentially large with the length of the alignment L. Fortunately, there exists a polynomial-time algorithm, the inside algorithm, for summing the probabilities of all derivations of an SCFG (all valid secondary structures in the case of RNA SCFG). By analogy to the forward algorithm for Hidden Markov Models (HMMs), the inside algorithm allows the structure-integrated likelihood, $p_{\mathcal{S}}(\mathcal{D}|\boldsymbol{\theta})$ (the analogue of the forward likelihood for HMMs), to be efficiently computed. Likewise, by analogy to the backward algorithm for HMMs, the SCFG outside algorithm together with the inside probabilities allows the posterior marginals of the hidden variables to be computed (in the case of an RNA SCFG: positional emission probabilities of base-pair and unpaired terminal symbols). A recursive formula for calculating the inside

probabilities is as follows:

$$I(U,q,r) = \begin{cases} p[U \to \bullet] p(\mathcal{D}_q | \dot{q}) & \text{if } q = r \\ \sum\limits_{U \to VW} \left[p[U \to VW] \sum\limits_{q \le k < r} I(V,q,k) I(W,k+1,r) \right] + & \\ p[U \to (V)] \sum\limits_{U \to (V)} I(V,q+1,r-1) p(\mathcal{D}_{q,r} | \widehat{q,r}) & \text{if } q < r \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

Note that the structure-integrated likelihood in the denominator of (2.26) is given by:

$$p_{\mathcal{S}}(\mathcal{D}|\boldsymbol{\theta}) = \sum_{\mathcal{S}} p(\mathcal{D}|\mathcal{S},\boldsymbol{\theta}) p_{\mathcal{G}}(\mathcal{S}) = I(S,1,L) \tag{2.28}$$

A recursive formula for calculating the outside probabilities is as follows:

$$O(U,q,r) = \begin{cases} 1 & \text{if } U = S, \\ & q = 1, r = L \\ \sum\limits_{V \to UW} \left[p[V \to UW] \sum\limits_{k > r} O(V,q,k) I(W,r+1,k) \right] + & \\ \sum\limits_{V \to WU} \left[p[V \to WU] \sum\limits_{k < q} O(V,k,r) I(W,k,q-1) \right] + & \\ \sum\limits_{V \to (U)} p[V \to (U)] O(V,q-1,r+1) p(\mathcal{D}_{q-1,r+1} | \widehat{q-1,r+1}) & \text{if } q < r \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

where $\mathcal{G}$ is a RNA secondary structure grammar.

The computational time complexity for both the inside and outside outside algorithms is $\mathcal{O}(L^3)$

2.2.10.3 Iterative implementation and parallelisation schemes

In practice, the inside and outside algorithms are typically implemented using an iterative approach (Lari and Young, 1991), rather than recursively as in (2.27) and

(2.29), respectively. Algorithm 2 and Algorithm 3 provide pseudocode for iterative implementations of the inside and outside algorithms for SCFGs in double emission normal form, respectively.

Figure 2.3 illustrates the calculation of the inside probability matrix, showing the order in which elements are computed and the data dependencies required to compute a particular element. Using these patterns, Sükösd *et al.* (2011) developed a strategy for CPU parallelism, whereby blocks of elements running diagonally along the inside matrix can be computed in parallel, as they do not have data dependencies. We extend this scheme to GPU architectures, whereby instead of blocks, each element along a diagonal is computed in parallel, with each element along a diagonal being independent. For large alignments ($L > 1000$) this implies thousands of computational threads executing the same set of instructions in parallel, but on different data (different elements of a particular diagonal), this is known as SIMD (Single Instruction Multiple Data) parallelism and is exactly the regime of parallelism for which GPU architectures are designed. As far as we know this is the first GPU implementation of the inside and outside algorithms.

2.2.11 Dinucleotide column likelihoods

Because the inside and outside algorithms consider every possible base-pairing they require a symmetric matrix $\mathbf{B}$ of *dinucleotide column likelihoods*. Each element $\mathbf{B}_{qr}$ of $\mathbf{B}$ corresponds to a dinucleotide column likelihood $p(\mathcal{D}_{q,r}|\widehat{q,r}, \mathcal{T}, \theta)$ for a pair of columns, q and r, in the alignment $\mathcal{D}$, which can be calculated using Felsenstein's algorithm. Since the diagonal of $\mathbf{B}$ is ignored and $\mathbf{B}_{qr} = \mathbf{B}_{rq}$ (i.e. $\mathbf{B}$ is symmetric), $\binom{L}{2}$ dinucleotide column likelihoods need to be calculated. Whilst, the number of computational steps is only $\mathcal{O}(L^2)$ in the alignment length L, compared to $\mathcal{O}(L^3)$ for the inside and outside algorithms, the amount of time per a computational step for computing the dinucleotide column likelihoods is substantially higher due the use of Felsenstein's algorithm. In Section 2.2.18 we discuss a partial likelihood caching strategy for ameliorating this.

Note that the inside and outside algorithms also require a vector, $\mathbf{S}$, of length of L single column likelihoods, where each element corresponds to $p(\mathcal{D}_q|\dot{q}, \mathcal{T}, \boldsymbol{\theta})$. However, this is trivial to compute compared to $\mathbf{B}$.

2.2.12 Sampling secondary structure configurations

The inside probabilities in Equation 2.27 can be used to sample secondary structure configurations according to their probability:

$$\tilde{\mathcal{S}} \sim p(\mathcal{S}|\mathcal{D}, \boldsymbol{\theta}) \tag{2.30}$$

Sampling terminal strings (secondary structures in our case) from an SCFG is analogous to sampling hidden state sequences using the forward-filtering backward-sampling algorithm for HMMs (Frühwirth-Schnatter, 1994). We provide an algorithmic description of the sampling procedure for an RNA SCFG here:

Assuming the double emission normal form, each element of $I(U, q, r)$ is composed of the probabilities of three event types: (i) branching, (ii) an unpaired position, and (iii) a base-pairing. Each of these events can be generated by one or more rules in the grammar.

For the KH99 grammar, sampling proceeds recursively starting with start symbol, S, and the probability of the entire alignment given by $I(S, 1, L)$. At each step a rule is sampled conditioned on the previous non-terminal symbol according to the rule, data, and inside probabilities. Each rule generates one of the three events types.

In the case of a branching event, a position is sampled along the current sub-alignment leading to the generation of two conditionally independent non-terminal symbols (L and S), L and S corresponding to left and right sub-alignments, respectively. The left sub-alignment ends at the sampled position and the right sub-alignment starts after the sampled position.

In the case of an unpaired position the terminal symbol, $\bullet$, is emitted at that position.

Algorithm 2 Iterative implementation of the inside algorithm for an RNA SCFG in double-emission normal form

for $U \to \bullet$ ***in*** *type1rules* **do**
  // consider all type 1 rules (rules generating unpaired positions)

  for $q = 1$ ***to*** L **do**
    I(U,q,q) = $p[U \to \bullet] \times p(\mathcal{D}_q|\dot{q})$ // likelihood of column q under the unpaired model
  end
end
for $b = 2$ ***to*** L **do**
  // consider subsequences of increasing length b

  for $r = 1$ ***to*** $L - b + 1$ **do**
    // consider all starting positions r for subsequences of length b

    for $U \to VW$ ***in*** *type3rules* **do**
      // consider all type 3 rules (rules generating a branching)

      for $h = r$ ***in*** $L + b - 2$ **do**
        // consider all breakpoint positions h generating a branching

        I(U,r,r+b-1) += $p[U \to VW]$
        $\times$ I(V,r,h) // probability of the left subsequence being generated by V
        $\times$ I(W,h+1,r+b-1) // probability of the right subsequence being generated by W
      end
    end
    for $U \to (V)$ ***in*** *type2rules* **do**
      // consider all type 2 rules (i.e. rules generating a base-pairing between positions r and $r + b - 1$)

      I(U,r,r+b-1) += $p[U \to (V)]$
      $\times\, p(\mathcal{D}_{r,r+b-1}|r, \widehat{r+b-1})$ // likelihood of dinucleotide column $(r, r+b-1)$ under the paired model
      $\times$ I(V,r+1,r+b-2) // probability of the subsequence between the base-paired positions being generated by V
    end
  end
end

Algorithm 3 Iterative implementation of the outside algorithm for an RNA SCFG in double-emission normal form

```
O(S,1,L) = 1.0
for c = 2 to L do
    b = L − c // consider subsequences of decreasing length b
    for r = 0 to L − b − 1 do
        // consider all starting positions r for subsequences of length b

        for U → VW in type3rules do
            // consider all type 3 rules (rules generating a branching)

            for h = r + b + 1 in L − 1 do
                O(V,r+1,r+b+1) += p[U → VW]
                × O(U,r+1,h+1) // outside probability leading to the generation of U
                × I(W,r+b+2,h+1) // inside probability of the right subsequence
            end
            for h = 0 in r − 1 do
                O(W,r+1,r+b+1) += p[U → VW]
                × O(U,h+1,r+b+1) // outside probability leading to the generation of
                U
                × I(V,h+1,r) // inside probability of the left subsequence
            end
        end
        for U → (V) in type2rules do
            // consider all type 2 rules (i.e. rules generating a base-pairing between
            positions r and r + b + 2)

            O(V,r+1,r+b+1) += p[V → (U)]
            × p(D_{r,r+b+2} | r, r + b + 2)
            × O(U,r,r+b+2)  // outside probability leading to the generation of U
        end
    end
end
```

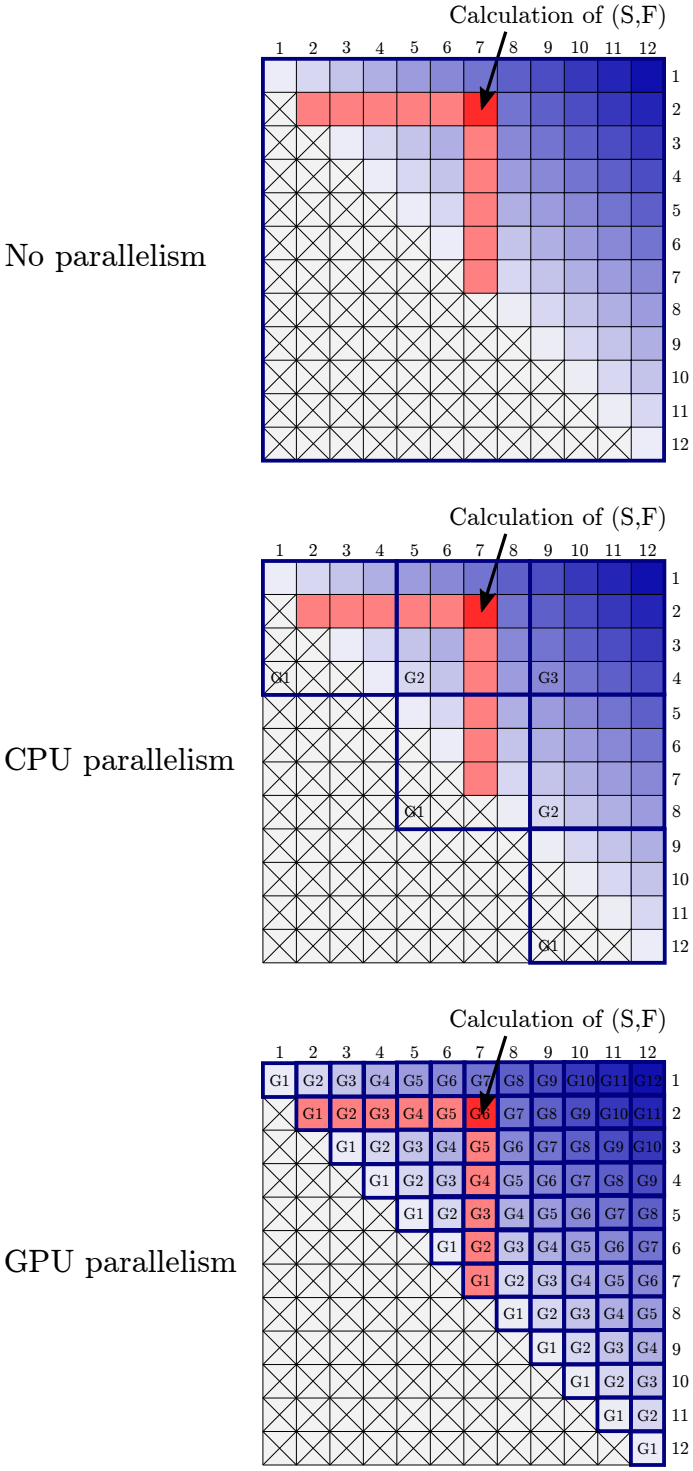


Figure 2.3: Illustration of an iterative implementation of the inside algorithm for an RNA SCFG under three different parallelisation scenarios (no parallelism, CPU parallelism, and GPU parallelism). The light to dark blue gradient moving across each matrix from the central diagonal to the top right-hand corner indicates the order in which each diagonal is computed. The light red elements indicate the data dependencies required to compute the single bright red entry of the inside matrix. The lower half of each matrix with each cell crossed out is not computed and can be ignored.

Finally, in the case of a base-pairing event, terminal symbols '(' and ')' are emitted, '(' corresponding to the first position of the current sub-alignment and ')' corresponding to the last position of the current sub-alignment (indicating that the start and end positions of the current sub-alignment are base-paired). Furthermore, a non-terminal symbol F corresponding to the remaining middle portion between the two base-paired positions is generated and is recursively sampled until no non-terminal symbols remain. With only a string of terminal symbols remaining the sampled secondary structure can be easily read off.

A code implementation, written in the `Julia` programming language is given in in Listing A.2. Note that the computational time complexity for sampling a single secondary structure, having precomputed the inside probabilities, is $\mathcal{O}(L^2)$.

2.2.13 Bayesian posterior inference

The posterior distribution of the continuous parameters, $\boldsymbol{\theta}$, conditional on the data $\mathcal{D}$ and a secondary structure $\mathcal{S}$, can be sampled using the Metropolis-Hastings algorithm and the relationship given by Bayes' formula:

$$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{S}) \propto p(\mathcal{D}|\mathcal{S},\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{2.31}$$

Where the likelihood term, $p(\mathcal{D}|\mathcal{S},\boldsymbol{\theta})$, is given by (2.23) and $p(\boldsymbol{\theta})$ is the prior.

We can also treat the secondary structure as unknown and assume a RNA SCFG prior, $p_{\mathcal{G}}(S)$, over secondary structures. Using the structure-integrated likelihood $p(\mathcal{D}|\boldsymbol{\theta})$:

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p_{\mathcal{S}}(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{2.32}$$

However, note that the structure-integrated likelihood term is computed every time a new set of parameters is proposed. As mentioned previously, this requires computing a matrix $\mathbf{B}$ of dinucleotide column likelihoods (requiring $\mathcal{O}(L^2)$ computational steps) and calculating the final structure-integrated likelihood term using the inside

algorithm (requiring $\mathcal{O}(L^3)$ computational steps). Therefore gathering enough samples to ensure an adequate sample size will be relatively slow. However, given that we can sample the conditional distribution, $p(\mathcal{S}|\mathcal{D}, \boldsymbol{\theta})$, using the sampling procedure outlined in Section 2.2.12 this leads to a potentially more efficient Metropolis-within-Gibbs approach (Chib and Greenberg, 1995). This approach works by alternatively sampling from the full conditional distributions:

$$\mathcal{S}^{(k)} \sim p(\mathcal{S}|\mathcal{D}, \boldsymbol{\theta}^{(k)}) \tag{2.33}$$

using the sampling procedure outlined in Section 2.2.12 and

$$\boldsymbol{\theta}^{(k+1)} \sim p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{S}^{(k)}) \tag{2.34}$$

using the Metropolis-Hastings algorithm. Whilst the first step still requires computing a matrix $\mathbf{B}$ of dinucleotide likelihoods and running the inside algorithm, the second step requires $\mathcal{O}(L)$ calculations and can be repeated for multiple iterations following the first sampling step. In our implementation we repeat this second step 50 times following the first step. Together, these give a Markov Chain Monte Carlo algorithm whose stationary distribution, $p(\mathcal{S}, \boldsymbol{\theta}|\mathcal{D})$, and associated marginals, $p(\mathcal{S}|\mathcal{D})$ and $p(\boldsymbol{\theta}|\mathcal{D})$, are the distributions of interest.

The reduction in the average computational time per sample allows us to draw many more samples. The efficiency of this approach relative to using the structure-integrated likelihood in (2.32), relies on the dependencies between $\mathcal{S}$ and $\boldsymbol{\theta}$ being relatively weak. Empirically, the Metropolis-within-Gibbs algorithm give the same posterior distributions, but significantly improved convergence (measured using Estimated Sample Size of continuous parameters across different datasets) per unit of computational time.

2.2.14 Maximum likelihood inference

The COBYLA optimization algorithm (Powell, 1994) in the NLOpt library (Johnson, 2014) was used to find the maximum likelihood (ML) parameters via the structure-integrated likelihood (2.26). Note that when doing so the priors over the continuous parameters were either ignored and estimated using ML, or the priors were used and the parameters were fully marginalised (see Section 2.2.7)

2.2.15 Posterior base-pairing and unpairing probabilities

The posterior base-pairing probability of a pair of positions q and r can be obtained by calculating the expected positional emission probabilities of type 2 rules (base-pairing rules) using the inside and outside probabilities:

$$p(q,r) = \frac{\sum\limits_{U\to(V)} O(U,q,r)p[U\to(V)]p(q,r)I(V,q+1,r-1)}{I(S,1,L)} \tag{2.35}$$

Likewise the unpairing probability of a position i can be obtained using the expected positional emission probabilities of type 1 rules (unpaired rules):

$$p(q) = \frac{\sum\limits_{U\to\bullet} O(U,q,q)p[U\to\bullet]p(q)}{I(S,1,L)} \tag{2.36}$$

2.2.16 Consensus structure prediction

The maximum expected accuracy (MEA) algorithm is an algorithm for finding the secondary structure with the highest sum of base-pairing and unpaired probabilities (Lu *et al.*, 2009). The MEA value and secondary structure is obtained by computing the dynamic programming matrix:

$$E_{q,r} = \max \begin{cases} p(q) + E_{q+1,r} \\ p(q,r) + E_{q+1,r-1} \\ \max\limits_{q+1<k<r} \alpha p(q,k) + E_{q+1.k-1} + E_{k+1,r} \end{cases} \tag{2.37}$$

Where $E_{q,r} \coloneqq 0$ for $q \geq r$, the MEA value is given by $E_{1,L}$, and the MEA structure is obtained via backtracking through the matrix, starting with $E_{1,L}$. The parameter, $\alpha = 2$, controls the trade-off between recall and precision (see Section 2.2.20) in predicting base-pairs, where higher α's encourage a greater number of base-pairings to be predicted over unpaired positions.

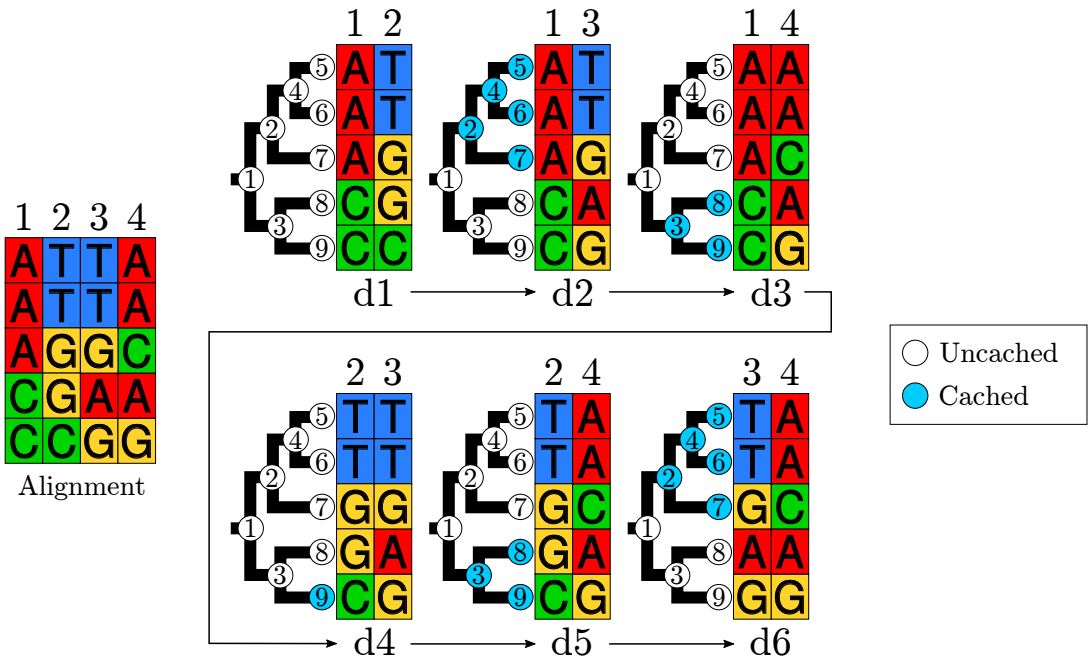
2.2.17 Proposal variances

The proposal variances, $\boldsymbol{\Sigma}$, used to update the continuous parameters of the model during Metropolis-Hastings inference, $\boldsymbol{\theta}$, were determined by (i) obtaining a maximum likelihood estimate $\hat{\boldsymbol{\theta}}$, (ii) initialising an MCMC chain using $\hat{\boldsymbol{\theta}}$ and running the chain for 200 iterations using a covariance matrix with a small variance, such that majority of proposals will be accepted, (iii) computing a covariance matrix from first the 200 samples, $\boldsymbol{\Sigma}^i$, (iv) using $\boldsymbol{\Sigma}^i$ to propose parameters in iteration $i+1$ and computing a new covariance matrix $\boldsymbol{\Sigma}^{i+1}$ from all the samples up to this point, and (v) continuing this procedure for 7500 iterations and freezing the covariance matrix after the final iteration to obtain a final covariance matrix, $\boldsymbol{\Sigma}$, used throughout to make proposals. This approach is based on the Adaptive Metropolis procedure described in Roberts and Rosenthal (2009).

2.2.18 Partial likelihood caching

For an alignment $\mathcal{D}$ of length L it is necessary to compute $\binom{L}{2}$ dinucleotide phylogenetic likelihoods using Felsenstein's algorithm. Despite having a lower order of complexity ($\mathcal{O}(L^2)$ computational steps) than the inside algorithm ($\mathcal{O}(L^3)$ computational steps), in practice this step often requires more computational time, thereby creating a computational bottleneck.

To ameliorate this bottleneck, we use the partial site caching strategy of Pond and Muse (2004) to reduce the number of likelihood calculations required. Figure 2.4 provides a graphical illustration of how this strategy works. The key idea is that a subtree identifies a subset of characters in a column (in our case: pairs of characters



Six (4 choose 2) combinations of dinucleotide sites

Computations required:

Without caching = 9 × 6 = 54

With caching = 9 + 4 + 6 + 8 + 6 + 4 = 37

Without caching (internal nodes) = 4 × 6 = 24

With caching (internal nodes) = 4 + 2 + 3 + 4 + 3 + 2 = 18

Figure 2.4: Dinucleotide likelihood calculations for an alignment of length 4, where all 6 (4 choose 2) dinucleotide columns have been enumerated, showing at each step a dinucleotide column and the partial likelihoods which have previously been computed and cached.

in a dinucleotide column), thus if two different columns identify the same subset of characters corresponding to the same subtree, the partial likelihoods of those subtrees will be the same (assuming the same underlying model for both columns). Thus recomputing the partial likelihood of the second column can be avoided by looking up the cached value that has been previously computed.

2.2.19 Datasets

Two classes of datasets were analysed. The first class of datasets consisted of non-coding RNA alignments obtained from the RFAM database (Burge *et al.*, 2012). Each RFAM dataset has an associated consensus secondary structure based on experimental RNA structure determination. These consensus secondary structures were only

used for secondary structure prediction benchmarking purposes and were not conditioned on during inference. RFAM datasets are denoted with 'RF' prefix in their name. The second class of datasets analysed, consisted of the complete genomes of single-stranded RNA and single-stranded DNA viruses obtained from the NCBI nucleotide database (Acland *et al.*, 2014) and aligned using MUSCLE (Edgar, 2004). Typically these datasets did not have associated consensus secondary structures.

2.2.19.1 Representative subset selection

Many of the viruses selected for analysis had a large number of complete genomic sequences available ($200-2000$ sequences) in the NCBI nucleotide database. Because the inference algorithms scale linearly in the number of sequences, we chose to reduce each full dataset F to a smaller representative subset S^* of N sequences, where N is typically 100-200 sequences. A genetic algorithm was used to find the optimal (or near-optimal) subset S^* containing N sequences such that following criteria was maximised:

$$S^* = \underset{S \subseteq F, |S|=N}{\operatorname{argmax}} \sigma(S) \tag{2.38}$$

where

$$\sigma(S) = \sum_{s \in S} \min_{s' \in S \setminus s} d(s, s') \tag{2.39}$$

where $d(s, s')$ is the hamming distance between aligned sequences s and s'.

In other words: S^* was chosen amongst possible subsets in F containing N sequences such that the sum of the distances between each sequence and its closest sequence in the subset was maximised. The result is a subset S^* that attempts to maximise the amount of diversity (*i.e.* information) in each dataset, whilst reducing the number of sequences used during inference, to help reduce the computational burden.

2.2.19.2 Linearisation of circular DNA viruses

Most known DNA viruses have circular genomes - genomes that form a closed loop without 5' and 3' ends. This presents a problem when modelling their secondary structures, as the underlying SCFG and inside-outside algorithm assume linear sequences.

To address this, we linearised every circular DNA virus alignment by choosing an appropriate point of linearisation. We chose to linearise the DNA virus alignments at their aligned origin of replication (*ori*) sites. All of the DNA viruses in our analysis contain known ori's that form highly stable secondary structures (Muhire *et al.*, 2014). We linearise each alignment within the loop of region of these structures, thereby ensuring that long-range base-pairings are maintained.

2.2.20 Measures of predictive accuracy

To measure the predictive accuracy of a secondary structure prediction it is necessary to have a *reference* secondary structure (typically determined from experiment) to measure the prediction against. In Figure 2.5 we give an example of a predicted structure and a reference structure and use them to compare different measures of predictive accuracy in this section.

The first few measures described here require counts of *true positives* (TP, the number of correctly predicted base-pairs), *false negatives* (FN, the number of base-pairs in the reference structure not in the predicted structure), and *false positives* (FP, the number of base-pairs predicted not in the reference structure). The number of *true negatives* (TN) is also required in some cases, which is defined as $\text{TN} = \frac{L(L-1)}{2} - \text{TP} - \text{FN} - \text{FP}$. Finally, the number of compatible base-pairs (ξ, defined by condition (i) in Section 2.2.8), defined as the number of FP base-pairs compatible with all base-pairs in the reference structure, is also sometimes used to reduce the effect false positives when measuring predictive accuracy.

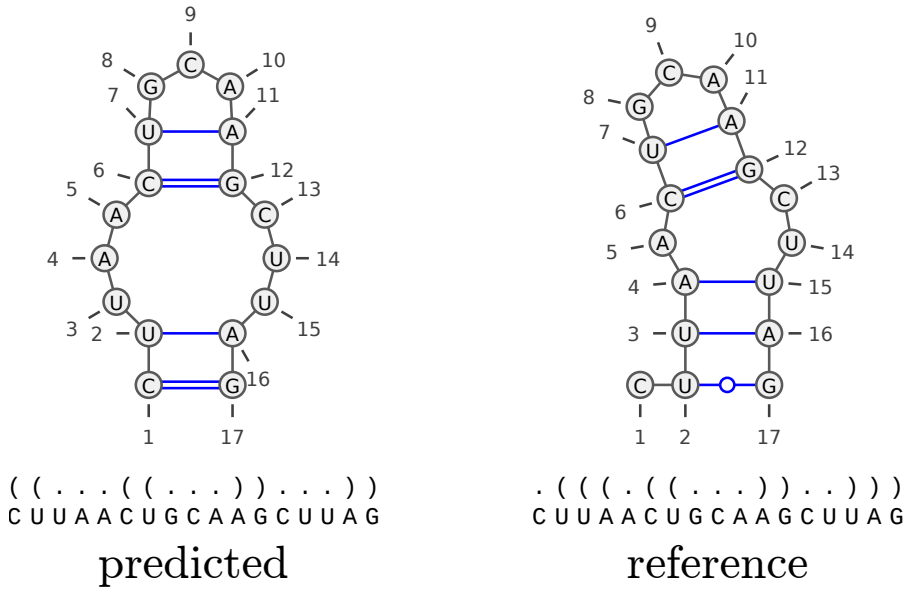


Figure 2.5: Two example secondary structures visualised using VARNA (Darty *et al.*, 2009) with the corresponding dot bracket structures indicated below. The first is a 'predicted' structure and the second is a 'reference' structure corresponding to the same sequence. True positive base-pairs (TP=2), false negative base-pairs (FN=3), false positive base-pairs (FP=2), true negative base-pairs (TN $= \frac{L(L-1)}{2} - \text{TP} - \text{FN} - \text{FP} = 136 - 2 - 3 - 2 = 129$), and compatible bas-pairs ($\xi = 2$).

2.2.20.1 Precision, recall, and F-score

Two simple measures of predictive accuracy are precision and recall (Gardner and Giegerich, 2004), defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.40}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.41}$$

Precision can be interpret as the number of correctly predicted base-pairings (TP) compared to the total number of predicted base-pairings (TP + FP), whereas recall can be interpreted as the number of correctly predicted base-pairings (TP) compared to the total number of base-pairings in the reference structure (TP + FN).

Note that recall can be increased by encouraging a particular prediction method to predict more base-pairings; however, this will likely result in an increase in the number false positives (FP) and a corresponding decrease in precision. Therefore any

particular method should aim to score high on both metrics, for this reason the two metrics are often "averaged" by taking their harmonic mean (termed the F1-score) as follows:

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.42}$$

2.2.20.2 Matthews correlation coefficient

Another commonly used measure of predictive accuracy is Matthews correlation coefficient (MCC; Baldi *et al.* (2000)). Like the F-score, MCC combines both precision and recall in a single measure. It is defined as follows:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - (\text{FP} - \xi) \times \text{FN}}{\sqrt{(\text{TP} + \text{FP} - \xi)(\text{TP} + \text{FN})(\text{TN} + \text{FP} - \xi)(\text{TN} + \text{FN})}} \tag{2.43}$$

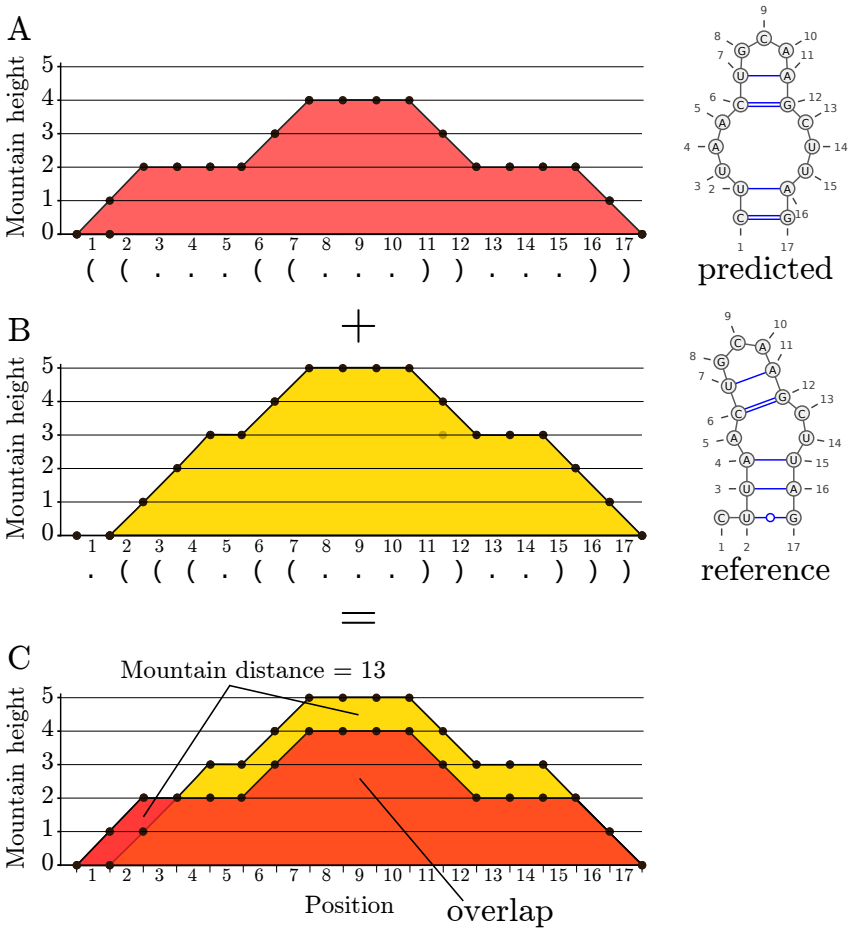
MCC ranges in value from -1 (completely wrong) to 1 (perfect prediction). However, in the case of secondary structure, MCC typically ranges from 0 to 1, because TN= 0 rarely occurs in practice (Gardner and Giegerich, 2004).

2.2.20.3 Mountain metric

The mountain metric is a measure of secondary structure distance based on the mountain representation of secondary structures, which follows naturally from the dot-bracket representation of secondary structures.

Using the mountain representations for secondary structures it is simple to graphically derive the mountain metric distance (see Figure 2.6 for an explanation). For a complete mathematical exposition see Moulton *et al.* (2000).

The mountain distance provides a nuanced measure of structural similarity compared to the simple base-pair metrics described above. For example, it better captures structural features compared to base-pair metrics that heavily penalize base-pairings that are even slightly offset relative to one another. For example: shifting the base-pairings, as in Figure 2.6, would be heavily penalized by simple base-pair metrics (resulting in an F1 score of 0.44 or 44% in this example), whereas the mountain sim-



Mountain diameter = 72

Normalised mountain distance = 13 / 72 = 0.181

Mountain similarity = 1 - 0.181 = 0.819 or 81.9%

Figure 2.6: Mountain plots of two secondary structures. The mountain height at a particular sequence position corresponds to the number of base-pairs that enclose that position. For example, immediately after the third position in both structures (plots A and B) the mountain height is 2, because both positions are enclosed by two base-pairs. The non-overlapping areas in red and yellow (plot C) provide a graphical depiction of the mountain distance between the two secondary structures. The mountain distance can be normalised by the mountain diameter. The mountain diameter is the maximum possible mountain distance between two structures of a given length.

ilarity metric better captures similarities in shape (resulting in a mountain similarity measure of 0.819 or 81.9%).

The mountain metric together with a set of secondary structures defines a *metric space*. A metric space consists of a set and a distance function, where the distance function has certain properties. One property is symmetry, that is $d(x, y) = d(y, x)$ - the distance remains the same regardless of the ordering of the two elements being compared. Note that symmetry does not hold for measures such as precision, recall, or MCC, and hence these measures can not be used as distance functions as part of metric spaces.

2.2.21 Combinatorics of RNA secondary structure

A useful theoretical result is the number of valid RNA secondary structures of a given length (Anderson, 2014). A recursive formula for counting the number of non-pseudoknotted secondary structures of length L is provided here. In particular, it counts all secondary structures that can be written in dot-bracket notation. An additional constraint is included such that there are at least g unpaired nucleotides between every base-pair. This constraint reflects the physical reality that hairpin loops with two or fewer unpaired nucleotides are energetically unfavourable to that extent that they are never observed (Hofacker *et al.*, 1998). The formula is as follows:

$$S(L) = S(L-1) + \sum_{k=1}^{L-g-1} S(k-1)S(L-k-1) \tag{2.44}$$

where $S(0) \coloneqq 1$ and $S(L) = 1$ for $1 \leq L \leq g$.

A graphical description of (2.44) is given in Figure 2.7. An asymptotic formula is available for $g = 1$, as follows: $S(L) \approx 1.104 \times L^{\frac{3}{2}} \times 2.618^L$ for large L (Nebel and Schulz, 2014).

Table 2.1 enumerates the counts of secondary structures for $g = 0, 1, 2, 3$ and for $L = 1, 2, \ldots, 30$. Figure 2.8 enumerates all sixteen secondary structure of length $L = 8$ with at least three unpaired nucleotides between every base-pair ($g = 3$).

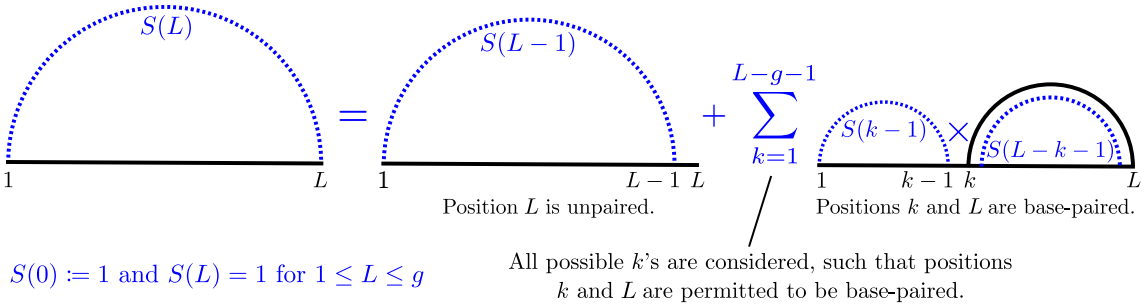


Figure 2.7: A graphical depiction of the recursive formula for counting non-pseudoknotted secondary structures of length L with at least g unpaired nucleotides between every base-pair.

Appendix A.3 provides `Julia` code for efficiently computing the counts and log-counts. Computation of log-counts uses fixed width float 64 bit data types, providing a fast and accurate approximation of the counts in log-space. Computation of the counts (not in log-space), whilst exact, requires 'big integer' data types which grow in memory and computation with the number of bits and bit operations required to store and calculate the counts, respectively.

2.2.22 Information Entropy

Information entropy is a measure of the spread of a probability distribution. It can be useful in characterising distributions over RNA secondary structures, where the large number of possible secondary structures often hinders summarization. Information entropy is defined as follows:

$$H(X) \coloneqq -\sum_{x} p(x) \log_2 p(x) \tag{2.45}$$

where X is a discrete random variable with possible values $\{x_1, \ldots, x_n\}$.

Information entropy has well-defined minimum and maximum values, where 0 is the minimum and where the maximum is determined by the number of possible outcomes N and is equal to $\log_2 N$. Low entropy indicates that probability mass is concentrated amongst a small number of outcomes, whereas high entropy indicates that probability mass is spread more equally amongst outcomes. Figure 2.8 provides

Table 2.1: Counts of secondary structure

L (length)	$g = 0$ oeis.org/A086246	$g = 1$ oeis.org/A004148	$g = 2$ oeis.org/A004149	$g = 3$ oeis.org/A023421
1	1	1	1	1
2	2	1	1	1
3	4	2	1	1
4	9	4	2	1
5	21	8	4	2
6	51	17	8	4
7	127	37	16	8
8	323	82	33	16
9	835	185	69	32
10	2188	423	146	65
11	5798	978	312	133
12	15511	2283	673	274
13	41835	5373	1463	568
14	113634	12735	3202	1184
15	310572	30372	7050	2481
16	853467	72832	15605	5223
17	2356779	175502	34705	11042
18	6536382	424748	77511	23434
19	18199284	1032004	173779	49908
20	50852019	2516347	390966	106633
21	142547559	6155441	882376	228505
22	400763223	15101701	1997211	490999
23	1129760415	37150472	4532593	1057683
24	3192727797	91618049	10311720	2283701
25	9043402501	226460893	23512376	4941502
26	25669818476	560954047	53724350	10713941
27	73007772802	1392251012	122995968	23272929
28	208023278209	3461824644	282096693	50642017
29	593742784829	8622571758	648097855	110377543
30	1697385471211	21511212261	1491322824	240944076

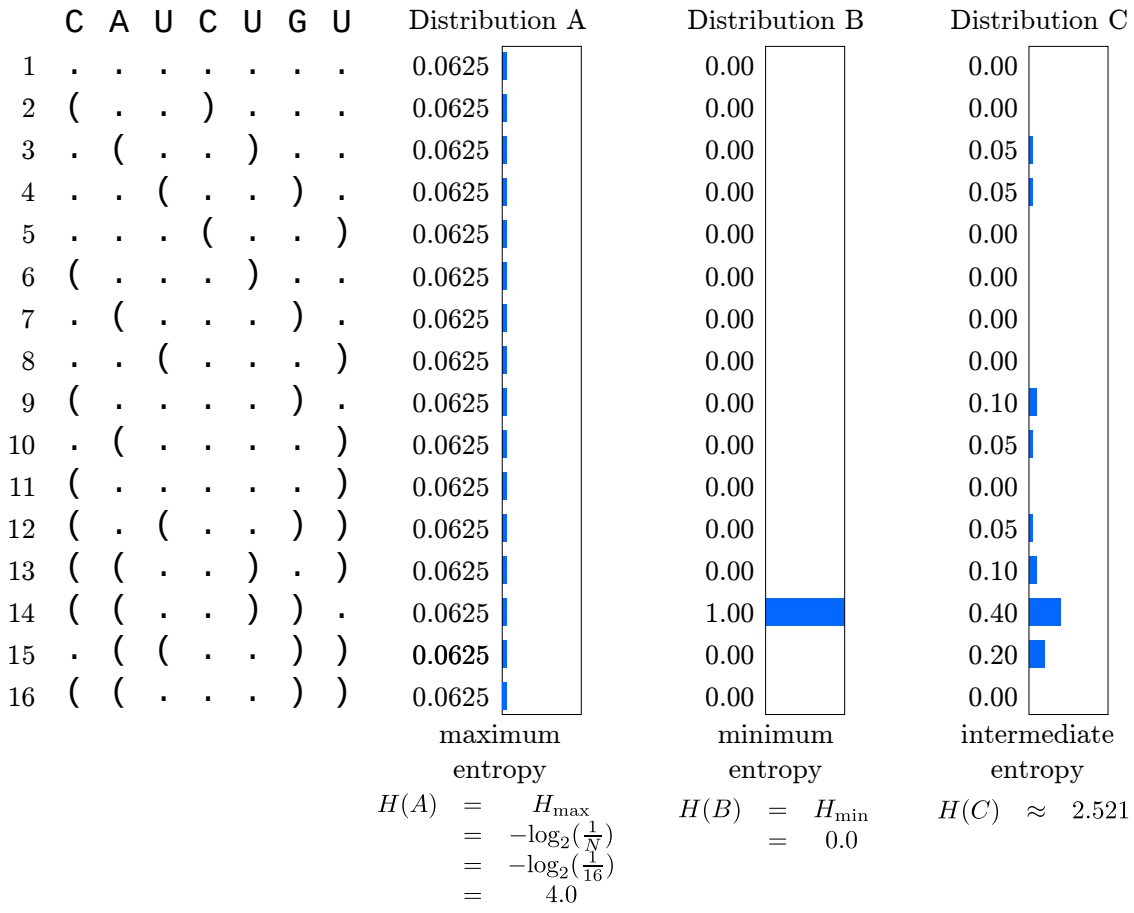


Figure 2.8: Three hypothetical distributions (A, B, and C) over $N = 16$ possible RNA secondary structures of length $L = 8$, illustrating three representative scenarios for information entropy (maximum, minimum, and intermediate entropy, respectively).

an illustrative example.

Due to the large number of secondary structures it is computationally intractable to compute the information entropy of a RNA secondary structure distribution using (2.45). Sükösd *et al.* (2013) show how to efficiently compute the information entropy for RNA SCFGs (given in double emission normal form) using the pairing, unpaired, and inside-outside probabilities calculated using Algorithm 2 and Algorithm 3. Note that is also possible to approximate the information entropy of a distribution, $H(X)$, using a set of independent samples $\tilde{X}$:

$$H(X) = \mathbb{E}[-\log_2 p(x)] \approx \frac{1}{|\tilde{X}|} \sum_{\tilde{x} \in \tilde{X}} \log_2(p(\tilde{x})) \tag{2.46}$$

We used this approach, due to the ease of implementation.

2.2.23 Column permutations

To assess the degree to which secondary structure dependencies present in real datasets influence model fit, each alignment was taken and its columns randomly permuted. Two such permuted datasets (p_1 and p_2) were generated for each real dataset. ML estimation using the structure-integrated likelihood was used to fit the parameters of each permuted dataset under the unconstrained model and the secondary structure information entropy calculated.

2.2.24 Spearman's rank correlation coefficient and confidence intervals

In some instances we use Spearman's rank correlation (Spearman, 1904) to test whether there is a monotonic relationship between two variables (an increasing or decreasing relationship, where the relationship is not necessarily linear). When doing so, we sometimes include 95% confidence intervals for the measured rank correlation coefficients to compare the results of two or more tests. The 95% confidence intervals for the Spearman's rank correlation coefficient ρ with n observations is given by (Ruscio, 2008):

$$\mathrm{CI}_{2.5} = \tanh(\operatorname{arctanh}(\rho) - 1.96/\sqrt{n-3}))$$

and

$$\mathrm{CI}_{97.5} = \tanh(\operatorname{arctanh}(\rho) + 1.96/\sqrt{n-3}))$$

2.2.25 Substructure ranking

For large sequences, rather than consider the entire secondary structure of a sequence, it is useful to consider smaller regions termed *substructures*. There are two primary reasons for considering substructures: smaller regions are more easily conceptualised, and functional components of secondary structures tend to correspond to sub-regions of a secondary structure (i.e. substructures). An example of a highly functional

substructure is the 250 nucleotide-long Rev Response Element (RRE) of HIV (Heaphy *et al.*, 1990; Daugherty *et al.*, 2010).

To rank substructures, the complete secondary structure of a nucleic acid sequence was subdivided into a number of smaller substructures 25-400 nucleotides in length. The substructures were then ranked by the numeric values within them. The ranking was performed by comparing two lists of numeric values: one corresponding to values within the substructure, and one corresponding to all values outside of the substructure (all values in the complete structure, but not in the substructure).

The two lists were compared using a Wilcoxon-rank sum test which assesses statistically whether values within the substructure are significantly higher (or lower) than values outside of it. The Wilcoxon-rank test statistic is used to compute a z-score for each substructure indicating the degree of difference. For example, a substructure with a large positive z-score would suggest that the distribution of numeric values within the substructure are significantly higher than the distribution of numeric values outside of the substructure. By using a Wilcoxon-rank sum test, this method attempts to account for the number of data points associated with each substructure. Substructures are subsequently ranked by sorting by their z-scores.

2.2.26 Visualisation of secondary structures

A feature of PINNACLE is the automatic generation of interactive secondary structure visualisations, overlaid with various posterior inferences or other relevant data. Visualisation of secondary structures is algorithmically challenging, it involves taking a raw secondary structure data file, which typically contains a list of paired and unpaired nucleotide positions, and producing an aesthetically pleasing 2-dimensional graphical layout consisting of the stems and loops that represent the secondary structure. Laying out the secondary structure such that stems and loops in different parts of the structure do not overlap with one another is the main challenge in visualising secondary structure. Most visualisation tools such as VARNA (Darty *et al.*, 2009) have little trouble in laying out structures that are smaller than 300 nucleotides and

can do so with minimal overlap; however, overlapping structural features become increasingly problematic for larger secondary structures.

To avoid the challenges involved with layout of large secondary structures, we subdivide large secondary structures into a number of smaller substructures as described in the previous section (Section 2.2.25), we then use VARNA (Darty *et al.*, 2009) to compute a layout, which is essentially a list of 2-dimensional graphical coordinates corresponding to each nucleotide. These two-dimensional coordinates are then used to visualise each substructure side-by-side, thus avoiding the problem of overlap, whilst enabling the entire secondary structure to be visualised in a single image.

A secondary structure is displayed as a series of circular nodes corresponding to nucleotides with base-paired nucleotides adjacent to one another. The circular nodes representing nucleotides can be assigned colours representing data values specific to that nucleotide site, for example: estimated substitution rates, or experimentally determined chemical reactivities. Depicted within each circular nucleotide is a sequence logo summarising the nucleotide composition at the corresponding alignment position. Data related to pairs of nucleotides, such as posterior base-pairing probabilities or degrees of coevolution are visualised using coloured links (thick lines between pairs of nucleotides).

The Scalable Vector Graphics (SVG) format was used to store the visualisations. SVG's interactivity features were also used to provide additional information in the form of tool tips when mousing over nucleotides or links. The visualisation strategies outlined above were motivated by Golden and Martin (2012). The visualisation of relevant information, such as posterior inferences, in the context of secondary structure is intended to help researchers in obtaining biological insights (see Figure 2.30 in Results and Discussion).

2.3 Results and Discussion

2.3.1 Column permutation benchmarks

To assess the degree to which secondary structure dependencies present in real datasets influence model fit, likelihoods and structure information entropies were computed for real datasets and corresponding permuted datasets using their ML parameter estimates (Section 2.2.23 in Methods). The structure-integrated likelihoods (which also represent the model evidence) for the permuted datasets were expected to be lower than that of the real datasets. Whilst, the structure information entropies were expected to be higher for the permuted datasets than the real datasets. This is due to the permuted patterns of coevolution being unlikely to coincide with stable secondary structure configurations compared to the real datasets, thereby spreading the probability mass over a larger number of secondary structure configurations.

The results are presented in Table 2.2. The maximum likelihood estimates of the structure-integrated likelihoods were indeed lower for the permuted datasets in every instance (see ΔMLL for p_1 and p_2). This serves partially to validate our model and is consistent with the presence of real secondary structure dependencies in the original datasets. As expected, the structure information entropies are generally higher for the permuted datasets, with the exception of RF00003, which has marginally lower structure information entropies for both of the permuted datasets. This result is surprising, RF00003 corresponds to the U1 spliceosomal RNA, a component of a spliceosome (Burge *et al.*, 2012), and a highly stable structure. Therefore it was expected that the real dataset would have a substantially lower structure information entropy than the permuted datasets. One explanation may be that the patterns of nucleotides within the real dataset are only weakly informative of the underlying secondary structure, given that the structural model is only informed by evolutionary information and not thermodynamics.

Table 2.2: Maximum log-likelihood (MLL), structure information entropy (H), maximum information entropy ($H_{\max}$), and normalised structure information entropy ($H/H_{\max}$) values for nine different datasets and for two random permutations (p_1 and p_2) of those datasets.

Dataset	MLL	ΔMLL unpaired	Length	H	$H_{\max}$	$H/H_{\max}$	ΔMLL p_1	ΔH p_1	$\Delta H/H_{\max}$ p_1	ΔMLL p_2	ΔH p_2	$\Delta H/H_{\max}$ p_2
WDV	-10731.6	-88.8	2755	1187.7	3806.9	0.312	-50.8	+276.6	+0.073	-56.4	+365.3	+0.096
MSV	-15341.5	-270.6	2755	1484.9	3806.9	0.390	-245.9	+99.3	+0.026	-234.8	+10.7	+0.003
RF00001	-27897.7	-3605.0	230	1.4	307.6	0.005	-2509.6	+6.4	+0.021	-2539.9	+6.2	+0.020
RF00002	-4399.1	-162.5	207	29.2	275.9	0.106	-115.4	+2.7	+0.010	-100.6	+13.1	+0.047
RF00003	-4653.0	-322.5	203	14.7	270.4	0.054	-255.8	-0.3	-0.001	-268.5	-3.6	-0.013
RF00004	-16716.6	-1733.6	278	7.9	373.8	0.021	-1085.6	+7.6	+0.020	-1301.3	+18.0	+0.048
Hepatitis A	-63755.3	-434.3	7572	3327.1	10490.7	0.317	-221.0	+720.0	+0.069	-200.1	+586.8	+0.056
Rhinovirus A	-217136.2	-6905.2	7308	1524.4	10124.4	0.151	-1118.7	+135.1	+0.013	-1120.7	+127.6	+0.013
HPV1	-64966.3	-438.5	7668	3023.8	10624.0	0.285	-281.6	+541.7	+0.051	-306.4	+810.8	+0.076

2.3.2 Partial site caching yields computational reductions

Table 2.3 lists datasets and the number of dinucleotide likelihood calculations required without partial site caching and with partial site caching for each. The factor reduction in the number of computations when using partial site caching for all the datasets tested was substantial, with at least a 5× reduction in all cases and a ∼15× reduction for the largest datasets.

The effectiveness of this strategy depends on the number of unique subtrees. Datasets with lower sequence diversity, will tend to have fewer unique subtrees, and hence require fewer likelihood calculations. The large number of dinucleotide columns (many of which will share identical first or second columns) implies that the fraction of unique subtrees will be relatively small compared to the total number, thus providing a substantial speed-up by avoiding a large number of redundant likelihood calculations. In the next section we show how further substantial speed-ups are obtained using a GPU implementation of Felsenstein's algorithm for computing dinucleotide likelihoods.

2.3.3 CPU and GPU timing benchmarks

The two computational bottlenecks in performing both maximum likelihood and Bayesian inference under our model are computing dinucleotide likelihood matrices (computed using an iterative version of Felsenstein's algorithm) and computing inside probability matrices (using an iterative version of the inside algorithm), as they are required repeatedly during inference. Highly optimized CPU implementations written in Julia were created for both of these steps; however, this was still relatively slow, and so GPU implementations written in CUDA were also implemented for both.

Figure 2.9 shows CPU and GPU timings for the dinucleotide likelihood matrices as a function of the number of unique dinucleotide patterns in a particular dataset. The number of computational steps is expected to grow with the number of unique dinucleotide patterns and hence this was chosen as a predictor of the computational time required. A roughly 50× speed-up was achieved with the GPU implementation

Table 2.3: Internal node computations without and with partial site caching for all pairwise combinations of columns for different datasets. Note that the datasets are ordered from highest to lowest number of likelihood calculations (without caching).

Dataset	Number of sequences	Alignment length	Computations without caching	Computations with caching	Factor reduction
FMDV	100	8349	3,450,023,874	111,747,561	30.9
Tobamovirus	141	6849	3,283,136,640	75,018,763	43.8
RF02543	88	8395	3,065,341,905	134,085,987	22.9
Bocavirus	199	5577	3,047,540,496	36,391,033	83.7
Rhinovirus A	108	7308	2,856,876,246	98,324,706	29.1
HPV 1	100	7668	2,616,179,742	64,392,203	40.6
Hepatitis A	82	7572	2,321,768,286	28,078,937	82.7
RF02541	102	5241	1,386,873,420	66,190,379	21.0
TYLCV	216	2925	915,138,900	22,911,540	39.9
RF02540	91	4236	807,275,700	52,978,933	15.2
BCTV	107	3215	537,316,520	13,802,367	38.9
RF00010	458	996	226,448,070	9,712,256	23.3
MSV	55	2755	204,856,290	4,952,972	41.4
WDV	54	2755	201,062,655	1,018,252	197.5
RF02542	46	1752	69,024,420	4,864,698	14.2
RF00030	66	1213	47,780,070	7,367,028	6.5
RF00174	430	441	41,621,580	3,488,548	11.9
RF02001	406	403	32,806,215	1,894,116	17.3
RF00011	114	604	20,577,978	2,531,712	8.1
RF00001	712	230	18,724,185	1,302,964	14.4
RF00209	92	569	14,705,236	719,672	20.4
RF01846	66	624	12,634,440	1,279,334	9.9
RF00004	208	278	7,777,606	1,149,997	6.8
RF00380	157	278	6,006,468	631,017	9.5
RF00379	106	335	5,874,225	815,904	7.2
RF00012	87	316	4,280,220	786,257	5.4
RF00026	188	215	3,726,810	409,518	9.1
RF00100	45	402	3,546,444	670,509	5.3
RF01854	92	315	3,362,940	602,410	5.6
RF00020	180	192	3,190,464	500,735	6.4
RF00003	100	203	1,783,761	288,081	6.2
RF00002	61	207	1,279,260	164,088	7.8

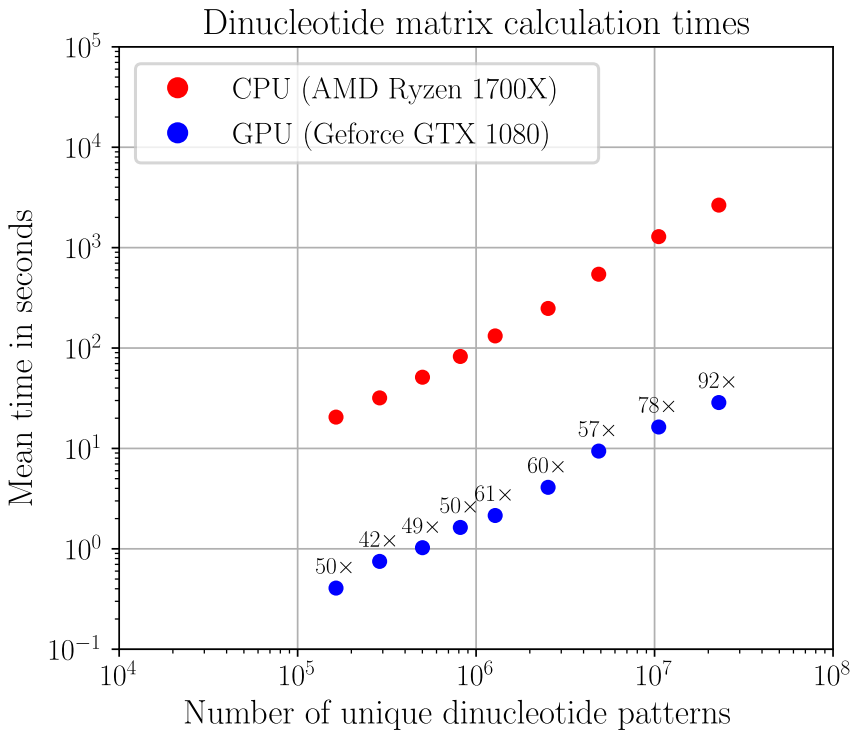


Figure 2.9: Dinucleotide matrix calculation timings in seconds ($\log_{10}$ axis) as a function of the number unique dinucleotide patterns ($\log_{10}$ axis), with the fold speed-up of the GPU version over the CPU version indicated above the GPU timings.

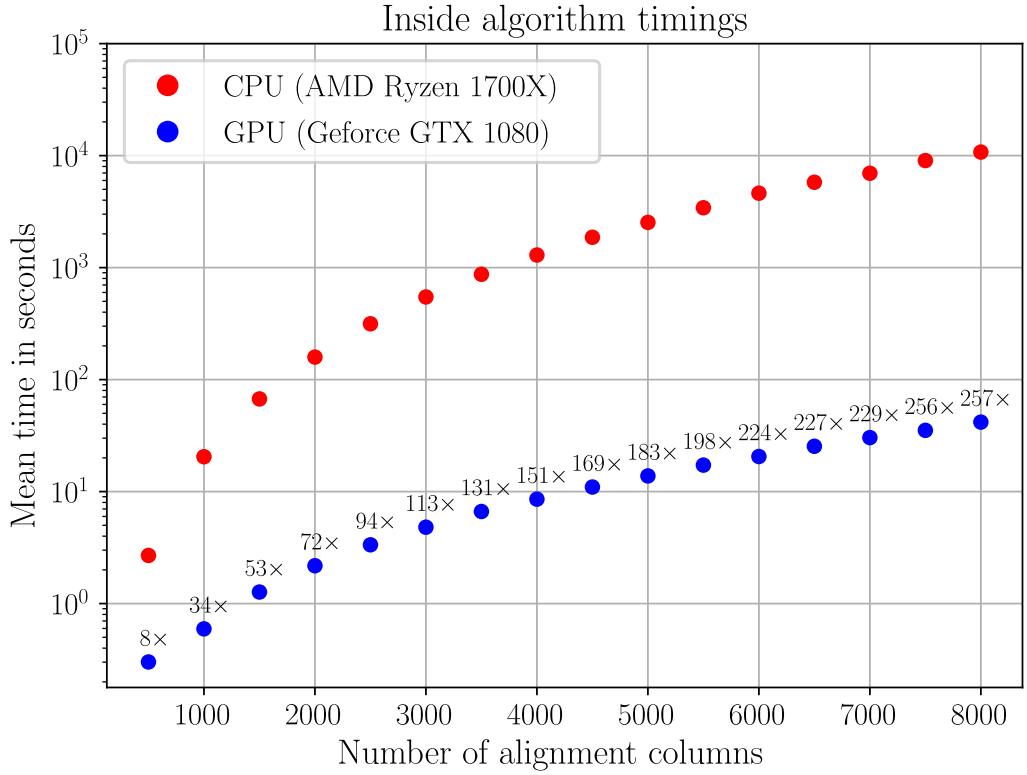


Figure 2.10: Inside algorithm timings in seconds ($\log_{10}$ axis) as a function of the number of alignment columns, with the fold speed-up of the GPU version over the CPU version indicated above the GPU timings.

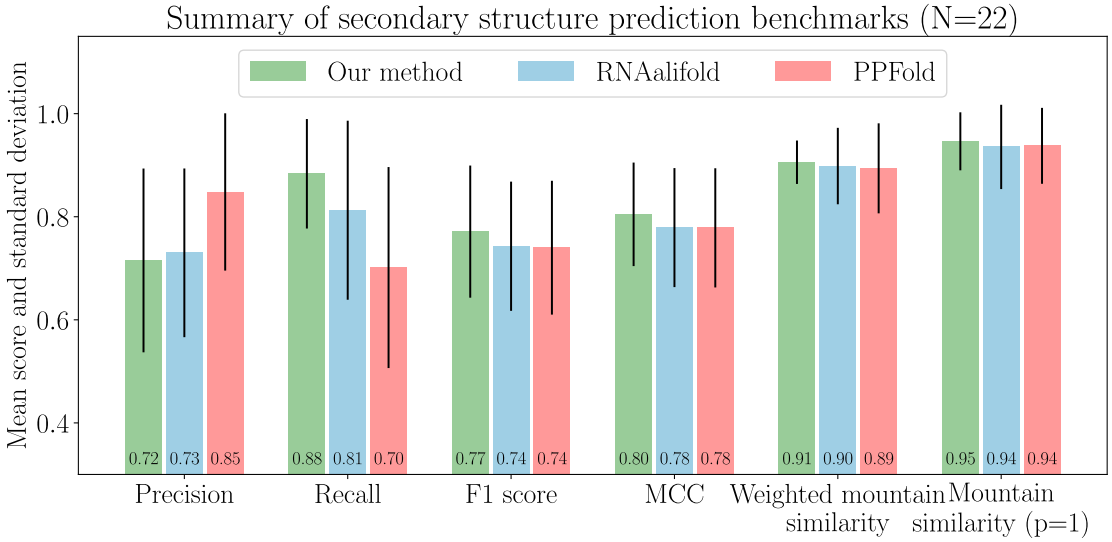


Figure 2.11: Summary of secondary structure prediction benchmarks

across most datasets compared to the single-threaded CPU implementation.

Figure 2.10 shows CPU and GPU timings for the inside matrix calculations. The number of computational steps in this case is expected to grow $\mathcal{O}(L^3)$ where L is the number of alignment columns in a particular dataset. A 50 to 200 fold speed-up was achieved for moderate size datasets, with the fold speed-ups for larger datasets being bigger, due to larger datasets better saturating the GPU.

The speed-ups seen here are significant, enabling us to analyse datasets which would typically be considered intractable. Note that CPU and GPU implementations were also developed for the outside algorithm with similar speed-ups seen (Figure A.1 in the appendix), although a GPU implementation was less important for the outside algorithm as it is only occasionally used to compute posterior base-pairing probabilities.

2.3.4 Benchmarks of RNA secondary structure prediction

Whilst our model was not designed to predict RNA secondary structure, the expected base-pairing and unpairing probabilities can be obtained (Section 2.2.15) and an MEA secondary structure calculated (Section 2.2.16), implying that the unconstrained co-evolution model can predict a canonical secondary structure for a given alignment.

Our method was compared to two comparative methods of RNA secondary structure prediction: RNAalifold and PPfold. The three methods were benchmarked on 22 alignments each having a corresponding experimentally determined canonical RNA secondary structure from the RFAM database (Burge *et al.*, 2012). Five different measures were used to compute predictive accuracy (described in Section 2.2.20).

Figure 2.11 show the results. Our method has lower precision but higher recall than the other two methods, implying that it predicts more base-pairs (higher recall), but with a higher number of false-positives (lower precision). In the F1-score and MCC measures, two measures that combine precision and recall, our method performs slightly better than RNAalifold and PPfold. Our method also performs marginally better in the weighted mountain similarity, and mountain similarity measures – two measures that take into account the overall 'shape' of the secondary structures being compared.

Overall, our method performs on a par with the other two well-established methods of comparative RNA structure prediction. This result was surprising given that the model was not tuned for secondary structure prediction. For example, maximum likelihood inference was used to determine the parameters, where the coevolution parameters (λ_{GC}, λ_{AU}, and λ_{GU}) were free to vary without strong prior knowledge of their values imposed. In the future we might attempt to improve the predictive accuracy of our structure predictions by performing Bayesian or MAP inference of the parameters using a prior learnt from a training dataset of alignments and corresponding reference structures.

It is worth noting that for particular values of the coevolution parameters (λ_{GC}, λ_{AU}, and λ_{GU}) the stationary distribution over dinucleotides (2.6) will become suitably concentrated amongst the canonical base-pairs, permitting secondary structure prediction from a single sequence; however, since this was not the goal of our model we did not pursue this.

2.3.5 The role of GU and GT base-pairs in single-stranded RNA and DNA

It is well-established that GU pairs can hydrogen bond in RNA to form base-pairs and that they are relatively weaker than GC and AU base-pairings (Rousset *et al.*, 1991). Despite being weaker, GU base-pairings are known to be involved in functional RNA structures (Gautheret *et al.*, 1995). Less well understood is the role of GT base-pairings in DNA. There are some reports in the literature of the role of GT base-pairings in double-stranded DNA helices (Early *et al.*, 1978; Ho *et al.*, 1985), but none were found in a literature search of GT base-pairings in single-stranded DNA. Whilst, we are unable study the strength of these base-pairing interactions via wet-lab experiments, we can use the modifications made to the M95 model in Section 2.2.2 to analyse alignments for evidence of evolutionary forces favouring the maintenance of formation of GT dinucleotides at base-paired positions. Furthermore, we can do so whilst accounting for an unknown secondary structure shared amongst an alignment of sequences.

Likelihood ratio tests (LRTs) for three of the five single-stranded DNA virus genome datasets tested (Human bocavirus, beet curly top virus, and tomato yellow leaf curl virus. $\Delta_{5,4}$ in tables below figures 2.13 - 2.15, respectively) showed no significant difference between the unconstrained model (model 5) and a model with GT coevolution restricted to be neutral ($\lambda_{\mathrm{GT}} \coloneqq 1$; model 3). With the Wheat Dwarf Virus dataset ($\Delta_{5,4}$ in table below Figure 2.12) meeting the $p < 0.05$ significance threshold for rejecting the GT neutral model, and Maise Streak Virus dataset ($\Delta_{5,4}$ in table below Figure 2.16) meeting the $p < 0.005$ significance threshold for rejecting the GT neutral model.

On the other hand, the GU neutral model was rejected in favour of the unconstrained model for all five non-coding RNA datasets (RF00001, RF00003, RF00010, RF00379, and RF01846. $p < 0.0005$ in all five cases. $\Delta_{5,4}$ in tables below figures 2.17 - 2.21). This is strong evidence of many GU dinucleotides being under selective maintenance in non-coding RNAs. This is not surprising as the primary selective

force acting on non-coding RNAs is expected to be the maintenance of functional secondary structure elements, rather than information storage and transfer functionality as in the case of mRNAs.

For four of the five RNA virus datasets tested (Rhinovirus A, Tobamovirus, human poliovirus 1, and foot-and-mouth disease virus. $\Delta_{5,4}$ in tables below figures 2.23 - 2.26) LRTs rejected the GU neutral model in favour of the unconstrained model ($p < 0.0005$ in all four cases). However, the GU neutral model could not be rejected in favour of the unconstrained for the Hepatitis A virus dataset ($\Delta_{5,4}$ in table below Figure 2.22).

Overall these results suggest that GT dinucleotides are under weak selective maintenance in single-stranded DNA virus genomes, and strong selective maintenance in RNA virus genomes and non-coding RNAs. This may suggest that GT base-pairings in DNA are chemically weaker relative to GU base-pairings in RNA and hence do not play a significant role in the formation of DNA secondary structure. We were unable to find measurements of free energies in the literature comparing the relative strengths of GT and GU base-pairings, therefore this appears to be a worthwhile area of future computational or experimental study.

An interesting possibility which we did not investigate here would be to analyse the mRNA transcripts of transcribed DNA virus genomes. Many DNA virus genomes are bidirectionally transcribed into mRNAs, implying that some mRNAs will be reverse complementary to the DNA genome, whereas some will be transcribed in the same direction as the DNA genome. Such an analysis may lead to evidence of selection acting to maintain base-pairings within mRNA transcripts, which would be suggestive of functional secondary structure elements within mRNA transcripts. Note that in cases of reverse complementarity, the only distinguishing dinucleotide is GU, as its complement is CA, whereas the complements of GC and AU are CG and UA, respectively, and hence would be treated the same by the model.

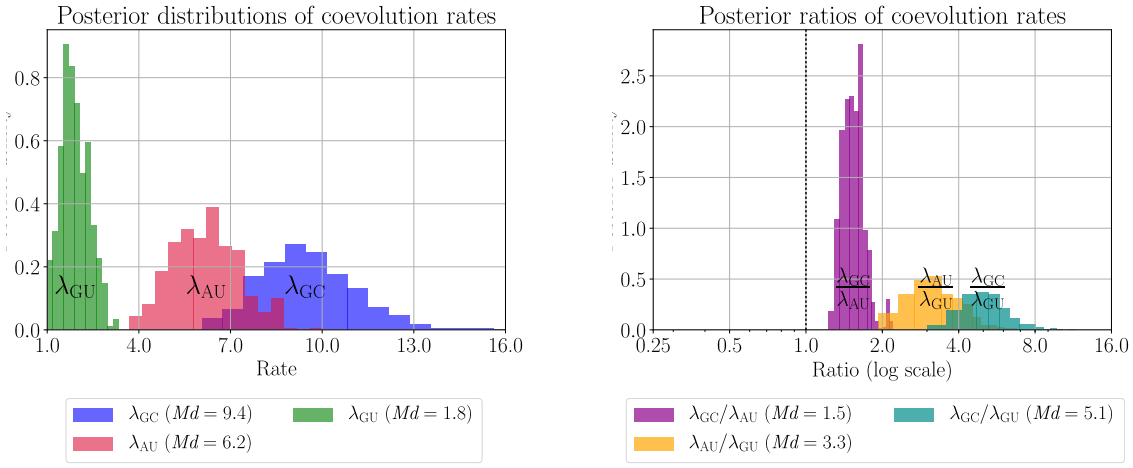
2.3.6 Relative coevolution rates

The relative selective strengths of the coevolution rates associated with GC, AU and GU dinucleotides were compared across both DNA and RNA virus genomes. The original M95 model assumed that $\lambda_{\mathrm{GC}} \coloneqq \lambda_{\mathrm{AU}}$; however, experimental evidence shows that GC base-pairings are chemically stronger than AU base-pairings in RNA (Mathews *et al.*, 1999), with both being substantially stronger than GU base-pairings.

To assess whether $\lambda_{\mathrm{GC}} \coloneqq \lambda_{\mathrm{AU}}$ is a reasonable assumption, we performed LRTs comparing the unconstrained model to a GC-AU constrained model (*i.e.* $\lambda_{\mathrm{GC}} \coloneqq \lambda_{\mathrm{AU}}$. $\Delta_{5,3}$ in tables below figures 2.12 - 2.26). For 14 of the 15 datasets, LRTs rejected the GC-AU constrained model in favour of the unconstrained model. The only exception being the human poliovirus 1 dataset, where the GC-AU constrained model could not be rejected.

The relative chemical strengths of GC, AU, and GU base-pairs are partially due to the number of hydrogen bonds that form between their constituent bases: three for GC base-pairs, two for AU base-pairs, and two for GU base-pairs. Despite GU forming the same number of hydrogen bonds as in AU, the physical geometry of the constituent bases leads to a relatively weaker interaction (Varani and McClain, 2000). Under the hypothesis that the fitness of RNA secondary structures will be positively correlated with their chemical stability, we expect that the relative chemical strengths associated with the three canonical base-pairs would reflect in the relative coevolution rates inferred by PINNACLE.

We addressed this question by comparing the posterior distributions of ratios of all three combinations of coevolution rates. The posterior distributions of the ratios (Right plot in figures 2.12 - 2.21) across all the DNA virus genome datasets and non-coding RNA datasets showed the following pattern of posterior probabilities for the ratios: $p(\frac{\lambda_{\mathrm{GC}}}{\lambda_{\mathrm{AU}}} > 1) > 0.99$, $p(\frac{\lambda_{\mathrm{AU}}}{\lambda_{\mathrm{GU}}} > 1) > 0.99$, and $p(\frac{\lambda_{\mathrm{GC}}}{\lambda_{\mathrm{GU}}} > 1) > 0.99$. Matching the expected relationship for the three coevolution rates: $\lambda_{\mathrm{GC}} > \lambda_{\mathrm{AU}} > \lambda_{\mathrm{GU}}$. Interestingly, the same was true for only two of the five RNA virus genome datasets (Hepatitis A virus and foot-and-mouth-disease virus. Top right of Figure 2.22 and

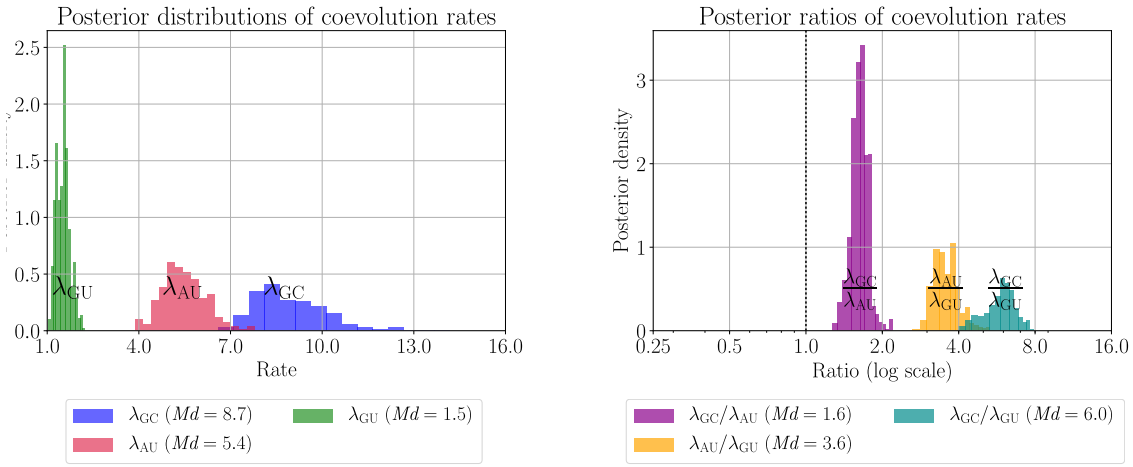


Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-10820.40		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-10738.09	$\Delta_{2,1} = 82.30$	***	5.90	5.90	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-10736.79	$\Delta_{3,2} = 1.31$	n.s.	7.39	7.39	1.85
4. $\lambda_{GU} \coloneqq 1$	13	-10733.63	$\Delta_{4,2} = 4.47$	**	6.65	5.08	1.00
5. Unconstrained	14	-10731.59	$\Delta_{5,3} = 5.20$	**	8.02	5.63	1.57
			$\Delta_{5,4} = 2.04$	*			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

Figure 2.12: Wheat dwarf virus (WDV. ssDNA virus. Family: *Geminiviridae*. Genus: *Mastrevirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).

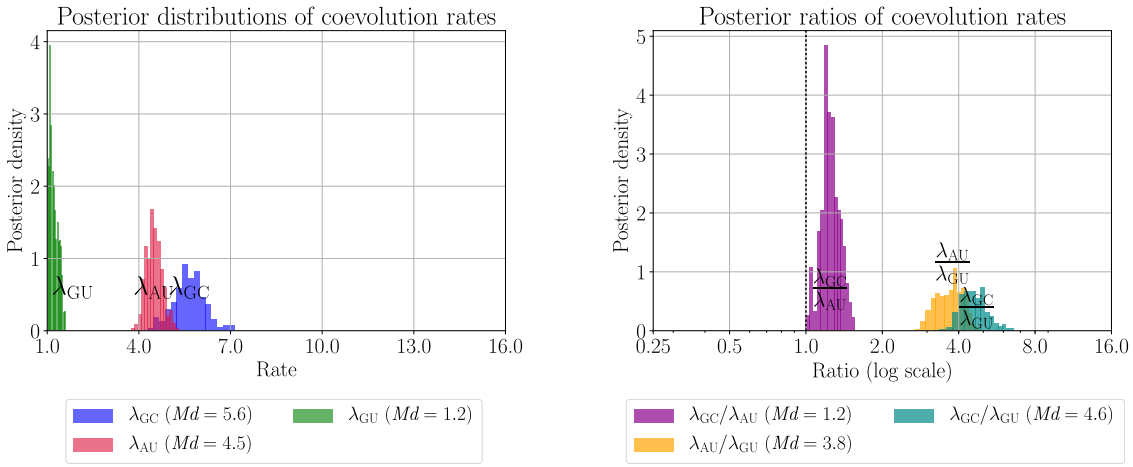
Figure 2.26), with two of the remaining three showing no clear pattern (Tobamovirus and human poliovirus 1. Top right of Figure 2.24 and Figure 2.25), and Rhinovirus A (top right of Figure 2.23) showing that GU dinucleotides were strongly favoured over AU dinucleotides at base-paired sites. This last result was surprising, a possible explanation is that the Rhinovirus A dataset had a high sequence diversity, and as a result the sequences did not all share the same canonical secondary structure, leading to inconsistent patterns of base-pairing and hence coevolution.



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-32331.65		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-32008.56	$\Delta_{2,1} = 323.08$	***	5.40	5.40	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-32008.23	$\Delta_{3,2} = 0.33$	n.s.	5.40	5.40	1.10
4. $\lambda_{GU} \coloneqq 1$	13	-31987.81	$\Delta_{4,2} = 20.75$	***	7.15	4.54	1.00
5. Unconstrained	14	-31985.98	$\Delta_{5,3} = 22.25$	***	7.59	4.60	1.25
			$\Delta_{5,4} = 1.84$	n.s.			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

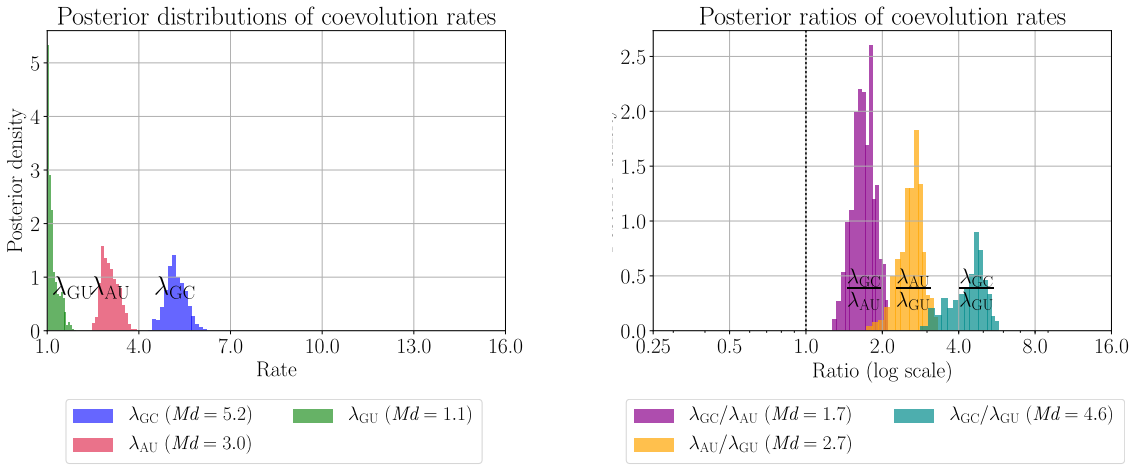
Figure 2.13: Human bocavirus (ssDNA virus. Family: *Parvoviridae*. Genus: *Bocaparvovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-32630.22		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-32098.80	$\Delta_{2,1} = 531.41$	***	4.57	4.57	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-32098.08	$\Delta_{3,2} = 0.72$	n.s.	4.80	4.80	1.19
4. $\lambda_{GU} \coloneqq 1$	13	-32094.09	$\Delta_{4,2} = 4.71$	**	5.19	4.27	1.00
5. Unconstrained	14	-32093.56	$\Delta_{5,3} = 4.51$	**	5.38	4.28	1.18
			$\Delta_{5,4} = 0.53$	n.s.			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

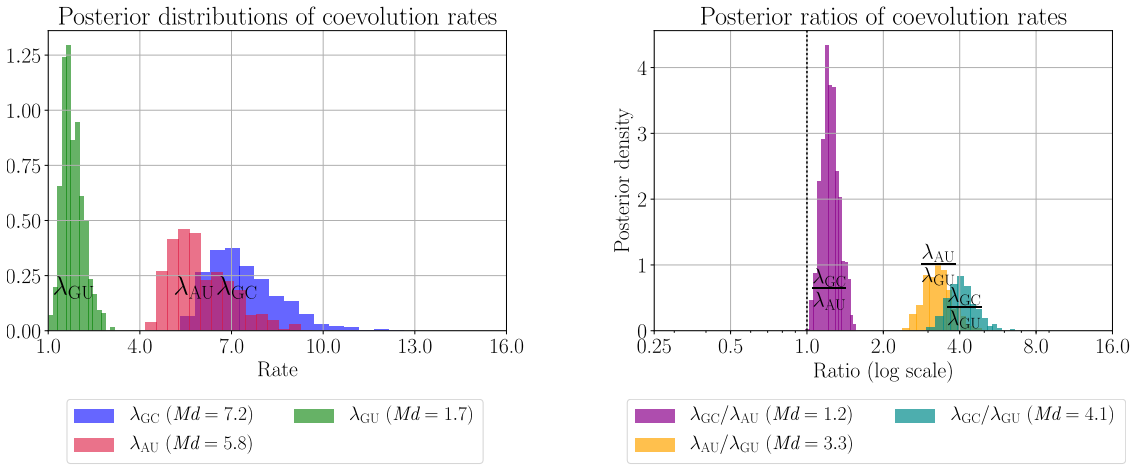
Figure 2.14: Beet curly top virus; (BCTV. ssDNA virus. Family: *Geminiviridae*. Genus: *Curtovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-41125.95		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-40759.98	$\Delta_{2,1} = 365.97$	***	3.84	3.84	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-40757.22	$\Delta_{3,2} = 2.76$	*	4.13	4.13	1.48
4. $\lambda_{GU} \coloneqq 1$	13	-40743.07	$\Delta_{4,2} = 16.91$	***	5.02	2.94	1.00
5. Unconstrained	14	-40743.03	$\Delta_{5,3} = 14.19$	***	5.07	2.99	1.00
			$\Delta_{5,4} = 0.04$	n.s.			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

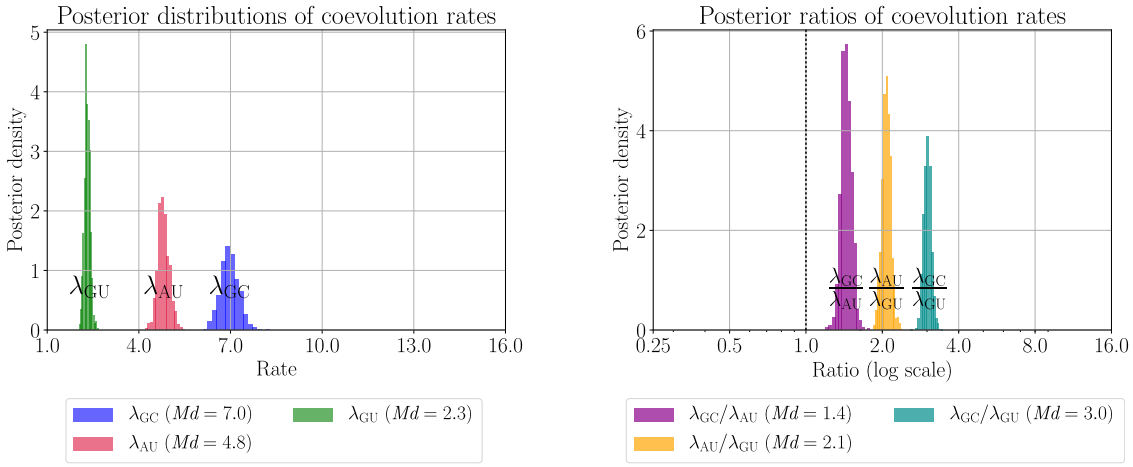
Figure 2.15: Tomato yellow leaf curl virus (TYLCV. ssDNA virus. Family: *Geminiviridae*. Genus: *Begomovirus*) . Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-15612.12		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-15351.36	$\Delta_{2,1} = 260.76$	***	6.59	6.59	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-15345.23	$\Delta_{3,2} = 6.13$	***	6.20	6.20	1.74
4. $\lambda_{GU} \coloneqq 1$	13	-15345.85	$\Delta_{4,2} = 5.51$	**	5.87	4.66	1.00
5. Unconstrained	14	-15341.48	$\Delta_{5,3} = 3.75$	*	6.54	5.22	1.57
			$\Delta_{5,4} = 4.37$	**			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

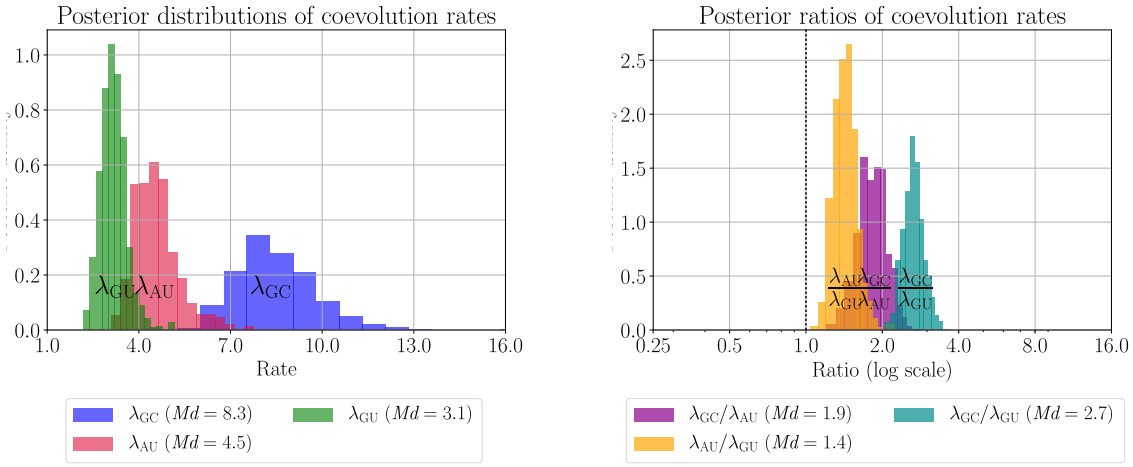
Figure 2.16: Maise streak virus (MSV. ssDNA virus. Family: *Geminiviridae*. Genus: *Mastrevirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-31502.65		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-28113.97	$\Delta_{2,1} = 3388.67$	***	6.60	6.60	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-27927.96	$\Delta_{3,2} = 186.01$	***	5.65	5.65	2.16
4. $\lambda_{GU} \coloneqq 1$	13	-28077.49	$\Delta_{4,2} = 36.48$	***	8.23	5.37	1.00
5. Unconstrained	14	-27897.66	$\Delta_{5,3} = 30.30$	***	6.84	4.63	2.25
			$\Delta_{5,4} = 179.83$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

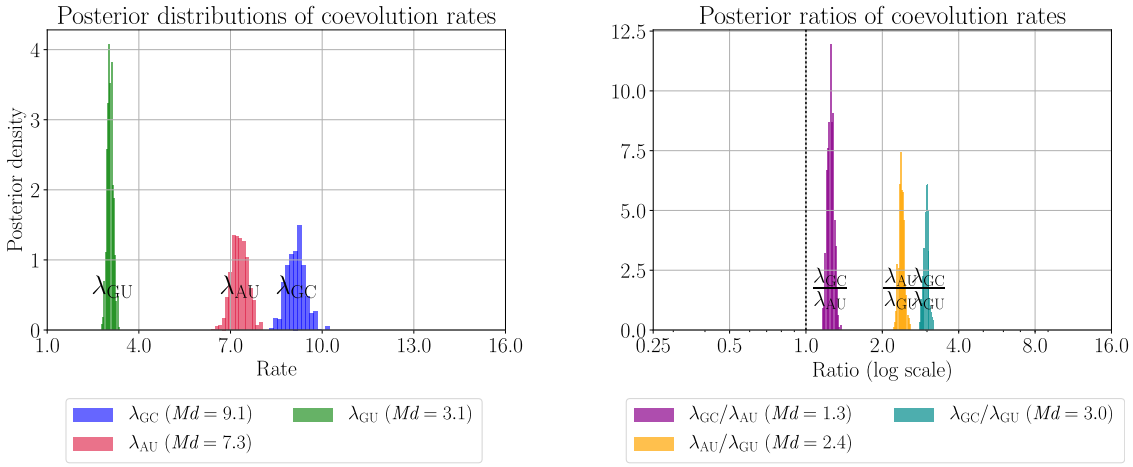
Figure 2.17: RF00001 (Non-coding RNA, 5S ribosomal RNA). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-4975.55		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-4713.41	$\Delta_{2,1} = 262.14$	***	3.76	3.76	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-4663.68	$\Delta_{3,2} = 49.73$	***	6.19	6.19	2.83
4. $\lambda_{GU} \coloneqq 1$	13	-4711.20	$\Delta_{4,2} = 2.21$	n.s.	4.13	3.12	1.00
5. Unconstrained	14	-4653.04	$\Delta_{5,3} = 10.64$	**	7.98	4.29	2.97
			$\Delta_{5,4} = 58.16$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

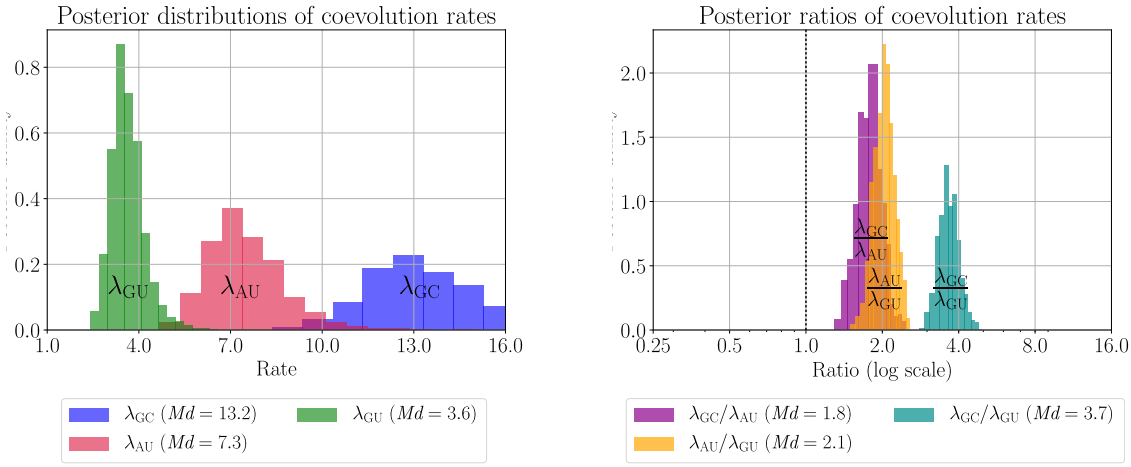
Figure 2.18: RF00003 (Non-coding RNA, U1 spliceosomal RNA). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-96436.32		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-85092.09	$\Delta_{2,1} = 11344.23$	***	6.46	6.46	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-84036.70	$\Delta_{3,2} = 1055.39$	***	8.51	8.51	3.14
4. $\lambda_{GU} \coloneqq 1$	13	-85067.98	$\Delta_{4,2} = 24.10$	***	8.88	7.07	3.01
5. Unconstrained	14	-84007.62	$\Delta_{5,3} = 29.07$	***	8.88	7.07	3.01
			$\Delta_{5,4} = 1060.36$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

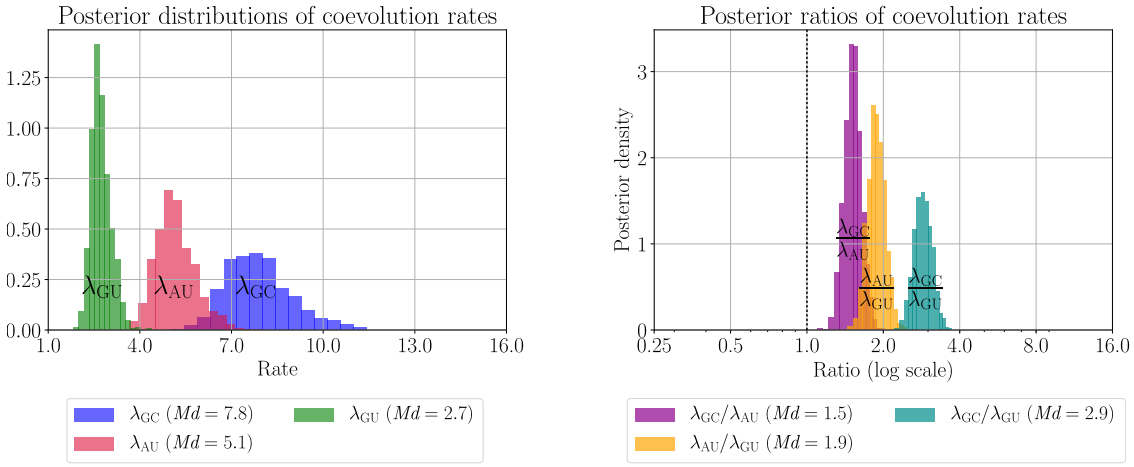
Figure 2.19: RF00010 (Non-coding RNA, Bacterial RNase P class A). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-11098.55		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-10539.47	$\Delta_{2,1} = 559.08$	***	7.83	7.83	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-10484.74	$\Delta_{3,2} = 54.73$	***	11.23	11.23	3.99
4. $\lambda_{GU} \coloneqq 1$	13	-10523.44	$\Delta_{4,2} = 16.03$	***	8.90	4.74	1.00
5. Unconstrained	14	-10470.32	$\Delta_{5,3} = 14.42$	***	13.24	7.28	3.57
			$\Delta_{5,4} = 53.12$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

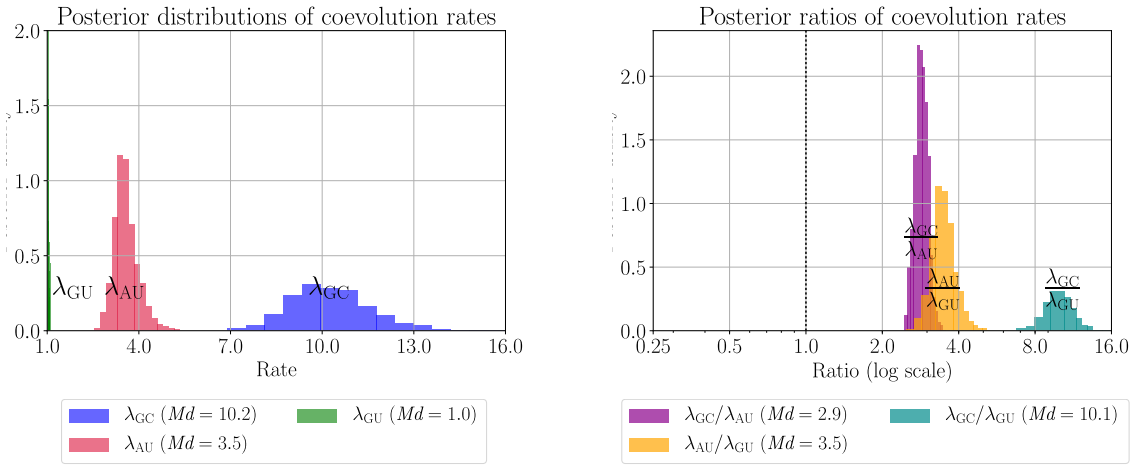
Figure 2.20: RF00379 (Non-coding RNA, YdaO/YuaA leader). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-11623.62		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-11107.97	$\Delta_{2,1} = 515.65$	***	4.83	4.83	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-11056.69	$\Delta_{3,2} = 51.29$	***	6.05	6.05	2.53
4. $\lambda_{GU} \coloneqq 1$	13	-11101.49	$\Delta_{4,2} = 6.49$	*	5.48	4.04	1.00
5. Unconstrained	14	-11043.98	$\Delta_{5,3} = 12.70$	***	7.58	4.98	2.64
			$\Delta_{5,4} = 57.51$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

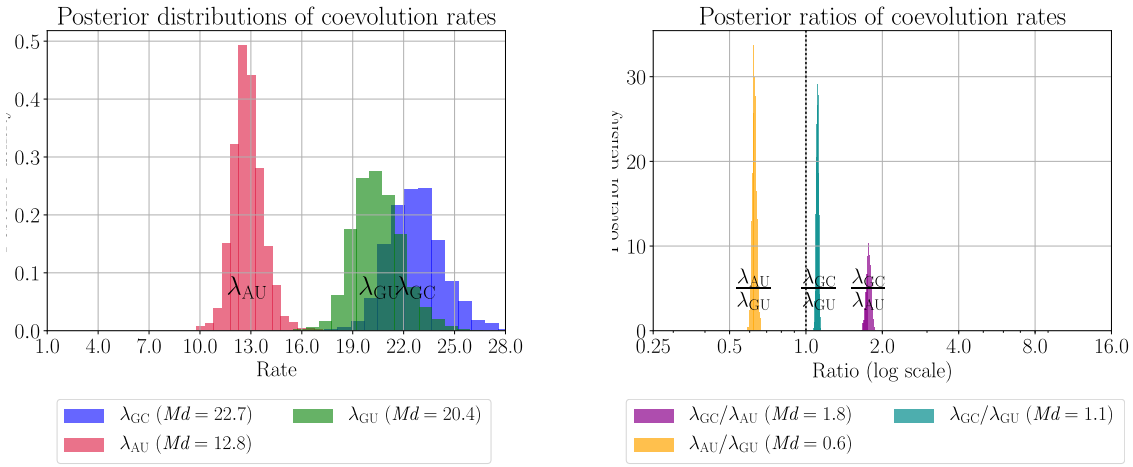
Figure 2.21: RF01846 (Non-coding RNA, Fungal small nucleolar RNA U3). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-64189.57		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-63899.43	$\Delta_{2,1} = 290.14$	***	2.83	2.83	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-63899.43	$\Delta_{3,2} = 0.0$	n.s.	2.88	2.83	1.00
4. $\lambda_{GU} \coloneqq 1$	13	-63755.35	$\Delta_{4,2} = 144.08$	***	12.81	4.26	1.00
5. Unconstrained	14	-63755.31	$\Delta_{5,3} = 144.08$	***	12.81	4.26	1.00
			$\Delta_{5,4} = 0.04$	n.s.			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

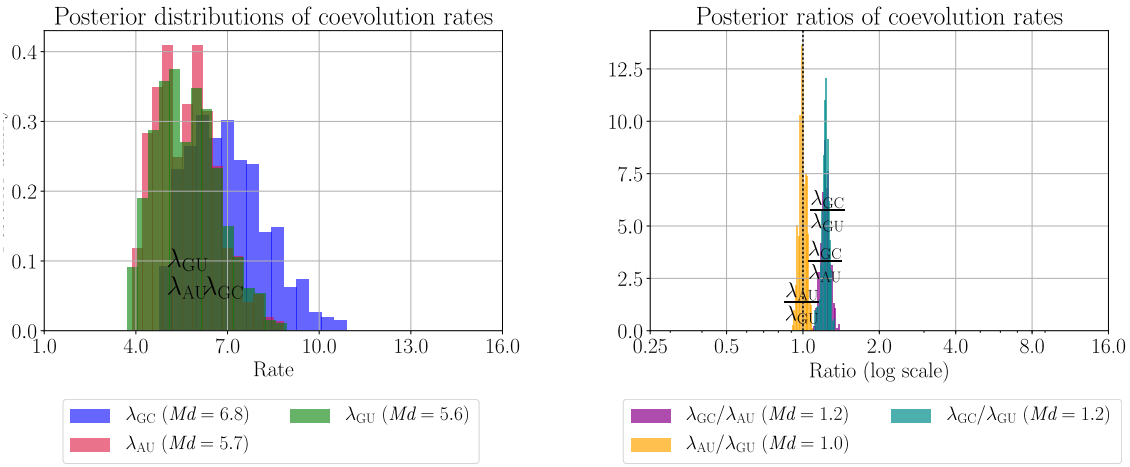
Figure 2.22: Hepatitis A virus (ssRNA virus. Family: *Picornaviridae*. Genus: *Hepatovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-224041.36		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-219773.78	$\Delta_{2,1} = 4267.58$	***	4.91	4.91	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-217484.94	$\Delta_{3,2} = 2288.84$	***	22.23	22.23	23.79
4. $\lambda_{GU} \coloneqq 1$	13	-219614.83	$\Delta_{4,2} = 158.95$	***	7.70	4.37	1.00
5. Unconstrained	14	-217136.20	$\Delta_{5,3} = 348.73$	***	24.60	13.62	21.92
			$\Delta_{5,4} = 2478.62$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

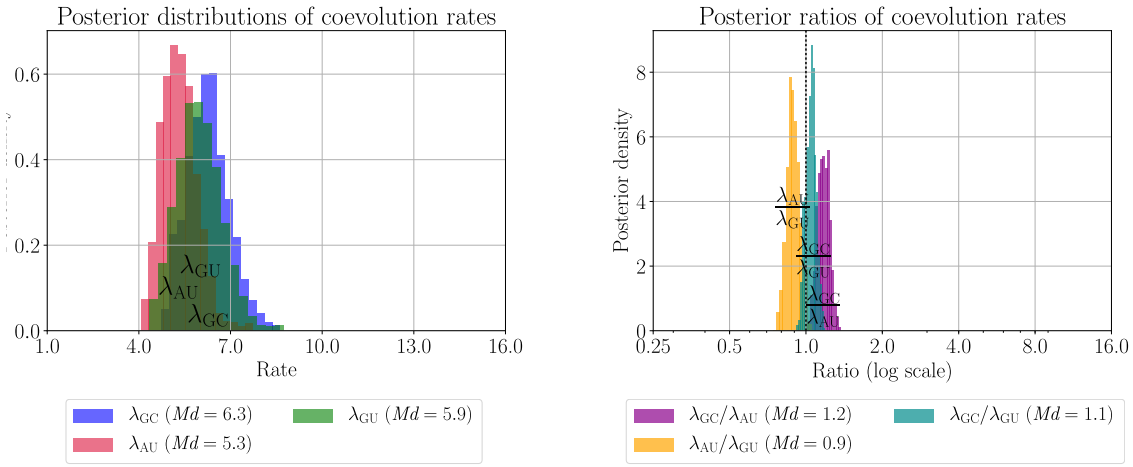
Figure 2.23: Rhinovirus A (ssRNA virus. Family: *Picornaviridae*. Genus: *Enterovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-89261.82		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-88916.79	$\Delta_{2,1} = 345.03$	***	2.67	2.67	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-88776.59	$\Delta_{3,2} = 140.20$	***	8.68	8.68	7.59
4. $\lambda_{GU} \coloneqq 1$	13	-88906.32	$\Delta_{4,2} = 10.47$	***	3.18	2.53	1.00
5. Unconstrained	14	-88767.09	$\Delta_{5,3} = 9.49$	***	9.38	7.64	7.53
			$\Delta_{5,4} = 139.23$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

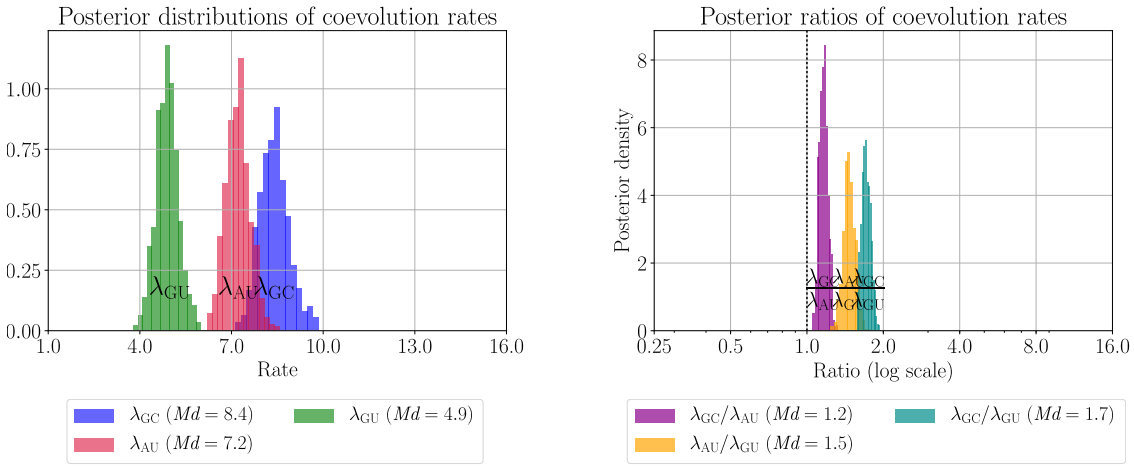
Figure 2.24: Tobamovirus (ssRNA virus. Family: *Virgaviridae*. Genus: *Tobamovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-65404.81		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-65016.58	$\Delta_{2,1} = 388.23$	***	6.16	6.16	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-64966.74	$\Delta_{3,2} = 49.84$	***	6.05	6.05	6.10
4. $\lambda_{GU} \coloneqq 1$	13	-65013.50	$\Delta_{4,2} = 3.07$	*	5.81	5.28	1.00
5. Unconstrained	14	-64966.32	$\Delta_{5,3} = 0.42$	n.s.	5.93	5.14	5.60
			$\Delta_{5,4} = 47.19$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

Figure 2.25: Human poliovirus 1 (HPV1. ssRNA virus. Family: *Picornaviridae*. Genus: *Enterovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).



Model	Params	MLL	ΔMLL	p-value	$\hat{\lambda}_{GC}$	$\hat{\lambda}_{AU}$	$\hat{\lambda}_{GU}$
1. Unpaired	10	-117142.57		-	1.00	1.00	1.00
2. $\lambda_{GC} \coloneqq \lambda_{AU}, \lambda_{GU} \coloneqq 1$	12	-114989.24	$\Delta_{2,1} = 2153.33$	***	5.72	5.72	1.00
3. $\lambda_{GC} \coloneqq \lambda_{AU}$	13	-114615.59	$\Delta_{3,2} = 373.66$	***	8.59	8.59	4.97
4. $\lambda_{GU} \coloneqq 1$	13	-114986.98	$\Delta_{4,2} = 2.27$	*	5.38	6.03	1.00
5. Unconstrained	14	-114610.85	$\Delta_{5,3} = 4.73$	**	9.31	8.16	5.21
			$\Delta_{5,4} = 376.12$	***			

* $p < 0.05$; ** $p < 0.005$; *** $p < 0.0005$; n.s. = not significant

Figure 2.26: Foot-and-mouth-disease virus (FMDV. ssRNA virus. Family: *Picornaviridae*. Genus: *Aphthovirus*). Bayesian posterior distribution of coevolution rates (top left) and ratios of coevolution rates (top right), and maximum likelihood estimates under five different models and likelihood ratio test p-values comparing selected nested models (Table below).

2.3.7 Degrees of coevolution are correlated with experimental SHAPE-MaP quantities

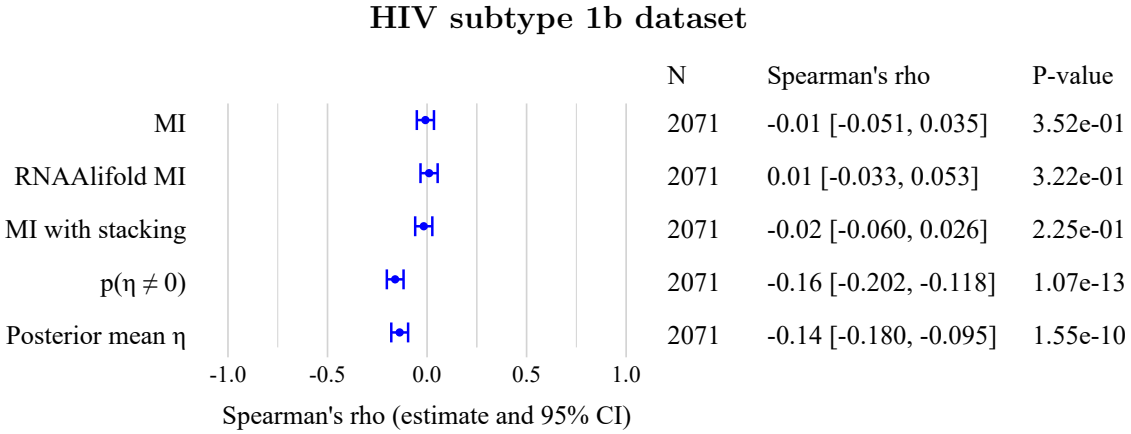
A notable example of a large RNA structure that has been partially experimentally determined is the HIV-1M subtype B NL4-3 isolate (Watts *et al.*, 2009; Siegfried *et al.*, 2014). Rather than relying solely on computational techniques for the determination of RNA secondary structure of the viral genome (consisting of 9173 nucleotides), the hybrid experimental-computational SHAPE-MaP (Selective 2'-hydroxyl acylation analysed by primer extension and mutational profiling; Siegfried *et al.* (2014)) approach was used to model the structure of the NL4-3 isolate. The SHAPE-MaP approach preferentially chemically mutates unpaired nucleotides, allowing the mutated nucleotides to be identified using DNA sequencing following reverse transcription. The SHAPE-MaP reactivity information is then used to constrain a thermodynamic RNA folding algorithm, allowing the construction of a secondary structure model which is consistent with the experimental data.

Three non-evolutionary computational measures covariation (A. Mutual information, B. RNAalifold mutual information, and C. Mutual information with stacking) as described in Lindgreen *et al.* (2006) and two evolutionary measures of coevolution inferred by PINNACLE (D. Posterior probability $\eta \neq 1$, and E. Posterior mean η), were compared to experimental SHAPE-MaP reactivities and partially experimental SHAPE-MaP pairing probabilities at base-paired sites corresponding to three different datasets (an HIV 1b dataset, an HIV group 1M dataset, and a SIV dataset). Simian Immunodeficiency Virus (SIV) is a species of virus closely related to HIV that infects non-human primates. The SHAPE-MaP reactivities, SHAPE-MaP pairing probabilities and base-pairings were derived from a SHAPE-MaP analysis of the HIV NL4-3 sequence (Watts *et al.*, 2009) when analysing the two HIV datasets and a SHAPE-MaP analysis of the SIVmac239 sequence (Pollom *et al.*, 2013) when analysing the SIV dataset. Given that high SHAPE reactivities indicate unpairing, it was expected that degrees of coevolution (or covariation) would be negatively correlated with SHAPE-reactivities. Conversely, given that paired nucleotides are

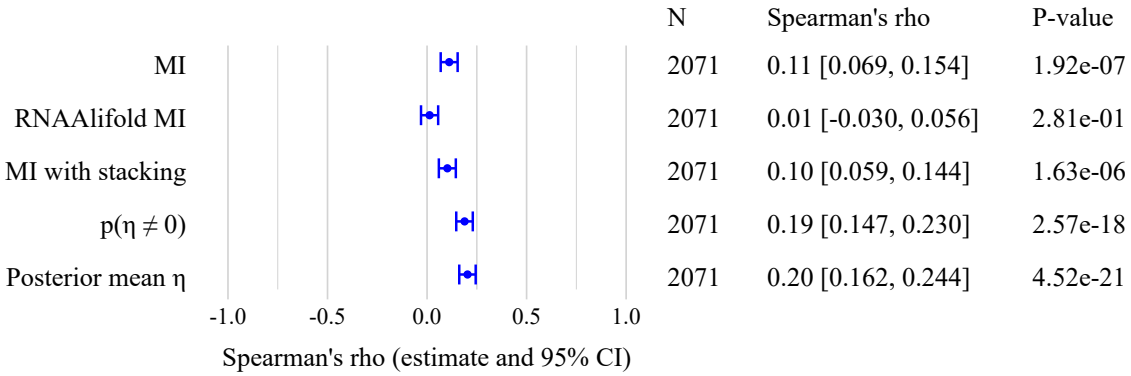
expected to be selectively maintained due to structure-related functional importance, a positive correlation between degrees of coevolution (or covariation) and SHAPE pairing probabilities was expected.

The results are depicted in Figures 2.27, 2.28, and 2.29 for the HIV subtype 1b, HIV group 1M, and SIV datasets, respectively. For all three datasets the two measures of coevolution (D and E) were statistically significantly correlated with both the SHAPE-MaP reactivities and SHAPE-MaP pairing probabilities using Spearman's rank correlation test, and the correlations were in the expected direction (negatively correlated for SHAPE-MaP reactivities and positively correlated for SHAPE-MaP pairing probabilities). For all three datasets the correlation coefficients were significantly stronger in the expected direction for the two coevolution measures (D and E) compared to the three covariation measures (A,B, and C; see the 95% confidences intervals for Spearman's rho). It should be noted that whilst many of the correlations were statistically significant, the magnitudes of the correlations were weak, to the point where the correlations were not easily visually discernible in scatterplots comparing coevolution/covariation and the SHAPE-MaP quantities.

Curiously, for the SIV dataset SHAPE-MaP reactivities were detected as being significantly positively correlated with the three measures of covariation (A, B, and C in Figure 2.29), rather than negatively correlated as expected. There is broad evidence to suggest that base-paired sites in a functionally important RNA structure tend to be be more conserved (less variable) due to being under selective constraint (Muhire *et al.*, 2014; Tuplin *et al.*, 2004) and that double-stranded RNA (i.e. base-paired positions) is less susceptible to basal mutational processes (Lindahl and Nyberg, 1974). Conversely, unpaired sites are expected to undergo relatively higher rates of mutation. These higher rates of mutation may cause the three non-evolutionary measures of covariation to be erroneously inflated, given that they do not fully account for site-to-site rate variation unlike the coevolution measures inferred under our model, which do. It should also be noted that the SIV dataset is highly diverse compared to the two HIV datasets. Given these factors, it is anticipated that weakly base-paired



(b) NL4-3 SHAPE-MaP reactivites

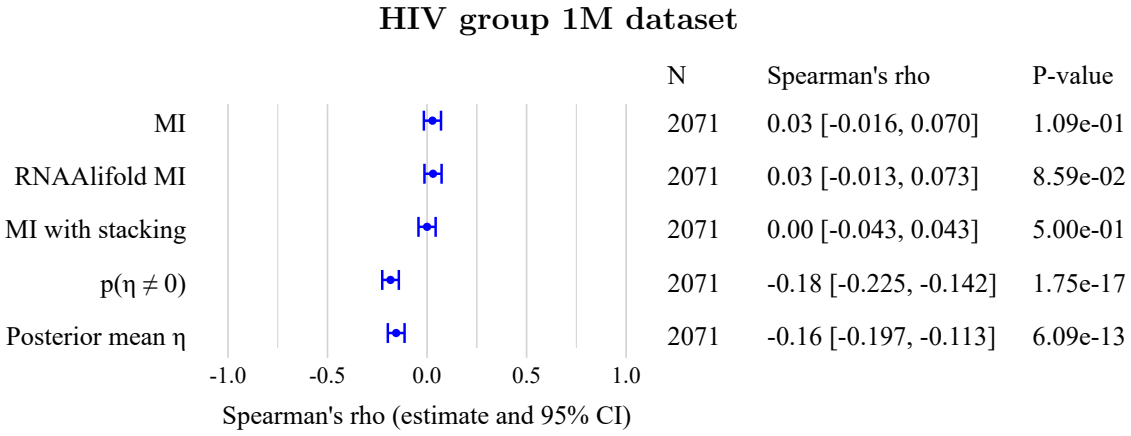


(c) NL4-3 SHAPE-MaP pairing probabilities

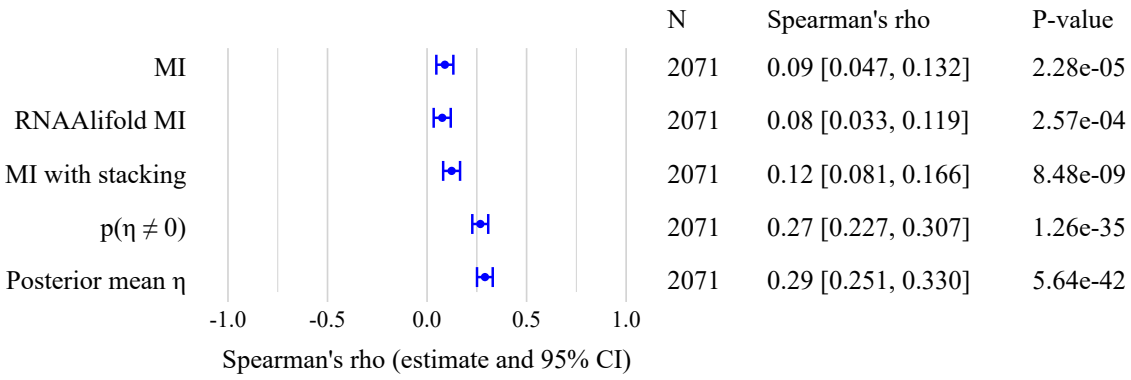
Figure 2.27: Spearman's correlations between five different measures of coevolution and base-pair averaged SHAPE-MaP reactivities (top) and base-pair averaged SHAPE-MaP pairing probabilities (bottom) for the HIV 1b dataset.

sites will have inflated degrees of covariation using measures A, B and C, which may explain the unexpected positive correlation.

Overall, these results provide some reassurance that our method is performing as expected and that the evolutionary measures of coevolution are more reliable than the three measures of covariation that do not take into account evolutionary dependencies amongst the sequences being analysed. The detected degrees of coevolution indicate that many of the base-pairings present in the NL4-3 secondary structure and SIV secondary structure models are real (i.e. the SHAPE-MaP secondary structures model are accurate) and that a large proportion of the predicted base-pairings have been selectively maintained, possibly since the common ancestor of the sequences being analysed in each dataset, likely due to having some degree of biological functionality related to RNA structure.

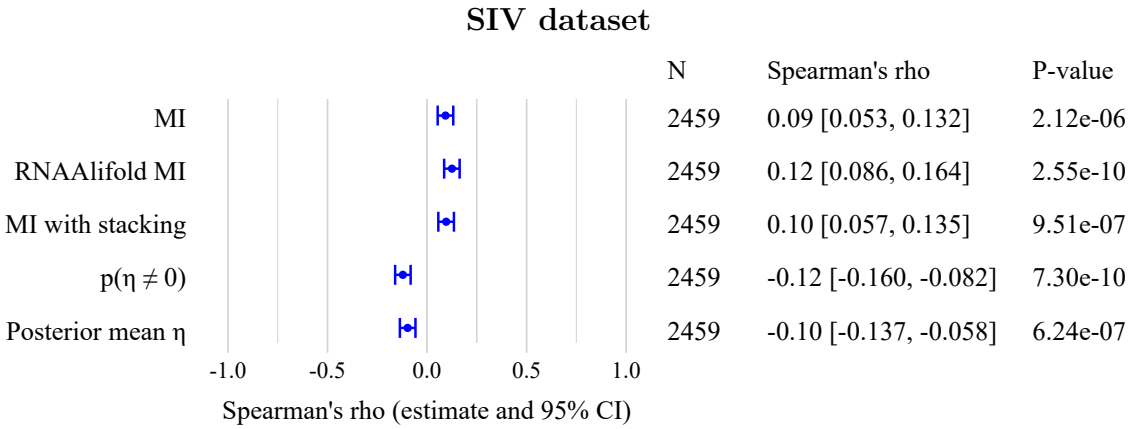


(b) NL4-3 SHAPE-MaP reactivites

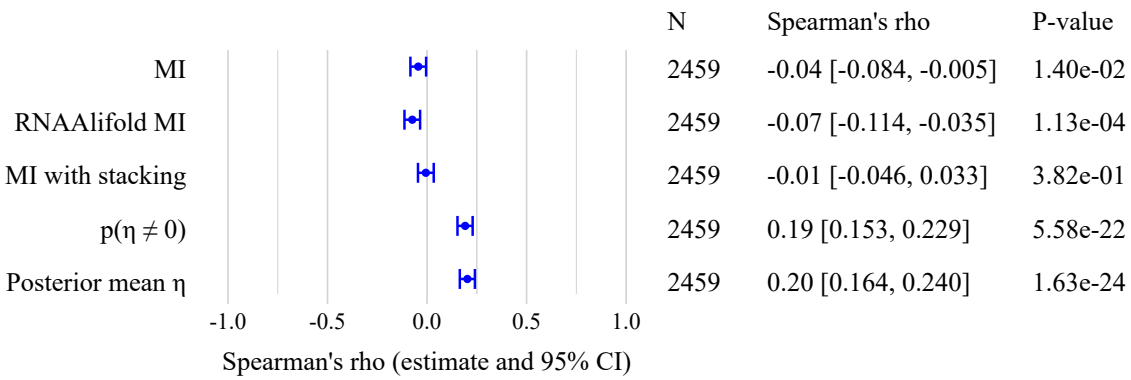


(c) NL4-3 SHAPE-MaP pairing probabilities

Figure 2.28: Spearman's correlations between five different measures of coevolution and base-pair averaged SHAPE-MaP reactivities (top) and base-pair averaged SHAPE-MaP pairing probabilities (bottom) for the HIV group 1M dataset.



(b) SIVmac239 SHAPE-MaP reactivites



(c) SIVmac239 SHAPE-MaP pairing probabilities

Figure 2.29: Spearman's correlations between five different measures of coevolution and base-pair averaged SHAPE-MaP reactivities (top) and base-pair averaged SHAPE-MaP pairing probabilities (bottom) for the SIV dataset.

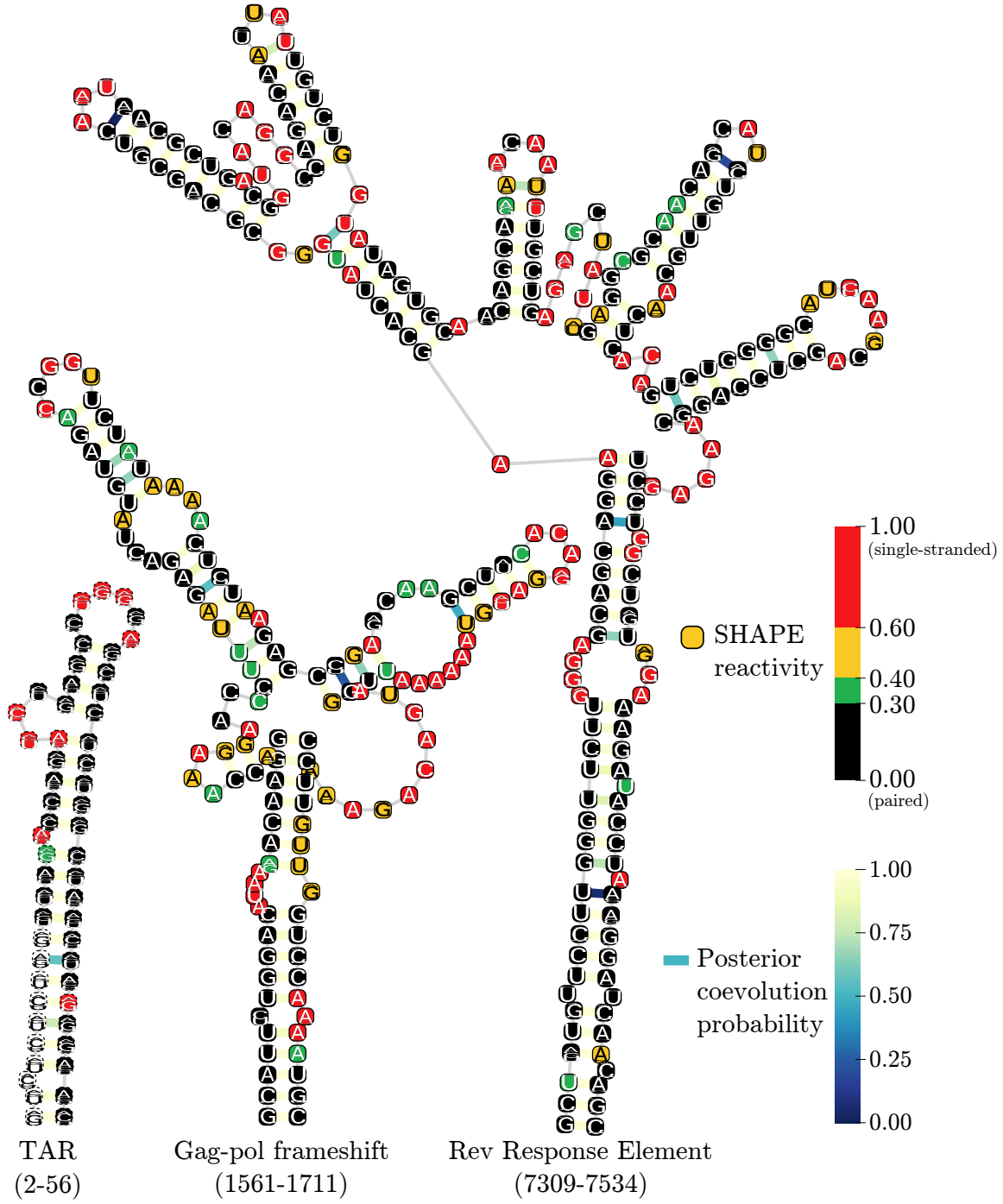


Figure 2.30: Visualisation of three well-known HIV-1 substructures with mapped NL4-3 SHAPE reactivities overlaid on nucleotides within the HIV-1 group M alignment using the same colour scheme as in Watts *et al.* (2009). Depicted within each circular nucleotide is a sequence logo summarising the nucleotide composition at the corresponding alignment position. The NL4-3 SHAPE-determined structure was used for visualisation and analysis purposes, with posterior coevolution probabilities inferred using the HIV-1 group M alignment depicted for each base-pair using coloured links - most base-pairs have high posterior coevolution probabilities (≈ 1) and are depicted using light yellow links to draw the reader's attention to the base-pairs with low posterior coevolution probabilities (blue links).

2.3.8 Ranking of substructures

The 1447 sequence HIV1-M alignment and canonical SHAPE-MaP secondary structure model were analysed using PINNACLE to produce an automatic ranking of 103 non-overlapping HIV NL4-3 SHAPE-MaP substructures ranked by the posterior probability of positive coevolution at base-paired sites within them. Notably, there appears to be an excess of substructures in the top ranking substructures (those with higher posterior coevolution probabilities) amongst the 103 that have been identified in the literature as having structure-related importance. The top ranked structure was the RRE (7276-7574, Figure 2.30). The RRE occurs in the genomes of all known HIVs and plays a crucial role in the regulation of HIV virion expression (Heaphy *et al.*, 1990; Daugherty *et al.*, 2010).

Both the 3' (9070-9138) and 5 (105-344)' untranslated regions (UTRs) were ranked 2nd and 3rd, respectively. Both these regions are non-coding and are known to contain secondary structures involved in regulation, therefore it is not surprising they have relatively higher posterior probabilities of coevolution. The 5' region is involved in regulation of translation (Damgaard *et al.*, 2004), whereas the 3' is believed to be involved in gene regulation (Watts *et al.*, 2009).

The 4th ranked substructure was the trans-activation response element (TAR, 2-56, Figure 2.30, Roy *et al.* (1990)). TAR is a trans-acting element that binds to the Tat protein (encoded by HIV) and is required for virus replication.

The 13th ranked substructure was the gag-pol frameshift (1561-1711, Figure 2.30, Chamorro *et al.* (1992)). This substructure regulates the ratio of HIV gag/gag-pol synthesis. Synthesis of the gag-pol polyprotein requires a -1 ribosomal frameshift, without this frameshift translation ends in synthesis of the gag protein alone.

The 25th ranked (1991-2326, longest continuous helix) was identified in both the SHAPE and SHAPE-MaP secondary structure models of HIV NL4-3, but has yet to be functionally characterized. Similarly, the 20th ranked substructure (1177-1312) has been identified in the SHAPE-MaP model as structurally well-defined, but also has no known function.

The apparent excess of substructures with established biological importance in the top ranking substructures suggests that the ranking may assist researchers in focusing their experimental analysis on those substructures that are most likely to be biologically relevant.

This pattern is expected, as the top ranked substructures are those containing base-pairs with the highest posterior probabilities of positive coevolution - evidence of those base-pairs being selectively maintained, which is likely due to the substructures in which they occur having structure-related functional importance (Tuplin *et al.*, 2002; Cheng *et al.*, 2012; Muhire *et al.*, 2014).

A ranking of the 103 substructures was also performed using the experimentally determined SHAPE reactivities (Table A.3 in the appendix), with lower SHAPE reactivities indicating substructures with an excess of base-paired nucleotides, and higher SHAPE reactivities indicating substructures with an excess of unpaired nucleotides (Watts *et al.*, 2009). The coevolution and SHAPE rankings were fairly consistent: the RRE was ranked 1st in both rankings, the 3' UTR 2nd in both rankings, the 3'UTR 3rd in the coevolution ranking and 8th in the SHAPE ranking, the TAR 4th in coevolution ranking and 17th in the SHAPE ranking, the gag-pol frameshift 13th in the coevolution ranking and 6th in the SHAPE ranking, and the longest continuous helix 25th in the coevolution ranking and 7th in the SHAPE ranking.

Table 2.4: 103 non-overlapping HIV NL4-3 substructures ranked from highest to lowest z-score based on posterior probabilities of positive coevolution within them.

Rank	Position	Length	Name and reference	Median	z-score
1	7276 - 7574	299	Rev Response element (RRE, Heaphy *et al.* (1990); Mandal and Breaker (2004))	0.99	4.73
2	9070 - 9138	69	3' Untranslated region (3'UTR, Siegfried *et al.* (2014))	0.99	2.84
3	105 - 344	240	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.98	2.71
4	2 - 56	55	Trans-activation response element (TAR, Roy *et al.* (1990))	0.99	2.50
5	4158 - 4175	18		1.00	2.25
6	9042 - 9057	16		0.99	2.22
7	3205 - 3223	19		1.00	2.20
8	871 - 913	43		0.99	2.13
9	3797 - 4058	262		0.98	1.99
10	2015 - 2121	107		0.98	1.98
11	997 - 1014	18		1.00	1.77
12	961 - 996	36		0.99	1.71
13	1561 - 1711	151	Gag-pol frameshift (Chamorro *et al.*, 1992)	0.98	1.70
14	8500 - 8783	284		0.97	1.68

Rank	Position	Length	Name and reference	Median	z-score
15	58 - 104	47	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.99	1.67
16	5010 - 5022	13		0.99	1.63
17	6518 - 6540	23		0.98	1.63
18	921 - 930	10		1.00	1.57
19	2245 - 2260	16		0.99	1.55
20	1177 - 1312	136		0.97	1.48
21	4068 - 4098	31		0.99	1.43
22	9059 - 9067	9		0.99	1.34
23	3269 - 3290	22		0.99	1.28
24	9141 - 9170	30		0.99	1.26
25	2145 - 2171	27	Longest continuous helix (Siegfried *et al.*, 2014)	0.98	1.25
26	4099 - 4129	31		0.99	1.20
27	2135 - 2144	10		0.99	1.01
28	1375 - 1387	13		0.97	1.01
29	2268 - 2310	43		0.97	0.93
30	4797 - 4899	103		0.97	0.87

Rank	Position	Length	Name and reference	Median	z-score
31	1103 - 1141	39		0.97	0.81
32	6048 - 6066	19		0.96	0.72
33	8440 - 8454	15		0.95	0.71
34	2378 - 2429	52		0.99	0.67
35	6490 - 6501	12		0.98	0.66
36	4573 - 4586	14		0.97	0.66
37	8971 - 8980	10		0.99	0.62
38	2537 - 2586	50		0.97	0.55
39	3392 - 3427	36		0.96	0.54
40	1026 - 1068	43		0.99	0.47
41	5117 - 5158	42		0.99	0.42
42	398 - 485	88	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.98	0.40
43	4176 - 4473	298	Central polypurine tract (CPPT, Siegfried *et al.* (2014))	0.97	0.32
44	3040 - 3058	19		0.98	0.31
45	1728 - 1758	31		0.97	0.29
46	2476 - 2532	57		0.97	0.23

Rank	Position	Length	Name and reference	Median	z-score
47	5192 - 5427	236		0.97	0.21
48	2618 - 2917	300		0.96	0.14
49	5081 - 5096	16		0.93	0.13
50	4475 - 4491	17		0.97	0.13
51	795 - 849	55		0.97	0.11
52	1971 - 1986	16		0.90	0.10
53	7647 - 7692	46		0.97	0.06
54	4754 - 4770	17		0.92	0.02
55	7833 - 7840	8		0.93	-0.01
56	1810 - 1918	109		0.97	-0.02
57	4614 - 4647	34		0.96	-0.07
58	6616 - 6650	35		0.91	-0.27
59	3092 - 3176	85		0.94	-0.30
60	759 - 775	17		0.97	-0.35
61	946 - 957	12		0.81	-0.36
62	4938 - 4999	62		0.95	-0.36

Rank	Position	Length	Name and reference	Median	z-score
63	6667 - 6678	12		0.91	-0.37
64	1316 - 1348	33		0.89	-0.37
65	3242 - 3257	16		0.96	-0.37
66	2201 - 2238	38		0.95	-0.44
67	8226 - 8268	43		0.97	-0.45
68	4786 - 4795	10		0.92	-0.55
69	3307 - 3351	45		0.94	-0.55
70	3178 - 3190	13		0.96	-0.61
71	5793 - 6013	221		0.95	-0.67
72	501 - 714	214	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.96	-0.75
73	4520 - 4529	10		0.92	-0.75
74	3074 - 3089	16		0.55	-0.82
75	6544 - 6589	46		0.94	-0.83
76	6777 - 6802	26		0.83	-0.84
77	6078 - 6096	19		0.90	-0.93
78	6273 - 6287	15		0.79	-1.04

Rank	Position	Length	Name and reference	Median	z-score
79	7602 - 7613	12		0.76	-1.06
80	1075 - 1101	27		0.84	-1.12
81	2972 - 3038	67		0.89	-1.18
82	7896 - 8152	257		0.94	-1.34
83	1395 - 1559	165		0.92	-1.35
84	4678 - 4722	45		0.78	-1.38
85	6680 - 6702	23		0.89	-1.42
86	7614 - 7645	32		0.86	-1.47
87	5167 - 5188	22		0.86	-1.54
88	7705 - 7770	66		0.91	-1.80
89	8982 - 9039	58		0.85	-1.80
90	8797 - 8917	121		0.85	-1.83
91	8461 - 8498	38		0.88	-1.88
92	5471 - 5770	300		0.95	-1.89
93	7852 - 7894	43		0.91	-2.01
94	6302 - 6488	187		0.90	-2.05

Rank	Position	Length	Name and reference	Median	z-score
95	8169 - 8205	37		0.54	-2.20
96	6706 - 6767	62		0.30	-2.33
97	3429 - 3699	271		0.91	-2.42
98	8275 - 8348	74		0.73	-2.46
99	1760 - 1785	26		0.64	-2.67
100	4723 - 4752	30		0.62	-2.71
101	8354 - 8377	24		0.32	-3.09
102	6133 - 6272	140		0.58	-3.92
103	6861 - 7155	295		0.79	-4.27

2.4 Concluding remarks

A new model, PINNACLE, was developed for modelling coevolution that is consistent with the maintenance of canonical base-pairing at dinucleotide sites within alignments of DNA and RNA sequences. To achieve this, an existing model, the M95 model (Muse, 1995), was extended in four ways. The first was to allow the three different canonical base-pairs to have different rates of coevolution instead of a single shared rate. The second was to allow substitution rates to vary across sites, including allowing the two positions in a dinucleotide site each to have a potentially different rate. This was done in order account for variable substitution rates, such as those expected within a codon. The third was to permit the strength of coevolution to vary across base-paired sites, enabling the identification of sites that are neutrally coevolving versus those that actively favour canonical base-pairing. The fourth and final extension was to allow for an unknown secondary structure, achieved by using an RNA SCFG prior over secondary structures and using a Metropolis-within-Gibbs procedure to integrate out the secondary structure during Bayesian inference.

Extending the model to permit an unknown secondary structure posed significant computational challenges. The first challenge was the need to compute likelihoods using Felsentein's algorithm for all $\binom{L}{2}$ pairs of dinucleotide columns. Fortunately, a large number of redundant calculations could be avoided due to a large proportion of column pairs sharing the same patterns of dinucleotides (Pond and Muse, 2004), resulting in at least a 5× speed-up. Additionally, a further 50× speed-up was achieved using a GPU implementation of Felsenstein's algorithm. The second bottleneck was the need to marginalise an unknown secondary structure using the inside algorithm. Computational speed-ups by factors of 50× - 200× were achieved using a GPU implementation of the inside algorithm. For Bayesian inference a Metropolis-within-Gibbs procedure was implemented to further reduce the number of times the dinucleotide matrix and inside calculations were required during inference.

Both maximum likelihood (ML) and Bayesian inference were used to perform inference for different analyses. Maximum likelihood inference allowed us to perform

likelihood ratio tests of various hypotheses, where Bayes factors were computationally intractable to compute using Bayesian inference. Bayesian inference was used to obtain posterior distributions over various parameters, including the rates of coevolution associated with the canonical base-pairs, and posterior probabilities and degrees of coevolution at base-pair sites.

To perform an initial validation of our model, column permutations of alignment datasets were performed to disrupt many of the secondary structure dependencies expected in real datasets. Consistent with the model behaving desirably, the structure-integrated maximum likelihoods values were lower for the permuted datasets, and the structure information entropy values higher for the permuted datasets overall.

The ability to marginalise an unknown secondary structure shared amongst an alignment of sequences, implies that PINNACLE is also capable of secondary structure prediction. Although PINNACLE was not designed with structure prediction in mind, nor was it tuned to achieve good accuracy on benchmarks of secondary structure prediction, we found that it performed similarly to two popular comparative secondary structure prediction methods: RNAalifold (Hofacker, 2009) and PPfold (Sükösd *et al.*, 2012). This result served to further validate our model.

Strong evidence was found for GU dinucleotides being selectively favoured at base-paired sites in five non-coding RNA datasets and four of five RNA virus genome datasets, with only one showing no evidence. Whereas strong evidence for GT dinucleotides being selectively favoured at base-paired sites in DNA virus genome datasets was only found for one DNA virus dataset tested, weak evidence for one other dataset, and no evidence for the remaining three DNA virus datasets. The overall pattern of GU dinucleotides playing a role in RNA molecules is consistent with numerous phylogenetic, and experimental analyses of RNA molecules (Woese *et al.*, 1980; Eddy and Durbin, 1994; Deigan *et al.*, 2009). The role of GT base-pairings in single-stranded DNA viruses remains unclear.

Finally, the model was applied to the HIV-1 NL4-3 secondary structure and two corresponding alignment datasets containing large numbers of HIV-1 sequences, and

an SIVmac239 secondary structure and a corresponding alignment of SIV sequences. It was found that correlations between the SHAPE-determined quantities and degrees of coevolution as detected using PINNACLE were stronger than correlations detected when comparing the determined quantities and degrees of covariation, as measured using three non-evolutionary methods. Interactive visualisations of the HIV-1 NL4-3 secondary structure with the inferred degrees of coevolution overlaid were automatically generated by PINNACLE. Furthermore, ranking substructures within the HIV-1 NL4-3 secondary structure by the inferred degrees of coevolution within them demonstrated an excess of substructures amongst the top that were also commonly cited in the literature as having structure-related importance.

The ability to infer degrees of coevolution in a statistically rigorous manner, together with PINNACLE's visualisation and ranking features is expected to be useful in assisting researchers in focusing their experimental analyses on those portions of large RNA structures that are most likely to be biologically relevant.

2.5 Future challenges

2.5.1 Dual codon and RNA constraints

A feature that was not fully accounted for in our model and that is especially important for viral genomes, such as HIV, is that their genomes simultaneously encode for proteins. This implies a dual evolutionary constraint, whereby selection may be acting on the amino acid sequence, whilst simultaneously acting to maintain base-pairing interactions in biologically functional RNA secondary structures.

Clearly, there is a degree of flexibility in which selection can simultaneously operate on both, due to the redundancy of the genetic code and the multiple possible canonical RNA base-pairing combinations. A complete evolutionary model of this dual codon-RNA constraint has not yet been developed. Presumably, this is because combining a codon and dinucleotide model with RNA constraints create dependencies amongst many positions in the sequence, which implies that the independence of dinucleotide or codon sites no longer hold. This has the consequence of increasing the size of the state space from 16 dinucleotides in the case of a dinucleotide model, or 61 codons in the case of a codon model to a state space of size 4^L, where each states represents a possible sequence of length L.

Such a model could be conceptualised as a continuous-time Bayesian network (CTBN, Nodelman *et al.* (2002)), which takes advantage of the sparsity of the rate matrix (by assuming no more than one nucleotide can change in an instant). For example, from any possible nucleotide sequence (state) there are at most $3L$ changes that have non-zero rates. Inference is challenging, because using matrix exponentiation to calculate transition probabilities is no longer tractable; however, one way to perform inference is by augmenting each branch of the tree with an evolutionary trajectory that is explicitly sampled (Yu and Thorne, 2006). It is possible to sample trajectories in a CTBN using a recently developed end-point conditioned sampling algorithm outlined in Rao and Teh (2013). A possible RNA codon model is outlined here:

$$R_{uv} = \begin{cases} 0 & \text{if sequences } u \text{ or } v \text{ contain a stop codon} \\ \alpha^c Q^i_{u(i),v(i)} & \text{if } \delta(u,v) = 1,\ AA(u,c) = AA(v,c),\ i \text{ is unpaired} \\ \beta^c Q^i_{u(i),v(i)} & \text{if } \delta(u,v) = 1,\ AA(u,c) \neq AA(v,c),\ i \text{ is unpaired} \\ \alpha^c M^b_{u(d),v(d)} & \text{if } \delta(u,v) = 1,\ AA(u,c) = AA(v,c),\ i \text{ is base-paired to } i' \\ \beta^c M^b_{u(b),v(b)} & \text{if } \delta(u,v) = 1,\ AA(u,i) \neq AA(c,c),\ i \text{ is base-paired to } i' \\ 0 & \text{if } \delta(u,v) > 1 \\ \sum\limits_{k \neq u} - R_{uk} & \text{if } \delta(u,v) = 0 \end{cases} \tag{2.47}$$

Where $\delta(u,v)$ denotes the number of nucleotide differences between sequences u and v, and i is the position at which u and v differ when $\delta(u,v) = 1$, $b = (i,i')$ specifies the base-paired positions when i is paired, c specifies a codon position, and $AA(u,c)$ specifies the amino acid translation of the codon at position c in sequence u. α^c is a site specific synonymous substitution rate, β^c is a site-specific non-synonymous substitution rate, Q^i is a site specific 4×4 GTR nucleotide model, and M^d is a site specific 16×16 Muse dinucleotide model. Note that the superscripts in α^c,β^c, Q^i, and M^b denote site specific rates. It is not known whether the stationary distribution of the above model exists and has a tractable closed form solution, therefore it is also not clear whether time-reversibility holds.

2.5.2 Evolution on secondary structure

A second limiting factor in our model is the assumption of a canonical RNA secondary structure shared across the entire evolutionary history of the sequences being analysed. This is considered a reasonable approximation for low and moderately diverged alignments, where many of the sequences are expected share a high proportion of the same base-pairs.

Notwithstanding, it is also likely that different parts of the tree relating the sequences will have at least some parts of those sequence adopting alternative secondary structure conformations. These regions are interesting from a functional perspective. The ability to identify these conformations and the mutations responsible for them may lead to significant insights into biological mechanisms relating to drug resistance or drug susceptibility in viruses such as HIV.

Holmes (2004) details a substantial attempt to model the evolution of RNA secondary structure, whilst simultaneously accounting for alignment uncertainty using the TKF91 model of insertion and deletion. Whilst this model does assume a canonical secondary structure, the structures of sequences on the tree may differ from the canonical structure via the deletion of base-pairs or the insertion of unpaired nucleotides compared to the canonical structure. A drawback of this approach is that it only explores a limited fraction of the full range of possible conformations that extant and ancestral sequences may adopt; however, a model considering all conformations would likely require advanced inference procedures, combining branch augmentation of sequences and secondary structures.

Chapter 3

A generative angular model of protein structure evolution

3.1 Introduction

Recently, several studies (Challis and Schmidler, 2012; Herman *et al.*, 2014) have proposed joint stochastic models of evolution which take into account simultaneous alignment of protein sequence and structure. These studies point out the limitations of earlier non-probabilistic methods, which often rely on heuristic procedures to infer parameters of interest. A major disadvantage of using heuristic procedures is that they typically fail to account for sources of uncertainty. For example, relying on a single fixed alignment, which is highly unlikely to be the *true* underlying alignment, may bias the inference of the posterior distribution over evolutionary trees.

We present a generative evolutionary model, ETDBN (Evolutionary Torus Dynamic Bayesian Network), for pairs of homologous proteins. ETDBN captures dependencies between sequence and structure evolution, accounts for alignment uncertainty, and models the local dependencies between aligned sites.

A key step in modelling protein structure evolution is selecting a suitable structural representation and corresponding evolutionary model. Early works by Gutin and Badretdinov (1994) and Grishin (1997) represented protein structure using three-dimensional Cartesian coordinates of protein backbone atoms and used diffusions processes to model the relationship between structural distance (measured using RMSD) and sequence similarity. More recent publications by Challis and Schmidler (2012) and Herman *et al.* (2014) likewise used the three-dimensional Cartesian coordinates of amino acid C_α atoms to represent protein structure and additionally used Ornstein–Uhlenbeck (OU) processes to construct Bayesian probabilistic models of protein structure evolution. These models emphasise estimation of evolutionary parameters such as the evolutionary time between species, tree topologies and alignment, and attempt to fully account for sources of uncertainty. For the sake of computational tractability, the aforementioned approaches treat the Cartesian coordinates associated with atoms as evolving independently of another. A non-probabilistic approach by Echave (2008) and Echave and Fernández (2010) referred to as the Linearly Forced Elastic Network Model (LFENM) treats protein structures as a collection of C_α atoms con-

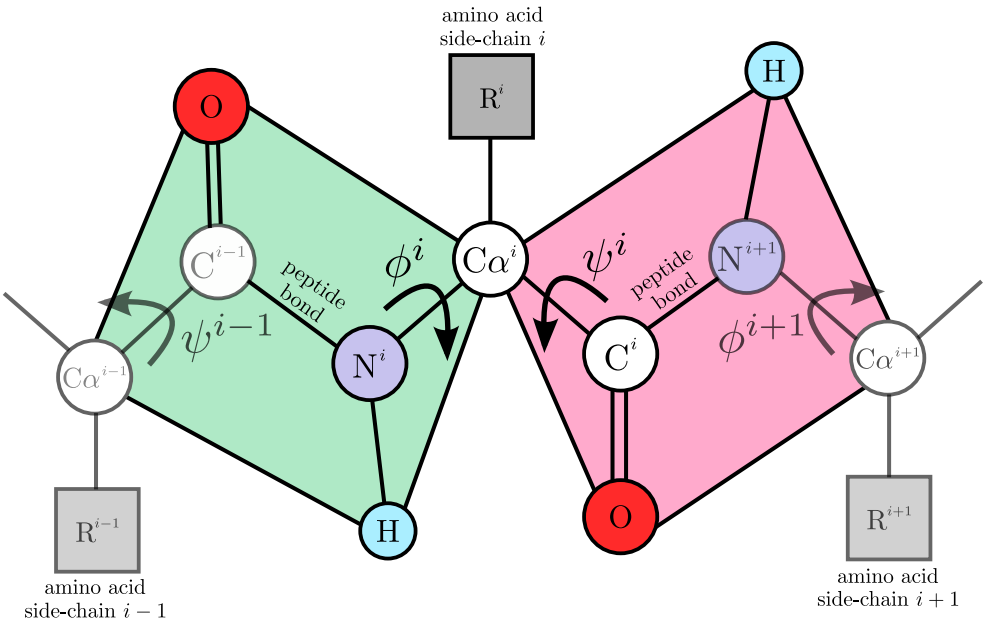


Figure 3.1: Dihedral angle representation for a small region of the protein backbone (three amino acids).

nected by spring forces. The major benefit of LFENMs is that they do not assume independence of atomic coordinates and take into account non-local structural dependencies. In their current formulation LFENMs do not capture the dependencies between amino acid sequence and structure and therefore do not account for the variable effect of sequence mutation on protein structure evolution.

Rather than using a Cartesian coordinate representation, our model, ETDBN, uses a dihedral angle representation motivated by the non-evolutionary TorusDBN model (Boomsma *et al.*, 2008, 2014). TorusDBN represents a single protein structure as a sequence of (ϕ, ψ) dihedral angle pairs, which are modelled using continuous bivariate angular distributions (Frellsen *et al.*, 2012). Likewise, ETDBN treats protein structure as a random walk in space, again making use of the ϕ and ψ dihedral angles (Figure 3.1).

The dihedral angle representation is informed by the chemical nature of peptide bonds. Each amino acid in a protein peptide chain is covalently bonded to the next via a peptide bond. Peptide bonds have a partial double bond nature that results in a planar configuration of atoms in space. This configuration allows the protein

backbone structure to be largely described as a series of ϕ and ψ dihedral angles that defines the relationship between the planes in three-dimensional space. Because each amino acid (except for the N and C terminus amino acids) is associated with a (ϕ,ψ) dihedral angle pair, only a sequence alignment is necessary to compare structures. On the other hand, models that use Cartesian coordinates typically need to superimpose structures in addition to requiring a sequence alignment (Herman *et al.*, 2014). Such superimposition introduces an additional source of uncertainty. A further advantage of the dihedral angle representation is that there are fewer degrees of freedom per amino acid and therefore typically fewer parameters required to model their evolution.

A novel stochastic diffusion process developed in García-Portugués *et al.* (2017) was used to model the evolution of dihedral angles. Additionally, a coupling was introduced such that an amino acid change can lead to a *jump* in dihedral angles and a change in diffusion process. This allows the model to capture changes in amino acid that are directionally coupled with changes in dihedral angle or secondary structure. As in Challis and Schmidler (2012) and Herman *et al.* (2014), the insertion and deletion (indel) evolutionary process is also modelled to account for alignment uncertainty (Thorne *et al.*, 1992).

The Ornstein–Uhlenbeck processes used in Challis and Schmidler (2012) and Herman *et al.* (2014) ignore bond lengths and treat C_α atoms as evolving independently for the sake of computationally tractability. Furthermore, the OU process makes Gaussian assumptions. From a generative perspective these properties will lead to evolved proteins with C_α atoms that are unnaturally dispersed in space. Bond lengths are also ignored in ETDBN, but can be plausibly fixed, or modelled. As a result, it is expected that the use of angular diffusions will more naturally capture the underlying local protein structure manifold; notwithstanding, global structural constraints are not imposed which may also lead to simulated proteins that lack certain global properties of protein structure, such as protein compactness.

Tree-like dependencies typically arise when considering two or more homologous proteins related via a common ancestor. These dependencies manifest themselves

most noticeably in the degree of amino acid sequence similarity between the homologous proteins. The strength of these dependencies is assumed to be a result of two major factors: the time since the common ancestor and the rate of evolution.

Most models of structural evolution ignore dependencies amongst sites because of the increased computational demand and model complexity associated with such models. These dependencies are expected to influence patterns of evolution, specifically patterns of amino acid substitution. The current model deals with local dependencies only – dependencies that are expected to arise due to interactions between neighbouring amino acids, for example, between amino acids in an α-helix. ETDBN does not account for global dependencies – dependencies that result in the globular nature of proteins (Boomsma *et al.*, 2008). In ETDBN, we attempt to model local dependencies only by using a Hidden Markov Model (HMM) to capture dependencies amongst neighbouring aligned positions. HMMs such as PASSML (Liò *et al.*, 1998) have been successfully used to predict protein secondary structure from aligned sequences; however, these models typically have the disadvantage that they assume a canonical secondary structure shared amongst all the sequences being analysed. This restricts analysis to closely related sequences where conservation of secondary structure is a reasonable assumption. ETDBN does not assume a canonical secondary structure, but instead uses a phylogenetic HMM approach (similar to Siepel and Haussler (2004)) that assumes dependencies between evolutionary processes at neighbouring aligned positions.

Parameters of ETDBN were estimated using 1200 homologous protein pairs from the HOMSTRAD database (Mizuguchi *et al.*, 1998). The resulting model provides a realistic prior distribution over proteins and protein structure evolution in comparison to previous stochastic models. Doing so enables biological insights into the relationship between sequence and structure evolution, such as patterns of amino acid change that are informative of patterns of structural change (Grishin, 2001). It was with these features in mind that ETDBN was developed.

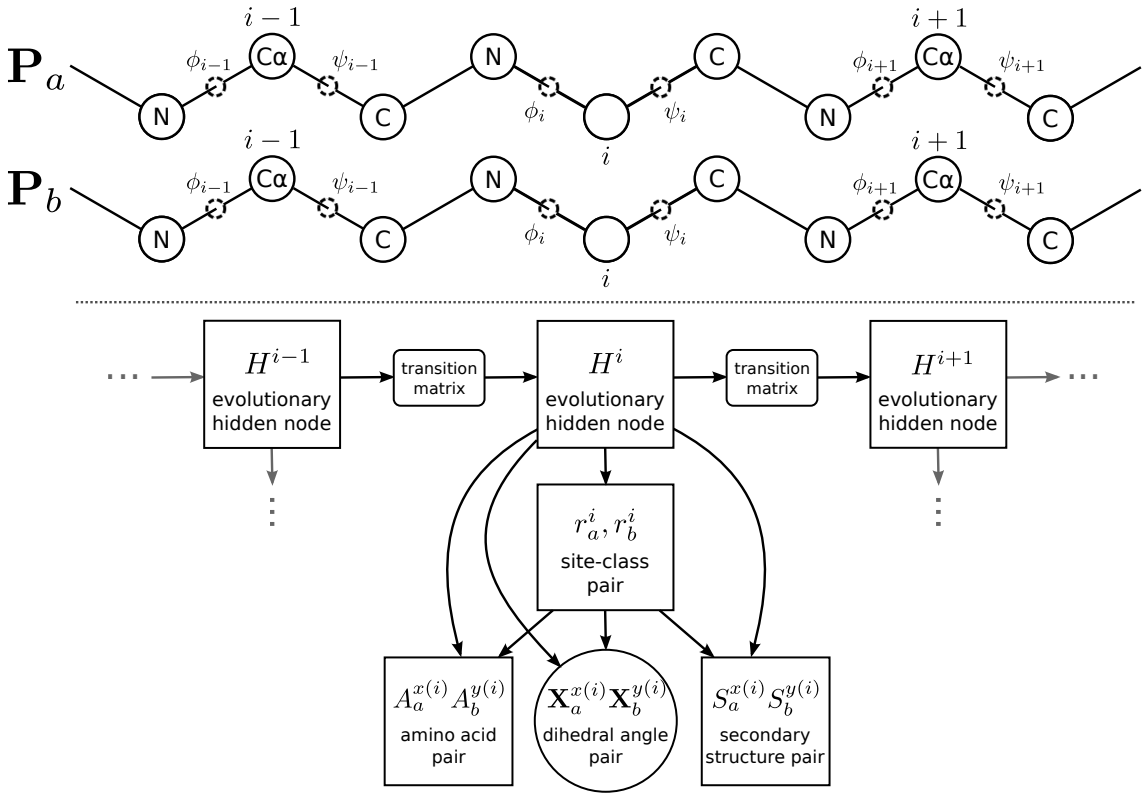


Figure 3.2: Above: dihedral angle representations of two aligned proteins ($\mathbf{P}_a$ and $\mathbf{P}_b$). A small section of two aligned protein backbones (three amino acids each) with ϕ and ψ dihedral angles are shown, together with C_α atoms which attach to the amino acid side-chains. Each amino acid side-chain determines the characteristic nature of each amino acid. Every aligned amino acid position corresponds to a hidden node in the HMM below.
Below: depiction of HMM architecture of ETDBN where each H along the horizontal axis represents an evolutionary hidden node. Horizontal edges between evolutionary hidden nodes encode neighbouring dependencies between aligned sites. Arrows between the evolutionary hidden nodes and site-class pair nodes encode the conditional independence between the observation pair variables $A_a^{x_i}, A_b^{y_i}$ (amino acid site pair), $\mathbf{X}_a^{x_i} = \langle \phi_a^{x_i}, \psi_a^{x_i} \rangle, \mathbf{X}_b^{y_i} = \langle \phi_b^{y_i}, \psi_b^{y_i} \rangle$ (dihedral angle site pair) and $S_a^{x_i}, S_b^{y_i}$ (secondary structure class site pair). Circles represent continuous variables and rectangles represent discrete variables.

3.2 Methods

3.2.1 Overview of evolutionary model

ETDBN is a dynamic Bayesian network model of local protein sequence and structure evolution along a pair of aligned homologous proteins $\mathbf{P}_a$ and $\mathbf{P}_b$. ETDBN can be viewed as an HMM (see Figure 3.2). Each hidden node of the HMM, corresponding to an aligned position, adopts an evolutionary hidden state specifying a distribution over three different observations pairs: a pair of amino acid characters, a pair of dihedral angles, and a pair of secondary structures classifications. A transition probability matrix specifies neighbouring dependencies between adjacent evolutionary states. For example, transitions along the alignment between hidden states encoding predominantly α-helix evolution are expected to occur more frequently than transitions between an evolutionary hidden state encoding predominantly α-helix evolution and another encoding predominantly β-sheet evolution.

Ideally, the hidden states would vary across the length of the alignment as captured by the HMM in the current model and evolve along the branches of the phylogenetic tree. This remains computationally intractable at present. Allowing the hidden states to evolve along the tree would permit large structural changes to occur, even those induced by a single mutation. At present we model such events using a jump model (see below).

Partially to mitigate this, each hidden state specifies a distribution over a pair of site-classes at each aligned position. This gives rise to the possibility of a 'jump event'. A jump event allows a large change in dihedral angle or secondary structure (e.g. helix to sheet) to occur at a given aligned position and also introduces a directional coupling between changes in amino acid that are informative of changes in dihedral angle or secondary structure conformation.

3.2.2 Observation types

The two proteins, $\mathbf{P}_a$ and $\mathbf{P}_b$, in a homologous pair are obtained from experimental data. An ith site observation pair, $\mathbf{P}^i = (\mathbf{P}_a^{x(i)}, \mathbf{P}_b^{y(i)})$, is associated with every aligned site i in an alignment $\mathbf{M}_{ab}$ of $\mathbf{P}_a$ and $\mathbf{P}_b$, where $\mathbf{M}_{ab}^i \in \left\{ \binom{x(i)}{y(i)}, \binom{x(i)}{-}, \binom{-}{y(i)} \right\}$ specifies the homology relationship at position i of the alignment (homologous, deletion with respect to $\mathbf{P}_a$ and insertion with respect to $\mathbf{P}_a$, respectively.), i is taken to run from 1 to m, m is the length of the alignment $\mathbf{M}_{ab}$, and $x(i) \in \{1, \ldots, |\mathbf{P}_a|\}$ and $y(i) \in \{1, \ldots, |\mathbf{P}_b|\}$ specify the indices of the positions in $\mathbf{P}_a$ and $\mathbf{P}_b$, respectively. $|\mathbf{P}_a|$ and $|\mathbf{P}_a|$ specify the number of sites in $\mathbf{P}_a$ and $\mathbf{P}_b$, respectively.

Each site observation, $\mathbf{P}_a^{x(i)}$ and $\mathbf{P}_b^{y(i)}$, contains amino acid and structural information corresponding to the two C_α atoms at aligned site i belonging to each of the two proteins. A site observation corresponding to a particular protein at aligned site i, $\mathbf{P}_a^{x(i)}$, is composed of three different data types associated with the C_α atom: an amino acid ($A_a^{x(i)}$; discrete, one of twenty canonical amino acids), ϕ and ψ dihedral angles ($\mathbf{X}_a^{x(i)} = \langle \phi_a^{x(i)}, \psi_a^{x(i)} \rangle$; continuous, bivariate), and a secondary structure classification ($S_a^{x(i)}$; discrete, one of three classes: helix (H), sheet (S), or coil (C)). Therefore, $\mathbf{P}_a^{x(i)} = (A_a^{x(i)}, \mathbf{X}_a^{x(i)}, S_a^{x(i)})$ and $\mathbf{P}_b^{y(i)} = (A_b^{y(i)}, \mathbf{X}_b^{y(i)}, S_b^{y(i)})$.

3.2.3 Model structure

The sequence of hidden nodes in the HMM is written as $\mathbf{H} = (H^1, H^2, \ldots, H^m)$. Each hidden node H^i in the HMM corresponds to a site observation pair, $\mathbf{P}_a^{x(i)}$ and $\mathbf{P}_b^{y(i)}$, at an aligned site i in the alignment $\mathbf{M}_{ab}$. Initially we treat the alignment $\mathbf{M}_{ab}$ as given *a priori*, but later modify the HMM to marginalise out an unobserved alignment.

The model is parametrised by q hidden states. Every hidden node H^i corresponding to an aligned site i takes an integer value from 1 to q for the hidden state at node H^i. In turn, each hidden state specifies a distribution over a site-class pair: (r_a^i, r_b^i) as a function of evolutionary time. A site-class pair consists of two site-classes: r_a^i and r_b^i. Each of the two site-classes takes an integer value 1 or 2, *i.e.*

$(r_a^i, r_b^i) \in \{(1,1),(1,2),(2,1),(2,2)\}$. We return to the specific role of the site-classes pairs in the next section.

The state of H^i together with the site-class pair, (r_a^i, r_b^i), and the evolutionary time separating proteins $\mathbf{P}_a$ and $\mathbf{P}_b$, t_{ab}, specify a distribution over three conditionally independent stochastic processes describing each of the three types of site observation pairs: $(A_a^{x(i)}, A_b^{y(i)})$, $(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)})$ and $(S_a^{x(i)}, S_b^{y(i)})$. This conditional independence structure allows the likelihood of a site observation pair at an aligned site i to be written as follows:

$$\begin{aligned} p(\mathbf{P}_a^{x(i)}, \mathbf{P}_b^{y(i)} | H^i, r_a^i, r_b^i, t_{ab}) = & \overbrace{p(A_a^{x(i)}, A_b^{y(i)} | H^i, r_a^i, r_b^i, t_{ab})}^{\text{amino acid evolution}} \\ & \times \overbrace{p(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)} | H^i, r_a^i, r_b^i, t_{ab})}^{\text{dihedral angle evolution}} \\ & \times \overbrace{p(S_a^{x(i)}, S_b^{y(i)} | H^i, r_a^i, r_b^i, t_{ab})}^{\text{secondary structure evolution}}. \end{aligned} \tag{3.1}$$

The assumption of conditional independence provides computational tractability, allowing us to avoid costly marginalisation when certain combinations of data are missing (e.g. amino acid sequences present, but secondary structures and dihedral angles missing).

3.2.4 Stochastic processes: modelling evolutionary dependencies

Each site-class couples together three time-reversible stochastic processes that separately describe the evolution of the three pairs of observation types, as in equation (3.1). Each site-class is intended to capture both physical and evolutionary features pertaining to sequence and structure. Parameters that correspond to a particular site-class are termed *site-class specific*, whereas parameters that are shared across all site-classes are termed *global*. The use of site-class specific parameters, such as site-class specific amino acid frequencies and dihedral angle diffusion parameters, as described

in the next section, is intended to model site-specific physical-chemical properties (Koshi and Goldstein, 1998; Halpern and Bruno, 1998; Lartillot and Philippe, 2004), such as a preference for being α-helical conformation at particular site combined with the associated amino acid and dihedral angle conformational preferences.

Amino acid evolution As is typical with models of sequence evolution, amino acid evolution, $p(A_a^{x(i)}, A_b^{y(i)}|H^i, r_a^i, r_b^i, t_{ab})$, is described by a Continuous-Time Markov Chain (CTMC). Each amino acid CTMC is parametrised in the following way: the exchangeability of amino acids is described by a 20×20 symmetric global exchangeability matrix $\mathbf{S}$ (190 free parameters; Whelan and Goldman (2001)), a site-class specific set of 20 amino acid equilibrium frequencies $\mathbf{\Pi}_r^h = \text{diag}\{\pi_1, \pi_2, \ldots, \pi_{20}\}$ (19 free parameters per site-class) and a site-class specific scaling factor Λ_r^h (1 free parameter per site-class). Note that h specifies a particular hidden state. Together these parameters define a site-class specific time-reversible amino acid rate matrix $\mathbf{Q}_r^h = \Lambda_r^h \mathbf{S}\mathbf{\Pi}_r^h$. The stationary distribution of $\mathbf{Q}_r^h$ is given by the amino acid equilibrium frequencies: $\mathbf{\Pi}_r^h$.

Secondary structure evolution Secondary structure evolution, $p(S_a^{x(i)}, S_b^{y(i)}|H^i, r_a^i, r_b^i, t_{ab})$, is also described by a CTMC. For the sake of simplicity we use only three discrete classes to describe secondary structure at each position: helix (H), sheet (S), and random coil (C).

The exchangeability of secondary structure classes at a position is described by a 3×3 symmetric global exchangeability matrix $\mathbf{V}$ and a site-class specific set of 3 secondary structure equilibrium frequencies $\mathbf{\Omega}_r^h = \text{diag}\{\pi_1, \pi_2, \pi_3\}$. Together they define a site-class specific time-reversible secondary structure rate matrix $\mathbf{R}_r^h = \mathbf{V}\mathbf{\Omega}_r^h$, with stationary distribution: $\mathbf{\Omega}_r^h$.

Dihedral angle evolution Central to our model is evolutionary dependence between dihedral angles, $p(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)}|H^i, r_a^i, r_b^i, t_{ab})$. Typically, the continuous-time evolution of the continuous-state random variables is modelled by a diffusive process

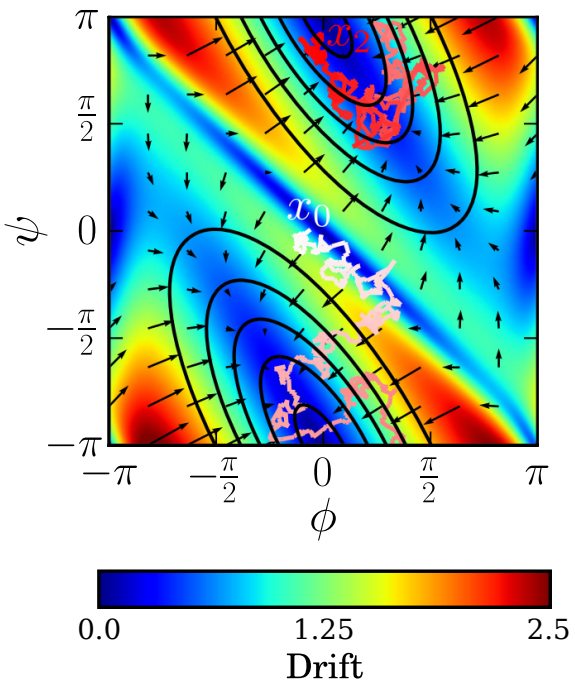


Figure 3.3: Drift vector field for the WN diffusion with $A = (1, 0.5; 0.5, 0.5)$, $\mu = (0, 0)$ and $\Sigma = (1.5)^2 I$. The colour gradient represents the Euclidean norm of the drift. The contour lines represent the stationary distribution. An example trajectory starting at $x_0 = (0, 0)$ and ending at x_2, running in the time interval $[0, 2]$ is depicted using a white to red colour gradient indicating the progression of time. The periodic nature of the diffusion can be seen by the wrapping of both the stationary diffusion and the trajectory at the boundaries of the square plane. The fact that stationary distribution is not aligned with the horizontal and vertical axes illustrates the dependence (given by α_3) between the ϕ and ψ dihedral angles.

such as the Ornstein-Uhlenbeck (OU) process, as in Challis and Schmidler (2012). However, an OU process is not appropriate for dihedral angles as they have a natural periodicity. For this reason, a bivariate diffusion that captures the periodic nature of dihedral angles, the *Wrapped Normal* (WN) *diffusion*, was specifically developed for this paper in García-Portugués *et al.* (2017). Please note that the following description of the WN diffusion was adapted from text largely written by Eduardo García-Portugués in Golden *et al.* (2017), but was adapted and included in this dissertation for completeness.

Topologically, the WN diffusion (see Figure 3.3 for a pictorial example) can be thought of as the analogue of the OU process on the torus $\mathbb{T}^2 = [-\pi, \pi) \times [-\pi, \pi)$.

The WN diffusion arises as the wrapping on $\mathbb{T}^2$ of the following Euclidean diffusion:

$$d\mathbf{X}_t = \overbrace{\mathbf{A}\sum_{\mathbf{k}\in Z^2}(\boldsymbol{\mu} - \mathbf{X}_t - 2\mathbf{k}\pi)w_{\mathbf{k}}(\mathbf{X}_t)}^{\substack{\text{drift}\\ \text{coefficient}}}\, dt \; + \; \overbrace{\boldsymbol{\Sigma}^{\frac{1}{2}}}^{\substack{\text{diffusion}\\ \text{coefficient}}}\, d\mathbf{W}_t, \tag{3.2}$$

where $\mathbf{W}_t$ is the two-dimensional Wiener process, $\mathbf{A}$ is the drift matrix, $\boldsymbol{\mu} \in \mathbb{T}^2$ is the stationary mean, $\boldsymbol{\Sigma}$ is the infinitesimal covariance matrix and

$$w_{\mathbf{k}}(\boldsymbol{\theta}) = \frac{\phi_{\frac{1}{2}\mathbf{A}^{-1}\boldsymbol{\Sigma}}(\boldsymbol{\theta} - \boldsymbol{\mu} + 2\mathbf{k}\pi)}{\sum_{\mathbf{m}\in\mathbb{Z}^2}\phi_{\frac{1}{2}\mathbf{A}^{-1}\boldsymbol{\Sigma}}(\boldsymbol{\theta} - \boldsymbol{\mu} + 2\mathbf{m}\pi)}, \tag{3.3}$$

$\boldsymbol{\theta} \in \mathbb{T}^2$, is a probability density function (pdf) for $\mathbf{k} \in \mathbb{Z}^2$. $\phi_{\boldsymbol{\Sigma}}$ stands for the pdf of a bivariate Gaussian $\mathcal{N}(0, \boldsymbol{\Sigma})$. The pdf (3.3) weights the linear drifts of (3.2) such that they become smooth and periodic.

It is shown in García-Portugués *et al.* (2017) that the stationary distribution of the WN diffusion is a WN$(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which has pdf:

$$p_{\mathrm{WN}}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{\mathbf{k}\in\mathbb{Z}^2}\phi_{\boldsymbol{\Sigma}}(\boldsymbol{\theta} - \boldsymbol{\mu} + 2\mathbf{k}\pi). \tag{3.4}$$

Despite involving an infinite sum over $\mathbb{Z}^2$, taking just the first few terms of this sum provides a tractable, and accurate approximation to the stationary density for most of the realistic parameter values.

Maximum Likelihood Estimation (MLE) for diffusions is based on the transition probability density (tpd), which only has a tractable analytical form for very few specific processes. A highly tractable and accurate approximation to the tpd is given for the WN diffusion. This approximation results from weighting the tpd of the OU process in the same fashion as the linear drifts are weighted in (3.2), yielding the following multimodal pseudo-tpd:

$$\tilde{p}(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, t) = \sum_{\mathbf{m}\in\mathbb{Z}^2} p_{\mathrm{WN}}(\boldsymbol{\theta}_2|\boldsymbol{\mu}_t^{\mathbf{m}}, \Gamma_t)w_{\mathbf{m}}(\boldsymbol{\theta}_1), \tag{3.5}$$

with $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{T}^2$, $\boldsymbol{\mu}_t^{\mathbf{m}} = \boldsymbol{\mu} + e^{-t\mathbf{A}}(\boldsymbol{\theta}_1 - \boldsymbol{\mu} + 2\pi\mathbf{m})$ and $\Gamma_t = \int_0^t e^{-s\mathbf{A}}\boldsymbol{\Sigma}e^{-s\mathbf{A}^T}ds$. The pseudo-tpd provides a good approximation to the true tpd in key circumstances: *i*) $t \to 0$, since it collapses in the Dirac's delta; *ii*) $t \to \infty$, since it converges to the stationary distribution; *iii*) high concentration, since the WN diffusion becomes an OU process. Furthermore, it is shown in García-Portugués *et al.* (2017) that the pseudo-tpd has a lower Kullback-Leibler divergence with respect to the true tpd than the Euler and Shoji-Ozaki pseudo-tpds, for most typical scenarios and discretisation times in the diffusion trajectory.

A further desirable property of the pseudo-tpd is that it obeys the time-reversibility equation, which in terms of $(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)})$ is

$$\begin{aligned}
&\tilde{p}(\mathbf{X}_b^{y(i)}|\mathbf{X}_a^{x(i)}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, t_{ab})p_{\mathrm{WN}}(\mathbf{X}_a^{x(i)}|\boldsymbol{\mu}, \tfrac{1}{2}\mathbf{A}^{-1}\boldsymbol{\Sigma}) \\
&\quad = \tilde{p}(\mathbf{X}_a^{x(i)}|\mathbf{X}_b^{y(i)}; \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, t_{ab})p_{\mathrm{WN}}(\mathbf{X}_b^{y(i)}|\boldsymbol{\mu}, \tfrac{1}{2}\mathbf{A}^{-1}\boldsymbol{\Sigma}).
\end{aligned}$$

Indeed, the WN diffusion is the *unique* time-reversible diffusion with the stationary pdf (3.4), in the same way the OU is with respect to a Gaussian. Time-reversibility is an assumption of the overall model and many other models of sequence evolution. A benefit of time-reversibility in a pairwise model such as ETDBN is that one of the proteins in a pair may be arbitrarily chosen as the ancestor, thus avoiding a computationally expensive marginalisation of an unobserved ancestor.

The probability of a dihedral angle observation pair $(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)})$, assuming that $\mathbf{X}_a^{x(i)}$ is drawn from the stationary distribution, is given by:

$$\begin{aligned}
&p(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)}|H^i, r_a^i, r_b^i, t_{ab}) \\
&\quad = p(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)}|\mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, t_{ab}) \\
&\quad \approx \tilde{p}(\mathbf{X}_b^{y(i)}|\mathbf{X}_a^{x(i)}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, t_{ab})p_{\mathrm{WN}}(\mathbf{X}_a^{x(i)}|\boldsymbol{\mu}, \tfrac{1}{2}\mathbf{A}^{-1}\boldsymbol{\Sigma}),
\end{aligned} \tag{3.6}$$

$\mathbf{A}$ and $\boldsymbol{\Sigma}$ are constrained to yield a covariance matrix $\mathbf{A}^{-1}\boldsymbol{\Sigma}$. A parametrization that achieves this is $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \sigma_2^2)$ and $\mathbf{A} = \left(\alpha_1, \frac{\sigma_1}{\sigma_2}\alpha_3; \frac{\sigma_2}{\sigma_1}\alpha_3, \alpha_2\right)$, $\alpha_1\alpha_2 > \alpha_3^2$. α_1 and α_2 are the drift components for the ϕ and ψ dihedral angles, respectively. α_3 captures

the dependence (correlation) between the dihedral angles. Depicted in Figure 3.3 is a WN diffusion with given drift and diffusion parameters.

3.2.5 Site-classes: constant evolution and jump events

We now turn to the meaning of the site-class pairs. Two modes of evolution are modelled: *constant evolution* and *jump events*. Constant evolution occurs when the site-class starting in protein $\mathbf{P}_a$ at aligned site i, r_a^i, is the same as the site-class ending in protein $\mathbf{P}_b$ at aligned site i, r_b^i, *i.e.* $r_a^i = r_b^i$. When this is the case the distribution over observation pairs at a site is specified by a single site-class.

As already stated, a site-class specifies the parameters of the three conditionally independent stochastic processes describing amino acid, dihedral angle, and secondary structure evolution. A limitation of "constant evolution" is that the coupling between the three stochastic processes is somewhat weak. This in part stems from the time-reversibility of the stochastic processes – swapping the order of one of the three observation pairs at a homologous site, e.g. (Glycine, Proline) instead of (Proline, Glycine), does not alter the likelihood in equation (3.1). Alternatively restated from a generative perspective: a 'directional coupling' of an amino acid interchange does not inform the direction of change in dihedral angle or secondary structure. For example, the replacement of a glycine in an α-helix in one protein with a proline at the homologous position in a second protein is in reality expected to break the α-helix in the second protein, strongly informing the plausible dihedral angle conformations in the second protein.

Ideally, we would consider a model in which the underlying site-classes were not fixed over the evolutionary trajectory separating the two proteins, as in the case of constant evolution as described above, but instead were able to 'evolve' in time. This permits occasional switches in the underlying site-class at a particular homologous site, creating a strong dependency between amino acid, dihedral angle, and secondary structure evolution that captures the directional coupling we desire. Such an approach is computationally demanding due to the introduction of context-dependence when

having to consider neighbouring dependencies amongst evolutionary trajectories at adjacent sites. However, if the neighbouring dependencies between aligned sites were ignored the model would remain tractable.

To approximate this 'ideal' model in a computationally efficient manner we introduce the notion of a *jump event*. A jump event occurs when $r_a^i \neq r_b^i$. Whereas constant evolution is intended to capture angular drift (changes in dihedral angles localised to a region of the Ramachandran plot), a jump event is intended to create a directional coupling between amino acid and structure evolution, and is also expected to capture angular shift (large changes in dihedral angles, possibly between distant regions of the Ramachandran plot).

The hidden state at node H^i, together with the evolutionary time t_{ab} separating proteins $\mathbf{P}_a$ and $\mathbf{P}_b$, specifies a joint distribution over a site-class pair:

$$p(r_a^i, r_b^i | H^i, t_{ab}) = p(r_a^i | H^i, r_b^i, t_{ab}) p(r_b^i | H^i), \tag{3.7}$$

where

$$p(r_a^i | H^i, r_b^i, t_{ab}) = \begin{cases} e^{-\gamma_{H^i} t_{ab}} + \pi_{H^i, r_b^i}(1 - e^{-\gamma_{H^i} t_{ab}}), & \text{if } r_a^i = r_b^i, \\ \pi_{H^i, r_b^i}(1 - e^{-\gamma_{H^i} t_{ab}}), & \text{if } r_a^i \neq r_b^i, \end{cases}$$

and $p(r_a^i | H^i) = \pi_{H^i, r_a^i}$ and $p(r_b^i | H^i) = \pi_{H^i, r_b^i}$. Note that these probabilities are analogous to the F81 DNA model, but for two states.

π_{H^i, r_a^i} and π_{H^i, r_b^i} are model parameters specifying the probability of starting in site-class r_a^i or r_b^i, respectively, corresponding to the hidden state specified by node H^i, whereas $\gamma_{H^i} > 0$ is a model parameter specifying the jump rate corresponding to the hidden state specified by node H^i.

The site-class jump probabilities have been chosen so that time-reversibility holds,

in other words:

$$p(r_a^i|H^i, r_b^i, t_{ab})p(r_b^i|H^i) = p(r_b^i|H^i, r_a^i, t_{ab})p(r_a^i|H^i).$$

The hidden state, H^i, at node i together with a site-class pair (r_a^i, r_b^i) and the evolutionary time t_{ab}, specifies the joint likelihood over site observation pairs:

$$\begin{aligned} &p(\mathbf{P}_a^{x(i)}, \mathbf{P}_b^{y(i)}|H^i, r_a^i, r_b^i, t_{ab}) \\ &\quad = \begin{cases} p(\mathbf{P}_a^{x(i)}, \mathbf{P}_b^{y(i)}|H^i, r_c^i, t_{ab}), & \text{if } r_a^i = r_b^i = r_c^i, \\ p(\mathbf{P}_a^{x(i)}|H^i, r_a^i)p(\mathbf{P}_b^{y(i)}|H^i, r_b^i), & \text{if } r_a^i \neq r_b^i. \end{cases} \end{aligned} \tag{3.8}$$

In the case of constant evolution, evolution at aligned site i is described by the same site-class r_c^i. Evolution is considered constant because each observation type is drawn from a single stochastic process specified by H^i and r_c. Note that the strength of the evolutionary dependency within an observation pair is a function of the evolutionary time t_{ab}.

In the case of a jump event the evolutionary processes are, after the evolutionary jump, restarted independently in the stationary distribution of the new site-class. Thus the site observations, $\mathbf{P}_a^{x(i)}$ and $\mathbf{P}_b^{y(i)}$, are assumed to be drawn from the stationary distributions of two separate stochastic processes corresponding to site-classes r_a^i and r_b^i, respectively. This implies that, conditional on a jump, the likelihood of the observations is no longer dependent on t_{ab}. Stated differently: a jump event is an abstraction that captures the end-points of the evolutionary process, but ignores the unobserved evolutionary trajectory linking the two site observations. The advantage of abstracting the evolutionary trajectory is that there is no need to perform a computationally expensive marginalisation over all possible trajectories, as might be necessary in a model where the hidden states evolve along a tree. The likelihood of

an observation pair is now simply a sum over the four possible site-class pairs:

$$\begin{aligned} &p(\mathbf{P}^{x(i)}, \mathbf{P}^{y(i)} | H^i, t_{ab}) \\ &\quad = \sum_{(r_a^i, r_b^i) \in R} p(\mathbf{P}^{x(i)}, \mathbf{P}^{y(i)} | H^i, r_a^i, r_b^i, t_{ab}) p(r_a^i, r_b^i | H^i, t_{ab}), \end{aligned}$$

where $R = \{(1,1),(1,2),(2,1),(2,2)\}$ is the set of four site-class pairs, $p(\mathbf{P}^{x(i)}, \mathbf{P}^{y(i)} | H^i, r_a^i, r_b^i)$ is given by (3.8), and $p(r_a^i, r_b^i | H^i, t_{ab})$ is given by (3.7).

Posterior inference: evolutionary times The posterior distribution of t_{ab}, conditional on the observed dihedral angles $\mathbf{X}_a$ and $\mathbf{X}_b$, is given by Bayes' theorem as

$$\begin{aligned} p(t_{ab} \,|\, \mathbf{X}_a, \mathbf{X}_b) &\propto p(\mathbf{X}_a, \mathbf{X}_b \,|\, t_{ab}) p(t_{ab}) \\ &= p(t_{ab}) \sum_{\mathbf{H}} p\big(\mathbf{X}_a^{x(1)}, \mathbf{X}_b^{y(1)} \,|\, H^1, t_{ab}\big) p(H^1) \\ &\quad \times \prod_{i=2}^{m} p\big(\mathbf{X}_a^{x(i)}, \mathbf{X}_b^{y(i)} \,|\, H^i, t_{ab}\big) p(H^i \,|\, H^{i-1}). \end{aligned} \tag{3.9}$$

The prior distribution over $p(t_{ab})$ is given by an exponential distribution with $\lambda = \frac{1}{10}$, chosen for its biological plausibility. The Metropolis-Hastings algorithm can be used to sample the posterior distribution, $p(t_{ab} \,|\, \mathbf{X}_a, \mathbf{X}_b)$, using the associated prior and likelihood in (3.9), which can be efficiently computed using the HMM forward algorithm. Analogous expressions hold for $p(t_{ab} \,|\, \mathbf{A}_a, \mathbf{A}_b)$ and $p(t_{ab} \,|\, \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a, \mathbf{X}_b)$.

Posterior inference: structure ETDBN can be used to sample (*i.e.* predict) the dihedral angles of a protein b, $\mathbf{X}_b$, from $\mathbf{A}_b$, $\mathbf{A}_a$, $\mathbf{X}_a$, $\mathbf{S}_b$, $\mathbf{S}_a$, or any combination of them. We describe in the next three paragraphs the procedure for achieving this.

Conditional on having observed $(\mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a)$, the sequence of missing dihedral angles $\mathbf{X}_b$ can be sampled using the following procedure. Firstly, similar to (3.9), the likelihood $p(t_{ab} \,|\, \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a)$ and the prior $p(t_{ab})$ are used to sample a single

evolutionary time, ${t_{ab}}^{(k)}$, using Metropolis-Hastings:

$$ {t_{ab}}^{(k)} \sim p(t_{ab} \,|\, \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a) \tag{3.10} $$

When only a single observation is present for a particular observation type (e.g. only $\mathbf{X}_a$ in $p(t_{ab} \,|\, \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a)$), it is assumed that the observation is drawn from the stationary distribution of the corresponding stochastic process. Conditional on the observations and newly sampled evolutionary time, $t_{ab}^{(k)}$, the forward probabilities already calculated in (3.10) can be used to draw a sequence of hidden states, $\mathbf{H}^{(k)} \sim p\big(\mathbf{H} \,|\, \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a, t_{ab}^{(k)}\big)$, using the Forward Filtering Backward Sampling algorithm (Frühwirth-Schnatter, 1994).

For each hidden state ${H^i}^{(k)}$ in the newly sampled hidden sequence $\mathbf{H}^{(k)}$, a site-class pair $(r_a^i, r_b^i)^{(k)}$ taking on one of four possible values is sampled with probability proportional (recall an application of Bayes' theorem in the first factor) to $p\big(A_a^{x(i)}, A_b^{y(i)}, \mathbf{X}_a^{x(i)} \,|\, {H^i}^{(k)}, r_a^i, r_b^i, {t_{ab}}^{(k)}\big) p\big(r_a^i, r_b^i \,|\, {H^i}^{(k)}, {t_{ab}}^{(k)}\big)$.

If ${r_a^i}^{(k)} \neq {r_b^i}^{(k)}$, a jump is implied and a dihedral angle pair, ${\mathbf{X}_b^{y(i)}}^{(k)}$ corresponding to aligned site i is drawn from the stationary distribution of the diffusion whose parameters are specified by ${r_b^i}^{(k)}$: ${\mathbf{X}_b^{y(i)}}^{(k)} \sim p\big(\mathbf{X}_b^{y(i)} \,|\, {H^i}^{(k)}, {r_b^i}^{(k)}\big)$. If ${r_a^i}^{(k)} = {r_b^i}^{(k)}$, constant evolution is implied and a dihedral angle pair corresponding to aligned site i is drawn from the diffusion whose parameters are specified by ${r_a^i}^{(k)} = {r_b^i}^{(k)}$ using the sampling procedure obtained from (3.5): ${\mathbf{X}_b^{y(i)}}^{(k)} \sim p\big(\mathbf{X}_b^{y(i)} \,|\, {H^i}^{(k)}, {r_b^i}^{(k)}, \mathbf{X}_a^{x(i)}, {t_{ab}}^{(k)}\big)$. Doing so produces a sequence of dihedral angles, ${\mathbf{X}_b}^{(k)}$, sampled from the correct posterior distribution.

3.2.6 Pairwise statistical alignment: modelling insertions and deletions with neighbouring dependencies

Protein sequences can not only undergo point mutation events, but also insertion and deletion (indel) events. We describe a modified pairwise TKF92 alignment HMM that models both local sequence/structure evolution and sequence alignment based

on Miklós *et al.* (2004).

Whilst it is possible to fix the alignment in advance by pre-aligning the sequences using an optimisation-based alignment method (such as MAFFT (Katoh *et al.*, 2002), or MUSCLE (Edgar, 2004)) or using a curated alignment (such as from the HOMSTRAD database), doing so ignores alignment uncertainty.

An alignment can be thought of as a statement about homology, such that when amino acid positions are aligned to indicate homology they are considered to have evolved solely via mutation along the evolutionary trajectory linking them and therefore not via an indel. As the evolutionary trajectory of indels is rarely observed in practice, it is difficult to make statements about the true underlying alignment (homology relationships). This is especially true when the sequences being compared are distantly related and/or the rate indel of evolution is high. The TKF92 model (Thorne *et al.*, 1992) gives a stochastic process modelling indel evolution.

For the pairwise case, the TKF92 model can be represented as an HMM using the formulation described in Miklós et al. (2008). This HMM formulation allows one to sum over all possible pairwise alignments in $\mathcal{O}(nm)$ time using the HMM forward-backward algorithm, where n and m are the respective lengths of the two sequences. Thereby accounting for alignment uncertainty due to insertions or deletions.

We implemented a modified version of this HMM implementation (Figure 3.4) such that each emitted pair of characters is drawn from one of q evolutionary hidden states. Additionally, we encode neighbouring dependencies amongst evolutionary hidden states along the alignment, by specifying a probability transition matrix $p(H^i, H^{i+1})$, that allows the hidden states to transition at "Insertion", "Deletion" or "Match" nodes. These hidden states are intended to encode local sequence and structure evolution.

The introduction of evolutionary hidden states with neighbouring dependencies increases the computational complexity from $\mathcal{O}(nm)$ in a model without evolutionary hidden states, to $\mathcal{O}(nmq^2)$ in a model with evolutionary hidden states, where q is the number of evolutionary hidden states. The probability of an observation pair

under this model depends on the homology relationship, $\mathbf{M}^i_{ab}$, at a given position is as follows:

$$p(\mathbf{P}^i|\mathbf{M}^i_{ab}, H^i, t_{ab}) = \begin{cases} p(\mathbf{P}^{x(i)}_a, \mathbf{P}^{y(i)}_b|H^i, t_{ab}) & \text{if observations at positions } x(i) \\ & \text{and } y(i) \text{ in proteins } a \text{ and } b, \\ & \text{respectively, are homologous.} \\ p(\mathbf{P}^{x(i)}_a|H^i) & \text{if the observation at position} \\ & \text{x(i) in protein } a \text{ is the result of} \\ & \text{an indel, the likelihood is given} \\ & \text{by the stationary distribution.} \\ p(\mathbf{P}^{y(i)}_b|H^i) & \text{if the observation at position} \\ & \text{y(i) in protein } b \text{ is the result of} \\ & \text{an indel, the likelihood is given} \\ & \text{by the stationary distribution.} \end{cases} \tag{3.11}$$

Where $\mathbf{M}^i_{ab} \in \{\binom{x}{y}, \binom{x}{-}, \binom{-}{y}\}$ specifies one of three possible homology relationships at position i in the alignment (homologous amino acids, deletion with respect to protein a, and insertion with respect to protein a, respectively.). Where $x \in \{1, \ldots, |a|\}$ and $y \in \{1, \ldots, |b|\}$ specify the indices of the positions in proteins a and b, respectively. Note that a disadvantage of our approach is that neighbouring amino acid positions in the presence of a deletion or a series of deletions are not treated as directly physically adjacent to one another with respect to that particular protein. However, the original HMM formulation where we treat the alignment as given *a priori* and similar phylogenetic HMMs (Liò *et al.*, 1998; Siepel and Haussler, 2004) suffer from the same shortcoming.

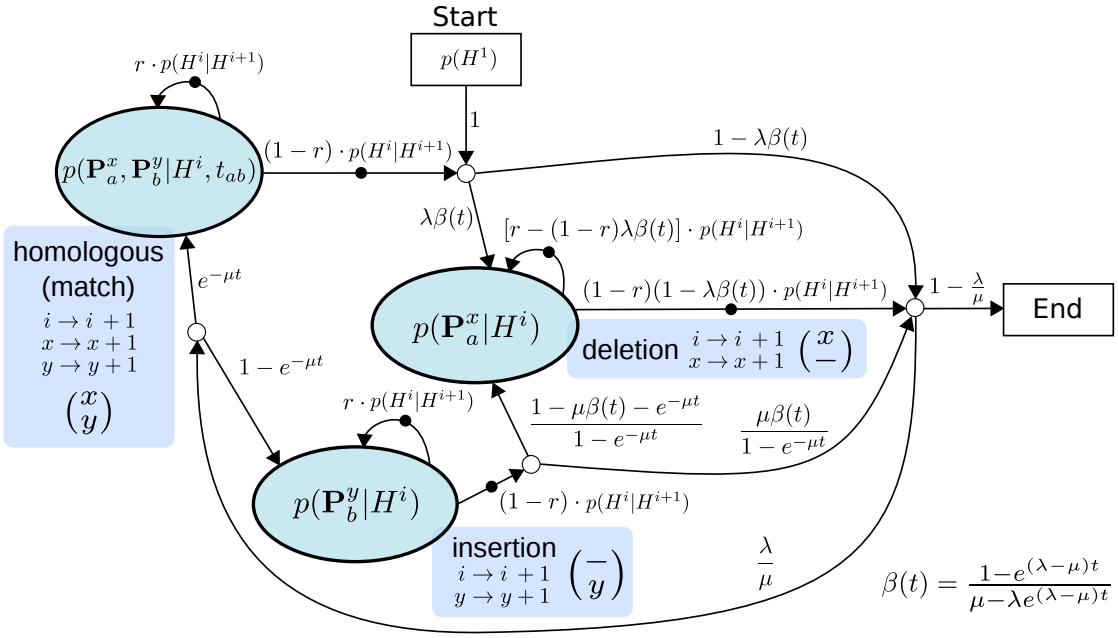


Figure 3.4: Diagrammatic representation of modified TKF92 alignment with neighbour-dependent evolutionary hidden states. The small white nodes represent non-emitting nodes. The three large oval nodes represent emitting nodes, emitting a deletion with respect to the first sequence $\binom{x}{-}$, an insertion with respect to the first sequence $\binom{-}{y}$ or a homologous pair of amino acids $\binom{x}{y}$. Edges with filled black circles indicate that the evolutionary hidden state is permitted to transition to a potentially different evolutionary hidden state, such transitions are permissible on all edges which exit one of the three emitting states. Parameters determining the transition probabilities are as follows: the insertion rate ($\lambda > 0$), the deletion rate ($\mu > 0$, the extension rate ($0 < r < 1$) and the evolutionary time t_{ab}. It is required that $\lambda < \mu$, such that the distribution ($t \to \infty$) of sequence lengths is finite at stationarity. The evolutionary hidden state transition probabilities, $P(H^i|H^{i-1})$, are given by a $q \times q$ transition probability matrix and the initial probabilities at the first alignment site, $p(H^i)$, is given by a length q probability vector.

3.2.7 Calculation of angular distances

For benchmarking and comparison purposes, the angular cosine distance was used to measure distances between pairs of dihedral angles, $\langle \phi_a, \psi_a \rangle$ and $\langle \phi_b, \psi_b \rangle$ (note that the positional superscripts $x(i)$ and $y(i)$ are not shown for ease of exposition). It is defined as follows (Downs and Mardia, 2002):

$$d(\langle \phi_a, \psi_a \rangle, \langle \phi_b, \psi_b \rangle) = \sqrt{4 - 2\cos(\phi_a - \phi_b) - 2\cos(\psi_a - \psi_b)}. \tag{3.12}$$

The maximum possible distance is $\sqrt{8} \approx 2.828$. Angular distance defined in this way has the property that when $\phi_a - \phi_b \approx 0$ and $\psi_a - \psi_b \approx 0$ are near zero it may be approximated by the Euclidean distance – using the small angle approximation for cosine ($\cos\theta \approx 1 - \frac{\theta^2}{2}$ when θ is near zero):

$$\begin{aligned} d(\langle \phi_a, \psi_a \rangle, \langle \phi_b, \psi_b \rangle) &\approx \sqrt{4 - 2(1 - (\phi_a - \phi_b)^2/2) - 2(1 - (\psi_a - \psi_b)^2/2)} \\ &= \sqrt{(\phi_a - \phi_b)^2 + (\psi_a - \psi_b)^2}. \end{aligned}$$

3.2.8 Identification of evolutionary motifs encoding jump events

To identify aligned sites having potential evolutionary motifs encoding jump events, a specific criterion was developed.

Inference was performed on each protein pair conditioned on the amino acid sequence and dihedral angles for both proteins, $(\mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a, \mathbf{X}_b)$. Homologous sites corresponding to a single hidden state and with evidence of a jump event ($r_a^i \neq r_b^i$) at posterior probability > 0.90 were identified, that is, the i's such that $p(H^i, r_a^i \neq r_b^i | \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a, \mathbf{X}_b) > 0.90$.

In a second filtering step, amino acid sequences and a single set of dihedral angles corresponding to one of the proteins were used ($\mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a$ or $\mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_b$) to infer the posterior probability, this time at a lower threshold: $p(H^i, r_a^i \neq r_b^i | \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a) > 0.50$ or $p(H^i, r_a^i \neq r_b^i | \mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_b) > 0.50$. This second criterion ensured that the evolutionary motif was identifiable under typical conditions where one has limited

access to structural information (in this case a single protein structure in a pair). Only those aligned sites meeting both criteria were selected for downstream analysis.

3.2.9 Training and test datasets

A training dataset of 1200 protein pairs (2400 proteins; 417,870 site observation pairs) and a test dataset of 38 protein pairs (76 proteins; 14125 site observation pairs) were assembled from 1032 protein families in the HOMSTRAD database.

For each protein family in HOMSTRAD (ranging in size from 2 to 22 homologous proteins each), a phylogenetic tree was inferred from the HOMSTRAD protein family sequence alignment using FastTree (Price *et al.*, 2010). Each protein family tree was taken and protein pairs selected such that the sum of the branches between pairs was maximised, whilst ensuring that no pair of proteins in the set shared an overlapping evolutionary history. This was done to maximise the amount of information, whilst minimising dependencies due to shared evolutionary history.

Dihedral angles were computed from the PDB coordinates of each protein structure using the BioPython.PDB package (Hamelryck and Manderick, 2003). Furthermore, each protein structure was obtained from the PDB database. The secondary structure was annotated at each amino acid position using DSSP (Touw *et al.*, 2015). In the case of gaps (missing residues), the fragments were treated as separate proteins without secondary structure annotation, to avoid misclassification by DSSP. In these cases, the secondary structure annotations were thus treated as latent variables. All 38 protein pairs in the test dataset were free of gaps and were from distinct protein families.

3.2.10 Model training

Stochastic EM (StEM, Gilks *et al.* (1995)) was used to train the model. StEM is a stochastic version of the well known Expectation-Maximization algorithm (Gilks *et al.*, 1995). Its distinguishing feature is that the E-step consists of filling in the values of the latent variables using sampling. Only a single value is sampled. StEM

is attractive due to its computational efficiency and its tendency to avoid getting stuck in local minima (Gilks *et al.*, 1995).

Forward Filtering Backward Sampling was used in the E-step to jointly sample alignment configurations (M_{ab}), hidden node states (H) and site-classes, (r_a, r_b). The Metropolis-Hastings algorithm was used to sample the four pair-specific continuous parameters $\theta_{ab} = \{t_{ab}, \lambda_{ab}, \mu_{ab}, r_{ab}\}$: evolutionary time ($t_{ab}$), insertion ($\lambda_{ab}$), deletion ($\mu_{ab}$), and geometric-extension (r_{ab}) rates. In other words, at iteration k for each pair of unaligned observation sequences O_a and O_b we draw samples, from the following joint-distribution:

$$Z_{ab}^{(k)} \sim p(M_{ab}, H, r_a, r_b, \theta_{ab} | O_a, O_b, \Psi^{(k)}).$$

In the M-step the samples from the previous E-step, were used to update the hidden node parameters ($\hat{\Psi}$) using efficient sufficient statistics (ESSs). Where ESSs were not available, the COBYLA optimization algorithm (Powell, 1994) in the NLOpt library (Johnson, 2014) was used to update the parameters.

3.2.11 Parallelisation of model training

The StEM algorithm is trivially parallelised. In the E-step, the parameters, alignment and hidden states can be independently sampled in parallel, conditioned on the parameters, $\Psi^{(k)}$, from the M-step. In the M-step the parameters corresponding to each hidden state can be independently updated in parallel conditioned on the samples from the E-step.

3.2.12 Model selection

Models with 8, 16, 32, 48, 52, 56, 60, 64, 68, 72, 76, 80, 96 and 112 hidden states were trained until convergence for varying numbers of repetitions (2 to 4) using different initial random number seeds. The highest log-likelihood model of each repetition was selected for downstream analysis.

Following that, marginal probabilities $p(D|\text{model})$ and corresponding Bayesian information criterion (BIC) scores were computed under each model by fixing the alignments to the respective HOMSTRAD alignments. The alignments were fixed *apriori* to make computation of the marginal likelihoods computationally tractable. Additionally, predictive accuracies under a homology modelling scenario, $p(X_b|A_a, A_b, X_a, \text{model})$, were calculated for each of the 38 protein pairs in the test dataset.

3.3 Results and Discussion

3.3.1 Model selection

Fourteen models were trained (8, 16, 32, 48, 52, 56, 60, 64, 68, 72, 76, 80, 96 and 112 hidden state models). The 64 hidden state model was chosen as the best model, as it had the lowest Bayesian Information Criterion (BIC, Figure 3.5).

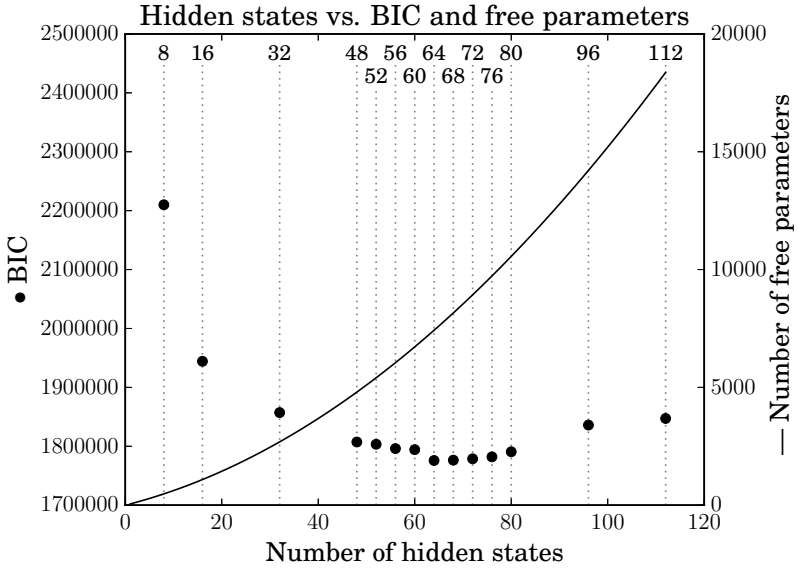


Figure 3.5: BIC scores (points) and number of free parameters (curve) as a function of the number of hidden states across 14 models (indicated above the dotted vertical lines) trained using the 1200 protein pairs in the training dataset. A 64 hidden state model had the lowest BIC score. Each model represents the best of several attempts.

3.3.2 Stationary distributions over dihedral angles capture the empirical distribution

Figure 3.6 illustrates the sampled and empirical dihedral angle distributions. There is a good correspondence between dihedral angles sampled under the model (Figure 3.6, left) and the empirical distribution of dihedral angles in our training dataset (Figure 3.6, right) for all three cases illustrated (all amino acids, glycine only and proline only). The correspondence is not surprising given that ETDBN is effectively a mixture model with a large number of mixture components.

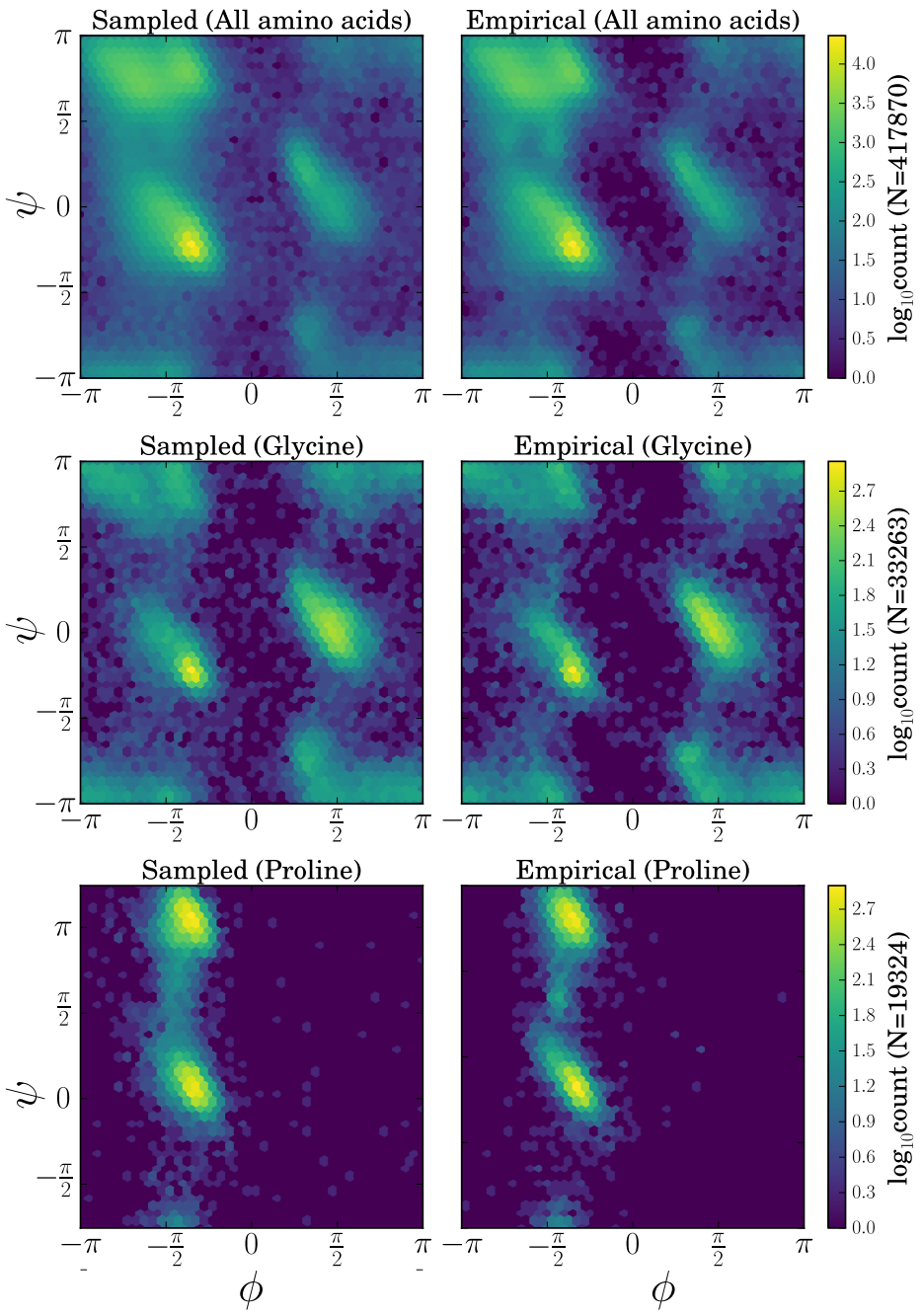


Figure 3.6: Ramachandran plots depicting sampled and empirical dihedral angle distributions. The top row depicts the distributions for all amino acids, the middle for glycine only and the bottom for proline only. The leftmost plots show dihedral angles sampled under the jump model, whereas the rightmost plots show the empirical distributions of dihedral angles in the training dataset.

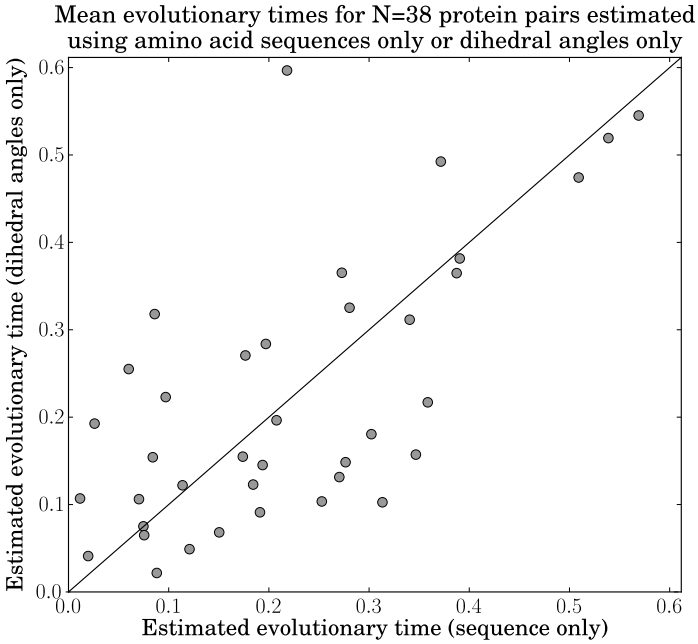


Figure 3.7: Scatterplot comparing evolutionary times estimated using pairs of homologous amino acid sequences only versus pairs of homologous sets of dihedral angles only for $N = 38$ proteins pairs in the test dataset. The x-coordinate of each point gives the estimated evolutionary time based only on the amino acid sequence, whereas the y-coordinate gives the estimated evolutionary time based only on the dihedral angles. The diagonal line represents $y = x$.

3.3.3 Estimates of evolutionary time from dihedral angles are consistent with estimates from sequence

Whilst ETDBN has a general scope with respect to applications (including acting as proposal distribution or as a building block in a homology modelling application), we envision the primary application being inference of evolutionary parameters.

Figure 3.7 compares evolutionary times estimated using only pairs of homologous amino acid sequences versus only pairs of homologous dihedral angles. As desired, the two estimates of evolutionary time for each protein pair are similar, as can be seen by the proximity of the points to the identity line.

A paired t-test gave a p-value of 0.578, thus failing to reject the null hypothesis that there is no difference between branch lengths estimated using sequence only vs. angles only. This indicates that there is sufficient evolutionary information in the dihedral angles to estimate the evolutionary times and that the model is consistent in its estimates, lacking a significant tendency to under-estimate or over-estimate the

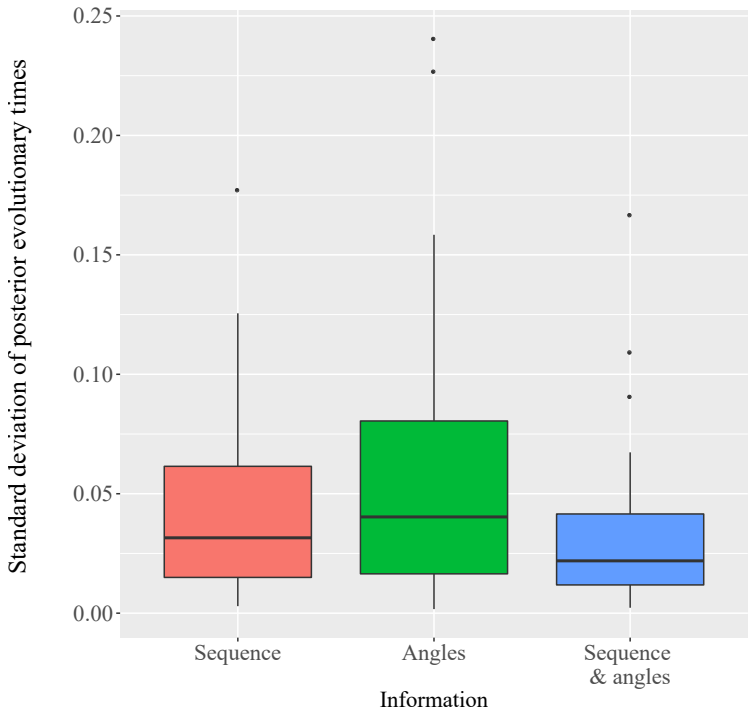


Figure 3.8: Figure produced by Eduardo García-Portugués. Box plots depicting the distribution of posterior standard deviations for the protein pairs in the test dataset under three different scenarios. The mean standard deviations were 0.43, 0.58 and 0.33 for A, B and C, respectively.

evolutionary times when either sequence or dihedral angles are used.

3.3.4 Using sequence and structure reduces the uncertainty associated with estimates of evolutionary time

The left half of Figure 3.8 depicts boxplots of posterior standard deviations of the evolutionary times inferred for the proteins in the testing dataset under three different conditions (sequence only, angles only, and both sequence and angles). A one-sided Wilcoxon signed-rank test was used to test whether the standard deviations were significantly smaller when using both types of observations, compared to using only sequence, or dihedral angle information. As might be expected, the standard deviations of the posterior evolutionary times were significantly smaller when both the sequence and dihedral angle observations were used for posterior inference, compared to using only sequences (p-value = 0.00014) or only dihedral angles (p-value = 0.00002). Hence, the model's precision on the posterior of t_{ab} improves

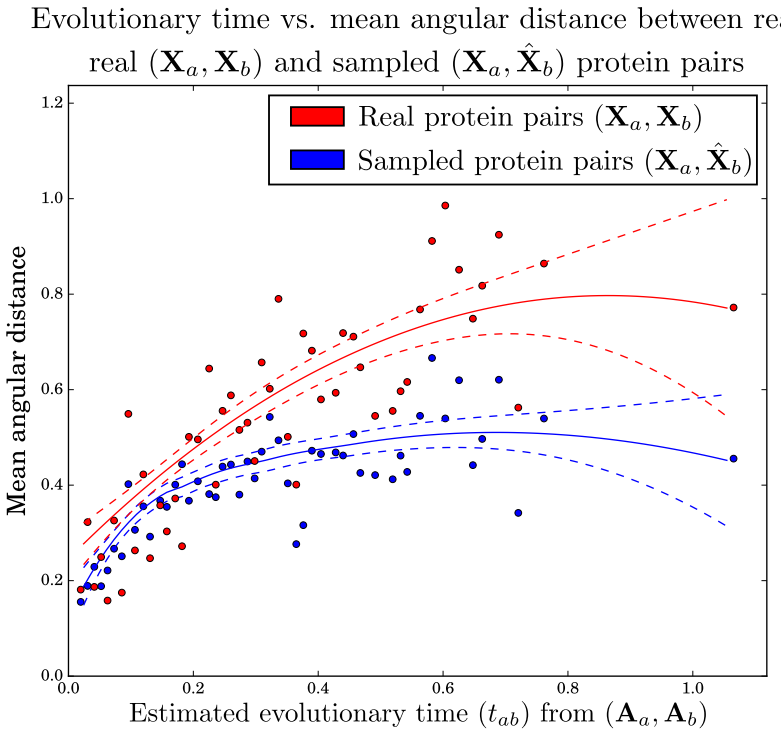


Figure 3.9: Evolutionary time vs. angular distances for real and corresponding sampled proteins pairs in the training dataset at 50 representative evolutionary times. Mean angular distances (see 'Calculation of angular distances' in the Supplementary Material) between the real dihedral angles in a protein pair (red, $\mathbf{X}_a$ and $\mathbf{X}_b$) and sampled dihedral angles in a sampled protein pair (blue, $\mathbf{X}_a$ and $\hat{\mathbf{X}}_b$) were compared to test how well the sampled dihedral angles reproduced the real angular distances. The dihedral angles ($\hat{\mathbf{X}}_b$) of each sampled protein pair were sampled by conditioning on both amino acid sequences and the homologous dihedral angles ($\mathbf{A}_a$, $\mathbf{A}_b$, $\mathbf{X}_a$), and the estimated evolutionary time ($\hat{t}_{ab}$) for the real protein pair. The regression curves were obtained by a quadratic LOcally-weighted regrESSion (LOESS), with smoothing parameter chosen by leave-one-out cross-validation. The 95% confidence intervals for the mean assume error normality.

as more conditioning information is considered. Finally, note also that the sequence alone provides sharper posteriors than the dihedrals alone, a fact likely explained by the direct relation between t_{ab} and $(\mathbf{A}_a, \mathbf{A}_b)$.

3.3.5 The relationship between evolutionary time and angular distance is adequately modelled

We investigated the relationship between evolutionary time and angular distance between real protein pairs and protein pairs where the dihedral angles of $\mathbf{P}_b$ ($\mathbf{X}_b$) were treated as missing and hence sampled (Figure 3.9).

As expected, for both real and sampled pairs, angular distance tends to increase as a function evolutionary time. For larger evolutionary times a plateau begins to emerge, this in part due to back-substitutions and the fact that the theoretical angular distance is bounded above by $\sqrt{8} \approx 2.828$.

When the evolutionary time is exactly zero ($t_{ab} = 0$) under our model, the angular distance between sampled dihedral angles is exactly zero (not shown in Figure 3.9); however, this is not expected to be the case for real protein pairs when the two sequences are identical (due to the inherently flexible nature of proteins, different experimental conditions, experimental noise, etc.). It is therefore not surprising that the regression curve for the real protein pairs does not pass through zero.

For small evolutionary times (< 0.2) the curves for the real and sampled protein pairs show a good correspondence; however, for larger evolutionary times the model tends to under-estimate angular distances. This may reflect that the tpd of the WN diffusion specified is localised around its mean, even when the evolutionary time is large, therefore dihedral angles distant from this mean are unlikely to be sampled. To a certain extent this is mitigated by the jump model, which occasionally allows for large changes in dihedral angle, but may still be somewhat limited in its flexibility, as jumps can only occur between two site-classes. The majority of protein pairs in our training dataset represent smaller evolutionary times (81.7% of evolutionary times are smaller than 0.4) and therefore protein pairs with larger evolutionary times and their associated jumps are under-represented in our dataset, which may also explain the under-estimation.

An additional reason may be that ETDBN does not attempt to model global dependencies. Echave and Fernández (2010) use a LFENM model (which does take into account global dependencies), and provide evidence showing that the majority of structural changes are due to collective global deformations rather than local deformations. A local model such as ETDBN, by definition, does not take into account global dependencies and therefore does not fully account for their contribution to structural divergence.

3.3.6 Structural inference

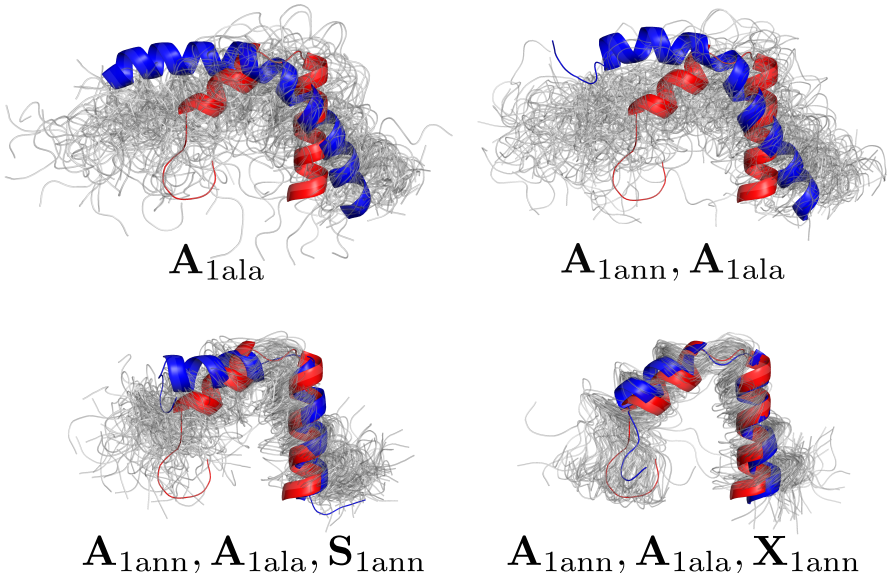


Figure 3.10: Samples from ETDBN of a 35 amino acid fragment of an annexin protein under four different scenarios. In red is the native protein structure (protein b, corresponding to positions 100-135 of PDB 1ala) and in grey is 100 samples from the model corresponding to the same region. In blue is the centroid structure that is based on 5000 samples from the model (the centroid is the sample with the lowest average RMSD to all other samples). The homologous protein used for the purposes of prediction (protein a) was PDB 1ann.

The conditional independence structure in (3.1) enables computationally efficient sampling from the model under different combinations of observed and missing data. For example, ETDBN can be used to sample (*i.e.* predict) the dihedral angles of a protein from its corresponding amino acid sequence, a homologous amino acid sequence, a homologous set of dihedral angles, the corresponding secondary structure, a homologous secondary structure, or any combination thereof.

In Figure 3.10 we provide a detailed graphical example of a single pair of homologous annexin proteins (PDB 1ala and PDB 1ann) for which sampling of PDB 1ala (*i.e.* prediction of the dihedral angles, $\mathbf{X}_{1ala}$, for PDB 1ala) was performed under four different conditions: (i) from the amino acid sequence of PDB 1ala alone (Figure 3.10 $\mathbf{A}_{1ala}$), (ii) the amino acid sequence of PDB 1ala and a homologous amino acid sequence of PDB 1ann (Figure 3.10 $\mathbf{A}_{1ann}, \mathbf{A}_{1ala}$), (iii) using both sequences and the secondary structure of the homologous protein (Figure 3.10 $\mathbf{A}_{1ann}, \mathbf{A}_{1ala}, \mathbf{S}_{1ann}$),

and (iv) using both sequences and the dihedral angles of the homologous protein (Figure 3.10 $\mathbf{A}_{1\text{ann}}, \mathbf{A}_{1\text{ala}}, \mathbf{X}_{1\text{ann}}$). PDB 1ala and PDB 1ann are moderately diverged, having an amino acid sequence identity of 56%. Whilst the full protein (PDB 1ann) was sampled, only a 35 amino acid fragment is depicted.

The reason for depicting only a small fragment is that ETDBN is considered a local model and is not designed to capture the global properties of protein structure. For example, proteins have a strong tendency to be globular (compact) in nature – a global, coarse-grained property not enforced by our model. Therefore ETDBN does not constitute a complete homology modelling method in itself. Rather, it can be used as a building block (much like fragment libraries model local structure (Rohl *et al.*, 2004)) in protein structure prediction and homology modelling methods.

To obtain an overall view, predictive accuracies of dihedral angles predictions using 38 protein pairs $(\mathbf{P}_a, \mathbf{P}_b)$ were measured. The dihedral angles of $\mathbf{P}_b$ in each pair were treated as missing, and 5000 sequences of dihedral angles were sampled under six different combinations of observations for each protein. Following that, predictive accuracy was measured using the average angular distance (see Section 3.2.7) between the sampled $(\hat{\mathbf{X}}_b)$ and known $(\mathbf{X}_b)$ dihedral angles. Figure 3.11 shows the results. In 3.11A, no data was used for prediction, hence the predicted angles correspond to random draws from the model. In 3.11B–3.11E, the following observations were used, respectively: $\mathbf{A}_b$, $(\mathbf{A}_a, \mathbf{A}_b)$, $(\mathbf{A}_a, \mathbf{A}_b, \mathbf{S}_a)$, and $(\mathbf{A}_a, \mathbf{A}_b, \mathbf{X}_a)$. Finally, in 3.11F the same combination of observations was used as in 3.11E, but the sequence alignment was treated as known *a priori* instead of marginalised. These results show that increasingly informative observations leads to better predictive accuracy, which is consistent with our expectations.

It should be emphasised that we do not expect ETDBN to compete with structure prediction packages such as Rosetta (Rohl *et al.*, 2004) or homology modelling software such as Arnold *et al.* (2006) in terms of predictive accuracy. Our current model is a local model of structure evolution – it is not even expected capture fundamental constraints such as the radius of gyration of a protein or other global features typical

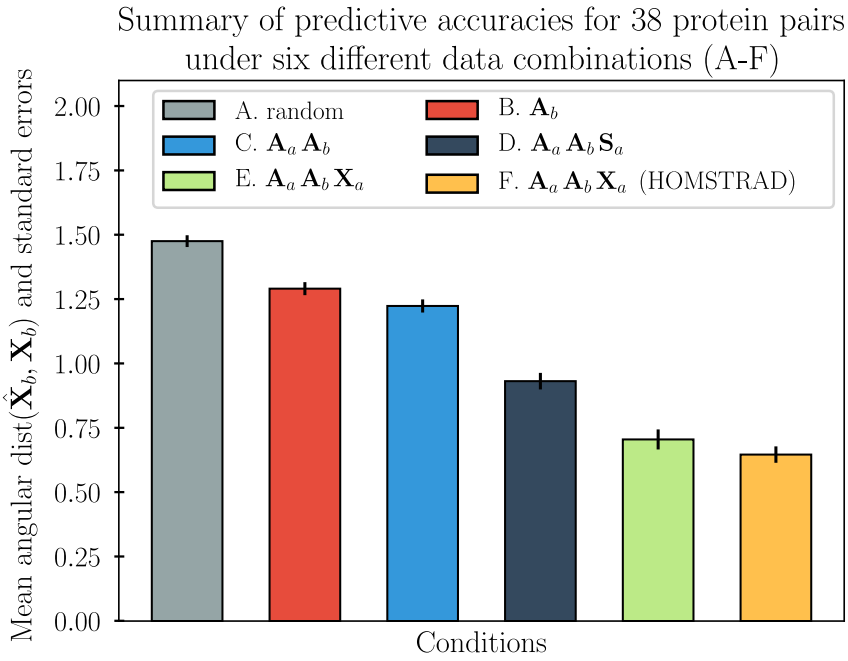


Figure 3.11: Mean predictive accuracies (measured using angular distance, lower is better) and standard errors for 38 protein pairs under six different combinations of observations. The dihedral angles $\mathbf{X}_b$ of protein b were treated as missing and were sampled under the model, whereas $\mathbf{P}_a$ was a homologous protein used for the purposes of prediction.

of proteins.

3.3.7 Evolutionary hidden states reveal common evolutionary motifs

One benefit of ETDBN is that the 64 evolutionary hidden states learned during the training phase are interpretable. We give an example of a hidden state encoding a jump event that was subsequently found to represent an *evolutionary motif* present in numerous protein pairs in our test and training datasets.

Evolutionary hidden state 3 (Figure 3.12) was selected from the 64 hidden states as an example of a hidden state encoding a jump event and capturing angular shift (a large change in dihedral angle). A notable feature of this hidden state is that the change in dihedral angles between site-classes r_1 and r_2 is associated with specific amino acid changes. In site-class r_1 the amino acid frequencies are relatively spread out amongst a number of amino acids. In site-class r_2 the frequencies are particularly

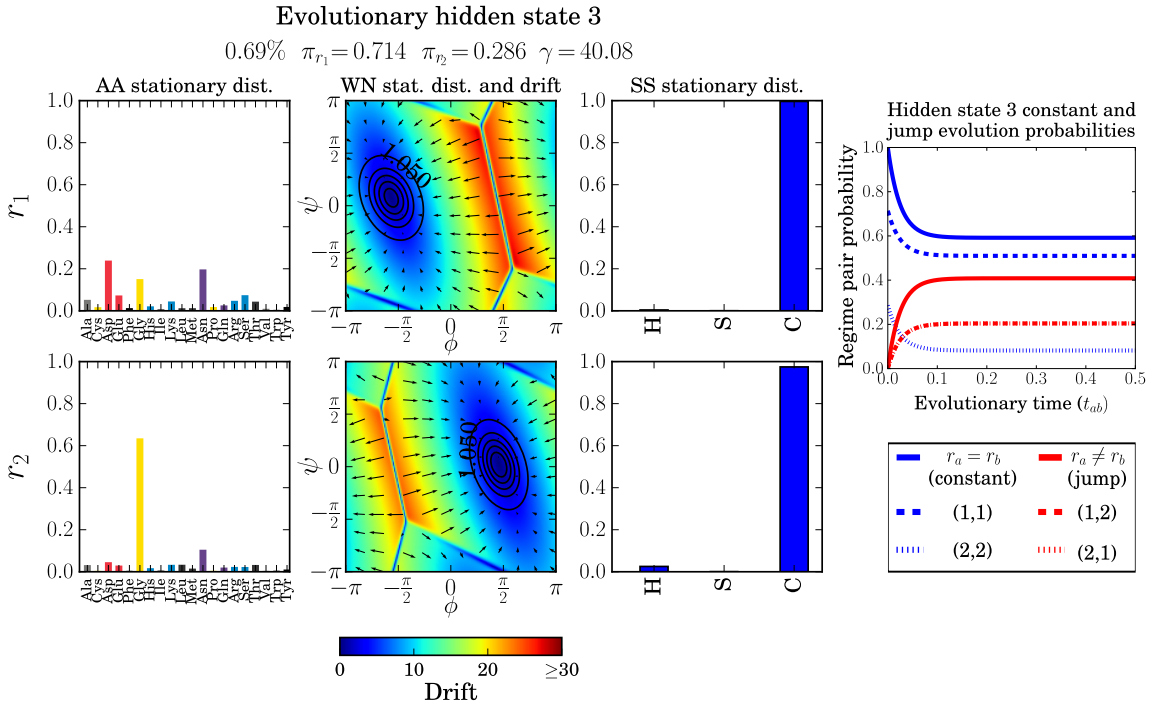


Figure 3.12: Depiction of evolutionary hidden state 3. This hidden state was sampled at 0.69% of sites (the average was 1.56%). The equilibrium frequencies of r_1 and r_2 were $\pi_1 = 0.691$ and $\pi_2 = 0.309$, respectively. The jump rate was $\gamma = 31.76$. The corresponding site-class pair probabilities are depicted to the right as a function of evolutionary time. Note that the dashed red lines depicting the probabilities for $(1, 2)$ and $(2, 1)$ superimpose exactly, because the probabilities are equal – this holds for the jump probabilities of all hidden states as it is required for time-reversibility. In the main figure, the two rows depict the parameters encoded by the two site-classes, respectively. Columns 1 and 3 depict the parameters governing the the amino acid and secondary structure stochastic processes, respectively. The secondary structure classes correspond to H=helix, S=sheet and C=coil. Column 2 depicts the WN diffusions. The stationary distributions of the WN diffusions are shown using black contour lines, the direction of the drifts are indicated by the arrows and the magnitude of the drifts at each position indicated using the colour gradient.

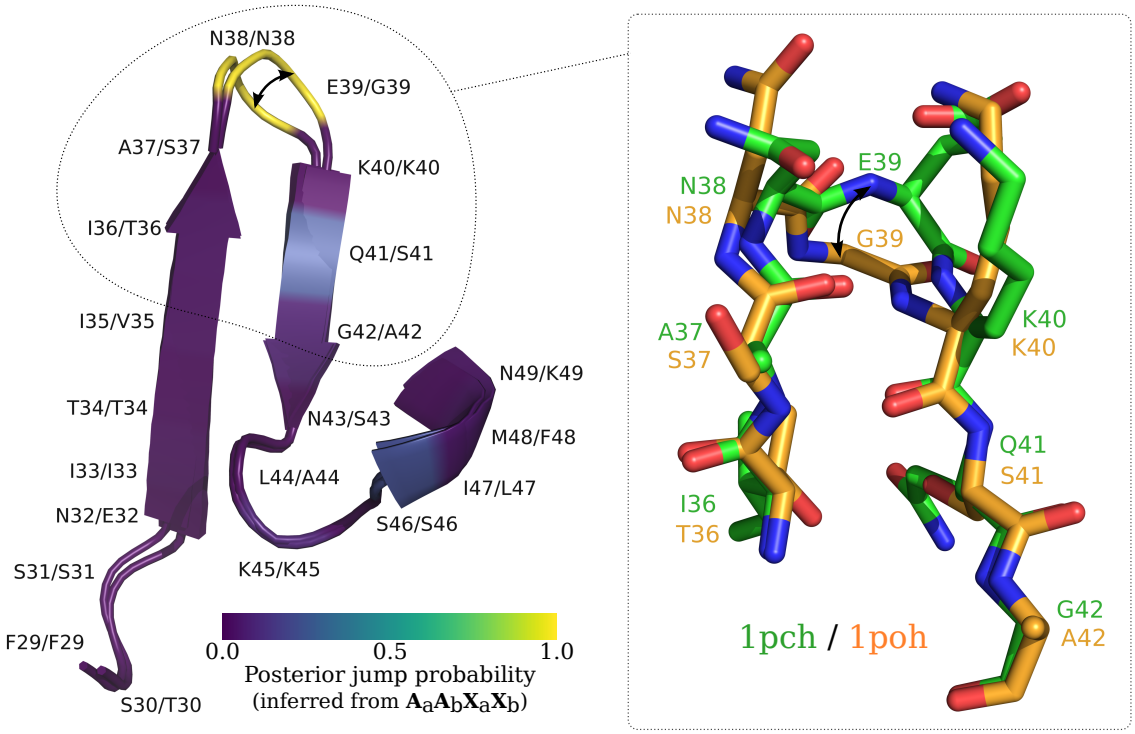


Figure 3.13: Depiction of two histidine-containing phosphocarriers, PDB 1pch and 1poh, superimposed. On the left is a cartoon representation of the two proteins corresponding to regions F29-K49 and F29-A42, respectively, with posterior jump probabilities at each position overlaid. On the right is a ball-and-stick representation giving atomic detail for a smaller region (I36-G42 and T36-A42, respectively). The exchange between a glutamate (E39 in 1poh) and a glycine (G39 in 1pch) is associated with a large change in dihedral angle as indicated by the curved arrows.

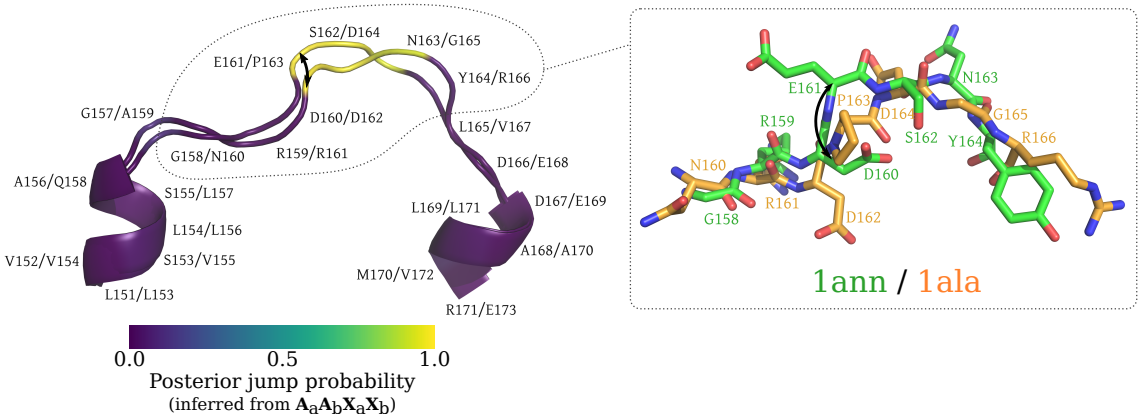


Figure 3.14: Depiction of two homologous annexin proteins, PDB 1ann and 1ala, superimposed. On the left is a cartoon representation of the two proteins corresponding to regions L151-R171 and L153-E173, respectively, with posterior jump probabilities at each position overlaid. On the right is a ball-and-stick representation giving atomic detail for a seven amino acid region (G158-Y164 and N160-R166, respectively). The exchange between a glutamate (E161 in 1ann) and a proline (P163 in 1ala) is associated with a large change in dihedral angle as indicated by the curved arrows.

concentrated in favour of glycine (Gly) and asparagine (Asp), with glycine being somewhat more probable in site-class r_2 than r_1. This suggests that, conditioned on hidden state 3, an exchange between a glycine to another amino acid is likely indicative of a jump and hence a corresponding change in dihedral angle. This is consistent with what we find in a subsequent analysis of evolutionary motifs. This particular jump occurs in coil regions.

Having selected hidden state 3, positions in 238 protein pairs were analysed for evidence of the corresponding evolutionary motif. 38 protein pairs in the test dataset and a further 200 from the training dataset were analysed using the criteria described in the Methods section. Using the first criterion, 84 protein sites in 59 protein pairs corresponding to $H^i = 3$ (evolutionary hidden state 3) were identified. Of the 84 protein sites, 34 protein sites met the second criterion.

We give an example of a homologous protein pair illustrating the identified evolutionary motif. Two histidine-containing phosphocarriers, 1pch (*M. capricolum*) and 1poh (*E. coli*), were identified as having the evolutionary motif (Figure 3.13) at homologous site E39/G39.

Most positions in the homologous pair have low posterior jump probabilities (≈ 0.0), except for positions N38/N38 and E39/G39, which both have high posterior jump probabilities (≈ 1.0). The exchange between a glutamate (at position 39 in 1poh) and a glycine (at position 39 in 1pch) appears to be responsible for the shift in dihedral angle. This exchange corresponds to a significant jump in dihedral angle: $\langle \phi_{\text{1poh,E39}}, \psi_{\text{1pch,E39}} \rangle = \langle -1.63, -0.06 \rangle \rightarrow \langle \phi_{\text{1poh,G39}}, \psi_{\text{1pch,G39}} \rangle = \langle 1.40, 0.22 \rangle$. The angular distance between the two dihedral angles is 2.01. This directional exchange is consistent with the amino acid frequency parameters specified by the two site-classes for hidden state 3 (Figure 3.12).

Site-class r_1 indicates that a number of amino acids (alanine, aspartic acid, glycine, histidine, lysine, asparagine, proline, glutamine, arginine, serine and theorine) other than glutamate plausibly coincide with the particular dihedral angle conformation specified by site-class r_1. The involvement of glycine in a jump is not surprising as

it is a small and flexible amino acid, whereas the role of asparagine is less clear. In our analysis of 238 protein pairs we found that of the seven positions meeting the criteria for hidden state 3 and involving an exchange with asparagine (Asn), four were an exchange between an asparagine and a glycine, whereas the remaining three were between asparagine and one of lysine, histidine or serine.

We also performed a search of protein pairs containing the jump represented by evolutionary hidden state 13 (Figure B.1 in the appendix). Twelve aligned sites in eleven different protein pairs corresponding to $H^i = 13$ (evolutionary hidden state 13) were identified. A homologous pair of annexin proteins, PDB 1ann (from *Bos taurus*) and PDB 1ala (from *Gallus gallus*), were selected out of the eleven pairs with the identified jump (Figure 3.14) at aligned site E161/P163. Most aligned sites in the annexin pair had low posterior jump probabilities (≈ 0.0), with the exception of three successive aligned sites starting with the aligned site of interest, E161/P163, which had high posterior jump probabilities (≈ 1.0). Site-classes r_1 and r_2 indicated that an exchange between a proline and a glutamate at positions 161 and 163 in PDB 1ann and PDB 1ala, respectively, is informative of the specified dihedral angle change.

3.3.8 Dihedral angle transition densities are informed by amino acids changes

As was seen in the previous example, the jump model allows amino acid changes to inform changes in dihedral angles. Here we provide illustrations of transition densities calculated under the model to further demonstrate that this is the case. For example, conditioned on hidden state 3 (Figure 3.12), a Glycine transitioning to a Glycine, is not expected to be strongly informative of a large change in dihedral angle, as can be seen in the dihedral angle transition densities at different time points in row 2 of Figure 3.15, which remain concentrated around the initial dihedral angle conformation. On the other hand, a Glycine to Asparagine conditioned on hidden state 3 (row 3 of Figure 3.15) is highly informative of a large dihedral change, with

the transition density rapidly shifting to a separate region of the Ramachandran plot.

Quite complex patterns of transition densities are possible under the full model, as can be seen in Figure 3.16, this is due to the HMM structure of ETDBN effectively creating a mixture of hidden states. Even in the case of self-transitions (e.g. Asparagine to Asparagine in row 2 of Figure 3.16), the transition densities have non-elliptical, and multi-modal structure.

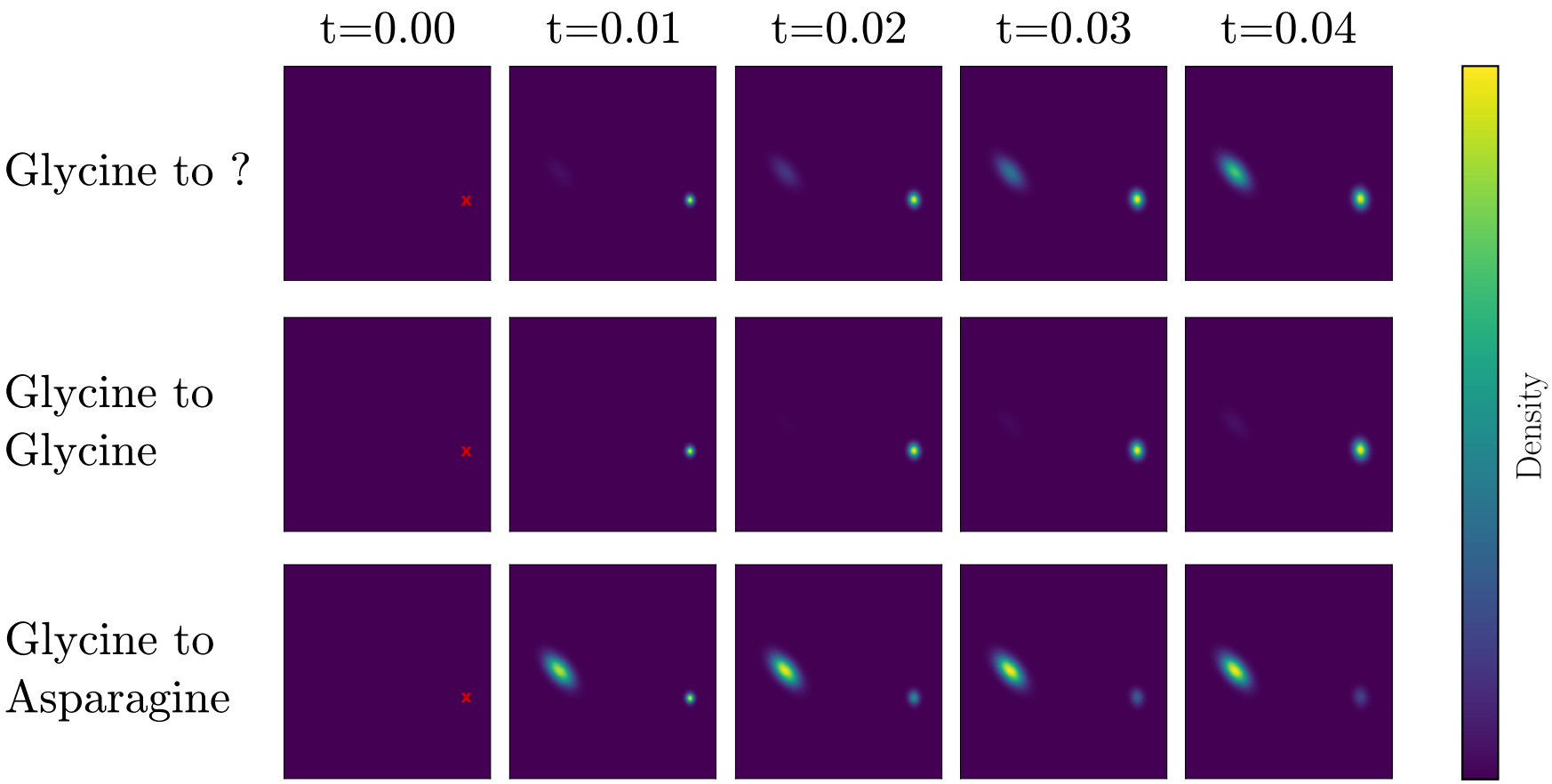


Figure 3.15: Dihedral angle transition densities at five different time points, conditioned on hidden state 3 and three different combinations of start and end amino acids (i. Glycine to any amino acid, ii. Glycine to Glycine, and iii. Glycine to Asparagine). The red x marks the initial dihedral angle conformation $(\frac{3\pi}{4}, -\frac{\pi}{4})$. Animations for the above transition densities are available in the PDF version of this document at the following links: i, ii, and iii.

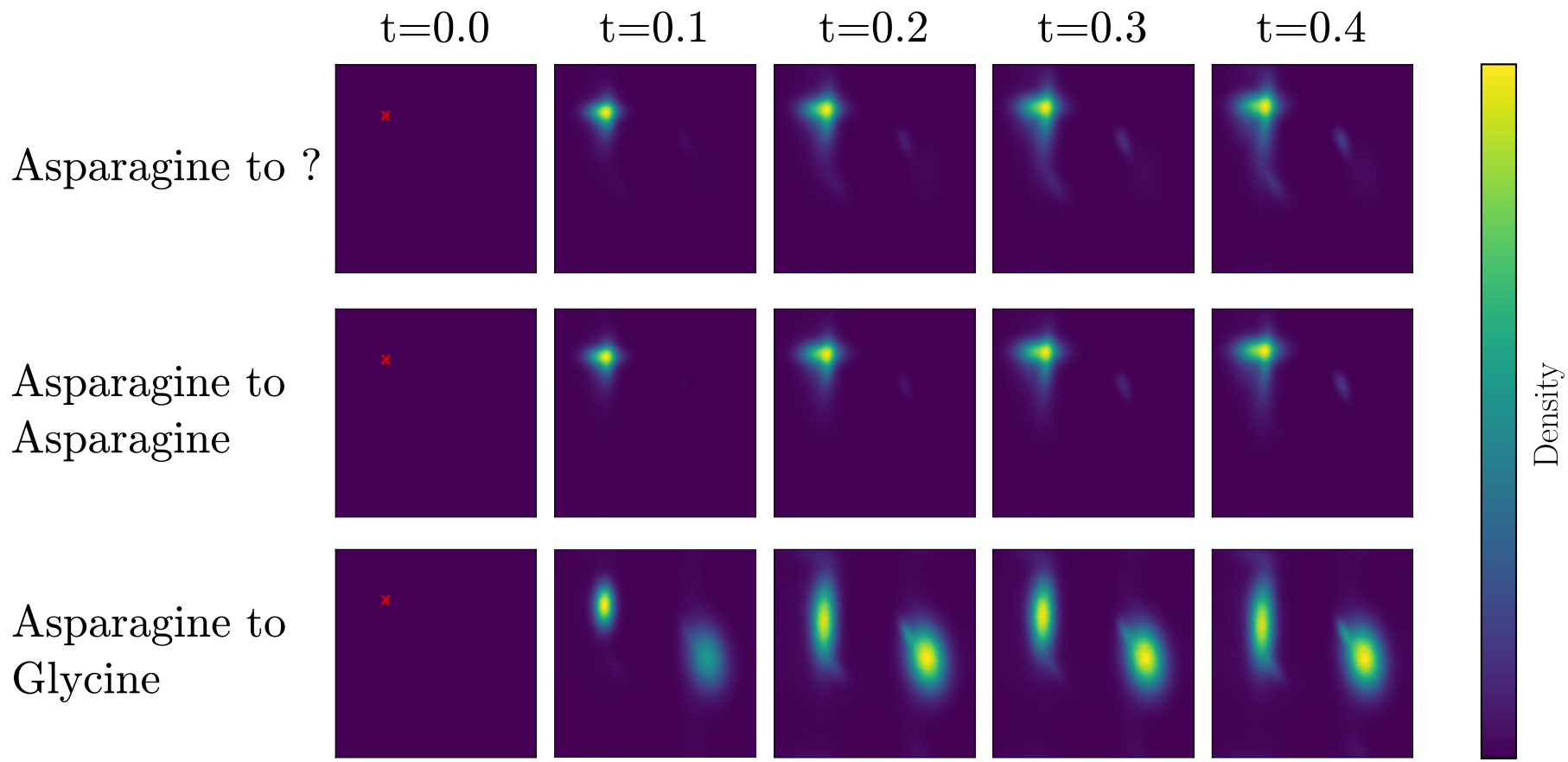


Figure 3.16: Dihedral angle transition densities at five different time points, summed over all 64 hidden states, and conditioned on three different combinations of start and end amino acids (i. Asparagine to any amino acid, ii. Asparagine to Asparagine, and iii. Asparagine to Glycine). The red x marks the initial dihedral angle conformation $(-\frac{\pi}{2}, \frac{\pi}{2})$. Animations for the above transition densities are available in the PDF version of this document at the following links: i, ii, and iii.

3.3.9 Using dihedral angles for alignment

A valuable feature of our model is its ability to account for alignment uncertainty by summing over possible pairwise alignments using the TKF92 model as a prior distribution over indel histories, whilst simultaneously taking into account neighbouring dependencies amongst aligned sites. Doing so results in a sample of alignments rather than a single alignment. Nevertheless, a single Maximum A Posteriori (MAP) pairwise alignment may be obtained from the alignment samples and used for downstream analysis.

ETDBN and several other alignment methods (namely StatAlign, BAli-Phy, MUSCLE and MAFFT) were used to infer pairwise alignments from simulated and real data under various combinations of data observations. For example, an amino acid sequence pair $(\mathbf{A}_a, \mathbf{A}_b)$, a secondary structure sequence pair $(\mathbf{S}_a, \mathbf{S}_b)$, a dihedral angle sequence pair $(\mathbf{X}_a, \mathbf{X}_b)$ and combinations thereof.

In the first set of benchmarks (Figure 3.17A), pairs of proteins were simulated from the ETDBN model conditioned on 38 different pairwise alignments and corresponding evolutionary times. This resulted in a set of 38 simulated pairwise alignments together with corresponding observations, implying that the true underlying alignments were known for each of the simulated protein pairs. ETDBN and a number of other alignment methods were used to infer pairwise alignments for each pair. The alignment similarity metric (Schwartz *et al.*, 2005) was used to measure the similarity between the inferred alignments and the true alignments, where higher similarity indicates better predictions.

It was found that, when using the simulated amino acid alignments alone, ETDBN (3.17A.5) outperformed all four other methods tested (3.17A.1 MUSCLE, 3.17A.2 MAFFT, 3.17A.3 StatAlign, 3.17A.4 BAli-Phy). These benchmarks of simulated data will favour ETDBN, because the data were simulated under ETDBN, and thus should not be considered a fair comparison.

More revealing in Figure 3.17A was the alignment similarity under ETDBN when using different combinations of simulated data observations. Secondary structure

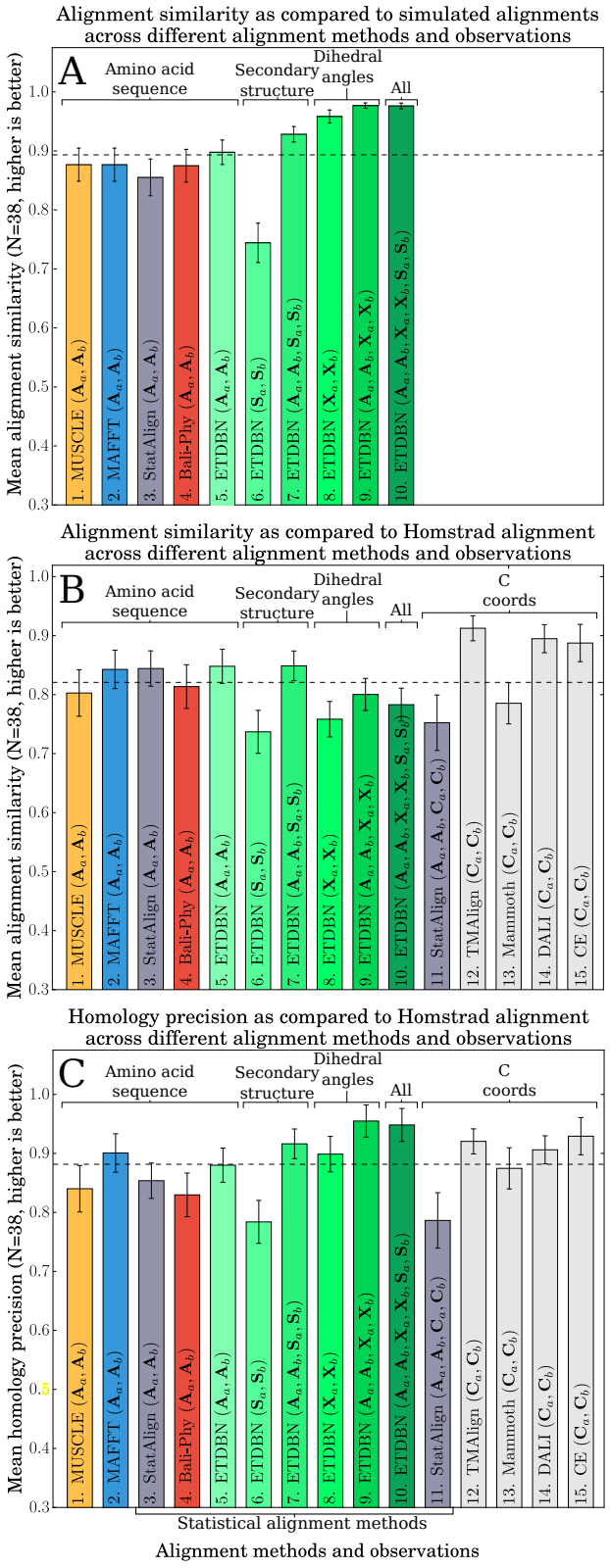


Figure 3.17: Alignment benchmarks. A: mean alignment similarity across different methods and combinations of data observations, where the observations were simulated from the ETDBN model conditioned on a single known alignment. B: mean alignment similarity, where the 38 HOMSTRAD alignments in the test dataset were taken as the true alignments. C: mean precision in predicting homologous site pairs in 38 sequence pairs from the test dataset, where homologous site pairs in the HOMSTRAD alignments were taken to be the true homologous site pairs.

alone (3.17A.6) performed the worst, which is unsurprising given that only three states were available to align the proteins. The second worst in terms of alignment similarity was amino acid sequences alone (3.17A.5), followed by amino acid sequences, and secondary structures (3.17A.7). Interestingly, using dihedral angles only (3.17A.8) outperformed both 3.17A.5 (sequences only) and 3.17A.6 (secondary structures only). Finally, using amino acid sequence together with dihedral angles (3.17A.9) or all three data types combined (3.17A.10) outperformed all other combinations. This illustrates that, at least under simulation conditions, increasing the number of data observations results in better alignment accuracy.

Following that, the alignment methods were benchmarked against 38 pairwise alignments consisting of real sequence and structure observations in the test dataset. These pairwise alignments were obtained from the HOMSTRAD alignments. The sequence identity of these pairwise alignments ranged from 10% to 93%, with an average sequence identity of 39%. In addition to methods $1-10$ in Figure 3.17A, five structural alignment methods were also used: 11. StatAlign (Herman *et al.*, 2014), 12. TMAlign (Zhang and Skolnick, 2005), 13. Mammoth (Ortiz *et al.*, 2002), 14. Dali (Holm and Rosenström, 2010) and 15. CE (Shindyalov and Bourne, 1998). Methods 11-14 were not benchmarked in 3.17A due to the lack of an appropriate model for simulating the evolution of three-dimensional protein structures including an indel evolutionary process.

When benchmarking the MAP estimated alignments against the HOMSTRAD alignments (Figure 3.17B), using real sequences alone for inference (A_a, A_b), ETDBN (3.17B.5) had a similar degree of accuracy when compared to several other sequence-based methods (3.17B.1 StatAlign, 3.17B.2 BAli-Phy, 3.17B.3 MUSCLE and 3.17B.4 MAFFT). This demonstrates that ETDBN has performance comparable to that of other commonly-used sequence alignment methods.

Using (S_a, S_b) alone, ETDBN (3.17B.6) had substantially lower alignment similarity compared to sequence only, which was expected given that a similar result was obtained for the simulated data (3.17A.6). However, when including the real

sequences (3.17B.7) the predictive accuracy was once again comparable to sequence only inferences (3.17B.1–3.17B.5).

When using $(\mathbf{X}_a, \mathbf{X}_b)$ alone (3.17B.8), the alignment similarity was found to be somewhat worse than the sequence only cases. Furthermore, when introducing the sequences (3.17.9) and secondary structures (3.17B.10) in addition to the dihedral angles, the similarity remained worse than the sequence only methods (3.17B.1–3.17B.5), despite the additional information. These results are in contrast to the results we obtained for simulated data (3.17A.8–3.17A.10).

The non-statistical structural alignment methods (3.17B.12–3.17B.15) faired the best, likely because they use a RMSD minimization criteria similar to that used to align the HOMSTRAD alignments. When interpreting these results it is important to note the HOMSTRAD alignments should not be considered the true underlying alignments and may even be strongly biased. For example, they likely favour the closest structural superimposition of structures as measured using RMSD, with the fewest number of indels. In the evolutionary modelling context our goal is to distinguish between homologous sites (sites that have evolved via mutation alone) and indels. In practice, it is extremely difficult to obtain the true underlying alignment (sets of homologies and indels), because it requires an experiment where every indel event since the common ancestor is observed, a seeming impossible task outside of simulation or laboratory conditions.

After further investigation, the trend of lower alignment similarity seen in Figure 3.17B.6–3.17B.10 when using ETDBN with structural observations compared to sequence only or non-statistical structural alignment methods was found to reverse (3.17C.6–3.17C.10) upon calculating the precision of predicting homologous sites (the fraction of sites which were predicted as homologous and were correctly predicted as such). Therefore, when only dihedral angle observations are used, ETDBN underpredicts the number of homologous sites; however, when a homologous site is predicted, it is correctly predicted more often than when using only amino acid sequences. In particular, ETDBN predicted fewer homologous sites with coiled secondary structure

compared to homologous sites with helical or sheet secondary structure. This under-prediction of homologous sites may be in part due to the WN diffusion used to model evolution of dihedral angles and the approximate nature of the jump model. Together they are expected to be capable of modelling angular drift (small changes in angle), but will be less capable of modelling angular shift (large changes in angle).

3.4 Concluding remarks

The main achievement of this work is a computationally tractable, generative and interpretable probabilistic model of protein sequence and structure evolution on a local scale.

Previous stochastic models of protein sequence and structure evolution emphasised estimation of evolutionary parameters (Challis and Schmidler, 2012; Herman *et al.*, 2014). ETDBN is somewhat of a departure from these previous models, but is likewise capable of estimating evolutionary parameters. We show that estimates of evolutionary times inferred under ETDBN are consistent regardless of whether amino acid sequence, or dihedral angle observations are used. In addition, the relationship between evolutionary time and angular distance in real proteins is adequately recapitulated in protein pairs sampled under the model, albeit the angular distance is under-estimated for larger evolutionary times, which might be explained by the limited flexibility of the jump model and the lack of taking into account global dependencies.

Like previous models, ETDBN is capable of dealing with alignment uncertainty by marginalising over indel histories; it predicts pairwise MAP consensus alignments with accuracy similar to that of score-based and statistical alignment methods.

The generative nature of ETDBN allows us to demonstrate that the underlying empirical distributions over dihedral angles (depicted using Ramachandran plots) are captured and that the model is capable of predicting missing observations, such as dihedral angles, from a variety of different data types. For example, an amino acid sequence, a homologous amino acid sequence, a homologous secondary structure, a homologous set of dihedral angles, or any combination thereof.

Due to its local nature, ETDBN does not constitute a homology modelling method in itself. Rather, it can be used as a building block, much like fragment libraries model local structure in protein structure prediction methods. ETDBN places the homology modelling problem on a statistical footing, enabling a number of approaches to later be used, such as multi-level modelling, *i.e.* combining fine-grained distributions (for

example, distributions over dihedral angles, such as ETDBN) and coarse-grained distributions (for example, distributions describing the global properties of proteins, such as compactness). In particular, a method referred to as 'the Reference Ratio method' can be used to combine fine-grained and coarse-grained distributions in a statistically principled manner (see Hamelryck *et al.* (2010); Frellsen *et al.* (2012)). However, we have shown that the current model can already be used for the inference of evolutionary parameters, such as alignment, in its present form.

In addition to multi-level modelling, probabilistic models, such as ETDBN, allow one to account for and to make rigorous statements about uncertainty (e.g. with respect to evolutionary time, alignment, etc.). In principle, ETDBN, like TorusDBN (Boomsma *et al.*, 2008), may used as a proposal distribution. In other words, ETDBN could be used to sample protein structures (possibly conditioned on various data observations) in a computationally efficient manner, such that the resulting samples are expected to be located in regions of high probability density with respect to the true underlying distribution.

A final key feature of our evolutionary model is its interpretable nature. This interpretability enables the identification of potential evolutionary motifs – common patterns of sequence-structure evolution. We identify one such evolutionary motif in 34 different homologous protein pairs. A major direction for future research is the further identification of such evolutionary motifs. Understanding these evolutionary motifs, may *i*) improve homology modelling predictions; *ii*) provide more accurate estimates of evolutionary parameters; and *iii*) produce better models of protein evolution that more realistically capture evolutionary trajectories through sequence and structure space, which may help identify functionally relevant positions that are potential drug targets.

3.5 Future challenges

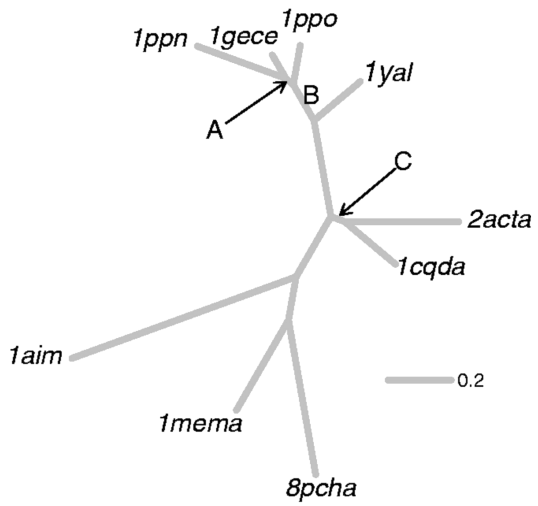
3.5.1 Pairwise to phylogeny

For reasons of computational tractability the implemented model is pairwise, but it is theoretically possible to generalise it to a phylogeny, such as in Herman *et al.* (2014) (see Figure 3.18). To consider a phylogenetic tree, it is necessary to marginalise the sequences and structures of the unobserved ancestral proteins at the internal nodes of the tree. Felsenstein's algorithm (Felsenstein, 1981) can be used to marginalise discrete ancestral states, such as amino acids in a computationally efficient manner; however, a similar efficient algorithm is not known to exist for marginalising the continuous ancestral dihedral angle states under the WN diffusion. Fortunately, MCMC approaches, such as the Metropolis-Hastings algorithm can be used to perform this marginalisation.

Alternatively, due to the periodic and bounded nature of dihedral angles (*i.e.* they exist on square plane, see Figure 3.3) it is also reasonable to discretise the dihedral angles and perform marginalisation using Felsenstein's algorithm. This is expected to result in an accurate approximation of the marginal likelihoods (assuming a sufficiently fine-grained discretisation).

The benefits of having a structural model of evolution on a phylogeny are numerous, for example: better estimates of phylogeny (as can be seen in Figure 3.18), amongst other parameters of interest (e.g. improved homology modelling, estimates of evolutionary rates, dating of sequences, etc.), and the ability to reconstruct ancestral sequences and structures more accurately.

Multiple statistical alignment Whilst it is possible to fix the multiple sequence alignment associated with a phylogeny in advance by pre-aligning the sequences, doing so ignores alignment uncertainty. To perform posterior inference, an MCMC approach that samples alignments along the tree using transducers (Holmes and Bruno, 2001) may be implemented; however, this does not consider neighbouring dependencies. To



	A	B	C
seq-only	0.53	1.00	0.61
seq+struc (*non-phylo*)	0.81	0.97	0.96
seq+struc (*phylo*)	0.97	1.00	1.00

Figure 3.18: Adapted from Herman *et al.* (2014). The consensus topology of a set of nine cysteine proteinases inferred using a series of models developed in Herman *et al.* (2014). Three models were compared: (i) an evolutionary model using sequence data only (*seq-only*), (ii) an evolutionary model using sequence and structural data but ignoring the relationship between sequence divergence and structural divergence (*non-phylo*), and (iii) a model using sequence and structural data whilst taking into account the relationship between sequence and structural divergence (*phylo*). The same consensus topology (depicted above) was obtained under all three models. The labelled nodes correspond to splits with significant uncertainty under the sequence-only model (the other three splits had posterior probability 1.00 in all cases). The table to the right shows the posterior probability of each of these labelled splits under the different model variants. Notice how the addition of structural data (*non-phylo* and *phylo* models) significantly reduces the uncertainty associated with the inferred splits.

overcome this, the transducers could be modified to include additional hidden states representing site-classes that also capture neighbouring dependencies.

3.5.2 Site-class switching and context-dependence

Although we believe our model provides a substantial improvement over current stochastic models of sequence and structural evolution, there is still scope for improvement. The WN diffusions used to model dihedral angle evolution adequately capture angular drift (small local changes in dihedral angle), but are less capable of capturing angular shift (large changes in dihedral angle). This is to a considerable extent mitigated by the introduction of jump events. A more realistic model would consider the entire evolutionary trajectory, allowing an arbitrary number of switches between site-classes together with neighbouring dependencies amongst adjacent sites along the evolutionary trajectory. Similar context-dependent models are typically computationally expensive and require sophisticated inference procedures (Robinson *et al.*, 2003; Yu and Thorne, 2006).

3.5.3 Error modelling

ETDBN ignores the uncertainty associated with the atomic coordinates of protein structures. This uncertainty arises due to the metastable nature of protein and experimental conditions under which they are obtained, amongst other factors (Gutin and Badretdinov, 1994). Dihedral angles will also have associated uncertainties, since they are derived from the same atomic coordinates. If these uncertainties are not modelled, it may lead to biased parameter estimates. For example, variability in dihedral angles may be incorrectly accounted for in the model as additional evolutionary time having occurred at the leaves of the tree, thus leading to inflated branch length estimates. This can be accounted for by introducing an error distribution (as in Herman *et al.* (2014)), that may use additional information, such as experimental B-factors to inform estimates of uncertainty. In a future iteration of our dihedral angle model we intend to model this uncertainty, using a similar approach, whereby

each noisily observed dihedral angle, $\tilde{\mathbf{X}}^i$, is assumed to be distributed according to an error distribution, $p(\tilde{\mathbf{X}}^i|\mathbf{X}^i, \mathbf{B}^i)$, conditional on an associated vector of B-factors $\mathbf{B}^i$, where each $\mathbf{X}^i$ is treated as the 'true' underlying dihedral angle and is marginalised.

3.5.4 Selection analyses

Given that we have constructed a diffusive model on dihedral angles, it is possible to infer the rates of dihedral angle evolution at each aligned site in a sequence alignment of proteins. These site-wise rates can act as a measure of selection. A site-to-site rate variation component could be introduced into the model (similar to Yang (1994)) allowing deviations from the expected rate of dihedral angle change to be measured. Comparing these rates to traditional selection coefficients that compute the ratio of non-synonymous codon substitution rates over synonymous substitution rates (e.g. FUBAR Murrell *et al.* (2013)), may provide useful biological insights, especially given that dihedral angle changes are expected to be more reflective of structural, and likewise functional, changes.

3.5.5 Global structural dependencies



Figure 3.19: Adapted from Hamelryck *et al.* (2012). The Reference Ratio Method applied to sample compact protein structures. The left picture shows twenty typical, non-compact samples obtained using TorusDBN alone, $q(\mathbf{X})$. The right picture shows twenty typical, compact samples from the reference ratio distribution, $p(\mathbf{X})$, which is obtained by combining TorusDBN, $g(\mathbf{X})$, with a suitable normal distribution over the radius of gyration, $p(\mathbf{y}) \sim \mathcal{N}(22\text{Å}, 4\text{Å})$. The amino acid sequence of ubiquitin was used.

Please note the following has been adapted from text written by Thomas Hamel-

ryck:

Although our model accounts for local positional dependencies, it does not consider global structural dependencies, this can lead to posterior samples of proteins that self-intersect or proteins that are unrealistically extended in space, rather than compact as is typical of proteins (Boomsma *et al.*, 2008, 2014). The left of Figure 3.19 shows an example of unrealistically extended proteins sampled from a local model (TorusDBN – the non-evolutionary analogue of ETDBN). Many statistical problems – such as modelling of speech, music or images – have a spatio-temporal structure: they exhibit distinct short-range or local and long-range or non-local features. Such problems - called multi-scale problems - are ubiquitous in science, medicine, business, and engineering.

Short-range or local models - for example based on HMMs are often tractable and easy to estimate at the expense of poor modelling of long-range features. One way to solve this dilemma is through the use of a recently developed method called the Reference Ratio Method (RRM), which is a series of techniques for applying probability kinematics (Diaconis and Zabell, 1982) – an alternative to Bayes' rule for updating belief. Valentin *et al.* (2014) have successfully applied the RRM to non-evolutionary models of protein structure, as in Figure 3.19. This may in principle be extended to evolutionary models such as ETDBN.

For example, the evolutionary analogue of a normal distribution may be used to model the radii of gyration of proteins (similar to Figure 3.19). Where the radii of gyration, $\mathbf{y}$, can be directly calculated from a set of dihedral angles, $\mathbf{X}$, using a function $\mathbf{y} = g(\mathbf{X})$. Because we have an evolutionary model, an OU diffusion can be used instead of a normal distribution to capture evolutionary correlations between the radii of gyration. The global model and detailed local model can then be combined using PK, leading to equations of the form:

$$p(\mathbf{X}) = \frac{p(\mathbf{y})}{q(\mathbf{y})} \times q(\mathbf{X}), \tag{3.13}$$

where $q(\mathbf{x})$ is the high-dimensional local model, $q(\mathbf{y})$ concerns the non-local features

as modelled by the local model (which is typically distributed unrealistically, *i.e.* substantial probability density on highly extended structures) and $p(\mathbf{y})$ concerns the desired non-local features whose distribution is in this instance an OU process with parameters that can be obtained empirically by applying the function $g(\mathbf{X})$ to a database of proteins. Note that $q(\mathbf{y})$ is intractable to compute as it is the density that results from applying the many-to-one function $g(\mathbf{X})$ to each element $\mathbf{X}$ in the state space and summing the corresponding probability $q(\mathbf{X})$ for each element $\mathbf{X}$ that maps to a particular $\mathbf{y}$. The set of techniques referred to as the RRM, was specifically developed to handle this intractability. The left-hand side of equation (3.13) can be interpreted as a Bayesian posterior that is valid on both the local and non-local scale.

The idea of combining different scales of information to improve protein structure prediction is receiving more attention as of late, with many proposing this as likely ingredient in solving the protein structure prediction problem (Morrone *et al.*, 2017). It is therefore only natural to consider how one might combine evolutionary information using a probability distribution such as ETDBN in a statistically rigorous manner with other types of information to improve protein structure prediction.

Chapter 4

Perspectives

While the evolution of sequence has been widely studied, the evolution of structure and the impact of structural constraints on sequence evolution has been less widely studied. This thesis attempts to address this. Chapter 2 describes a sequence evolution model, PINNACLE, which takes into account nucleic acid structural constraints, assuming a simple set of canonical base-pairing rules and a nucleic acid secondary structure that is shared across the entire evolutionary history of the sequences being analysed. Whereas, Chapter 3 outlines a protein evolution model, ETDBN, that allows both structure and sequence to evolve.

It is known that both protein and RNA structure evolve more slowly than the sequences from which they fold, and therefore it is often considered reasonable to assume that the same structural conformation or structural ensemble is shared across a particular evolutionary history, as is assumed in PINNACLE. However, it also clear that as sequences diverge, so do their structures. ETDBN was able to identify a number of common evolutionary motifs - coupled changes in sequence and structure that are expected to impact function. Models such as ETBDN which allow structure to evolve enable the identification of such evolutionary motifs and allow one to reconstruct plausible ancestral structures. This can be helpful in elucidating which sequence and structural constituents are responsible for a particular change in phenotype (Wilson *et al.*, 2015). This is useful from a practical perspective in developing

treatments for a disease that target a particular structure or engineering a molecule to obtain a desired phenotype.

A benefit of the slower rate of structural evolution relative to sequence evolution is that it enables the modelling of evolution further back in time. Challis and Schmidler (2012) have shown that a joint model of sequence and structural evolution permits the inference of distances between proteins in the so-called 'twilight zone' with much lower uncertainties than a model that uses sequence information alone. Similarly, the results of ETDBN show that uncertainties associated with estimated parameters are lower when using structural information in addition to sequence. This is not to say that models that do not directly account for structural evolution are invalid. For example, it is clear from PINNACLE-inferred base-pairing probabilities and selection coefficients that many base-pairings are conserved and have been selectively maintained since the common ancestor of the sequences being analysed, and hence are expected to be functionally consequential. Arguably, features which are conserved in this manner are easier to study and make better drug targets. The disruption of such conserved features is expected to be broadly lethal for a given subset of pathogens, and can, therefore, be targeted as such.

A common statistical paradigm used throughout this thesis was Bayesian inference. Bayesian inference differs from Maximum Likelihood (ML) inference in that prior probability distributions are placed over parameters and posterior probability distributions over parameters are obtained following inference. This differs from ML inference where only a single maximum likelihood estimate for each parameter is obtained. This distinction implies that Bayesian inference is able to jointly take into account the prior uncertainty associated with a set of parameters, and is able to quantify the uncertainties associated with each estimated parameter following inference. For example, PINNACLE uses an SCFG to place a prior distribution over possible secondary structure configurations. Posterior inference enables the prediction of the secondary structure and provides information about the uncertainty of the prediction, for example: base-pairing probabilities between each pair of nucleotides.

A benefit of Bayesian methodology is that it allows prior knowledge to be incorporated in a principled manner, for example: information obtained from wet-lab experiments, such as SHAPE reactivities (Wilkinson *et al.*, 2006), can easily be included in PINNACLE's prior, which can be used to further inform which nucleotides are expected to be base-paired or unpaired. Furthermore, as discussed in Future Perspectives in Chapter 3, the use of Bayesian models allows for 'multi-level modelling' using a form of Bayesian updating called probability kinematics (Diaconis and Zabell, 1982). Probability kinematics can be used to combine information that exist on different scales from one another, for example: local protein structure (e.g. dihedral angles) vs. global protein structure (protein compactness).

Another key statistical methodology used in this thesis were CTMCs – better known as substitution models in the context of sequence evolution. CTMCs have been used in both PINNACLE and ETDBN for modelling the evolution of discrete state characters, such as nucleotides, dinucleotides, amino acids, and secondary structure classes. An assumption generally relied upon when using CTMCs is that of site independence. Site independence implies that the state of a variable corresponding to a particular site at given instant in time is independent of the state of all other variables corresponding to all other sites. The absence of site independence is referred to as context-dependence, and associated models are referred to as context-dependent models. Site independence assumptions are widespread in evolutionary modelling, primarily due to the computational tractability they provide.

Context-dependencies are in fact captured on a limited scale in PINNACLE via the modelling of dinucleotide evolution, because a dependency is modelled between each pair of nucleotide sites assumed to be base-paired; however, inference is tractable using standard methodology due to the fact that the dependencies do not extend beyond an individual base-paired site. For more complicated context-dependent models, the effect of context-dependence may extend to all sites in a sequence. One such example is an HMM-like evolutionary model, where the evolution of hidden states at each site depend on their adjacent neighbours at each point in time. Inference

methodology can no longer treat each site or small subsets of sites as independent, but rather have to consider the effect of each site at each point in time, which implies that the evolutionary trajectories at each site need to be jointly modelled, resulting in a higher computational burden. The HMM and jump model in ETDBN were used to avoid such context-dependencies, effectively approximating an HMM-like evolutionary model. The hidden states of the HMM were used to capture neighbouring dependencies expected in protein structure but without a time evolution component (*i.e* the hidden states were not permitted to change as a function of evolutionary time), whereas the jump model was used to capture evolution whilst ignoring neighbouring dependencies.

Although not used in this thesis, computational statistical methodology now exist which allow reasonably efficient inference under context-dependent models Yu and Thorne (2006); Rao and Teh (2013). In my view, these methodologies will play a central role in future developments in the field of evolutionary modelling. Both ETDBN and PINNACLE can benefit from such methodology. ETDBN can benefit by using a more realistic HMM-like evolutionary model, that doesn't require the approximations of the combined HMM and jump model. On the other hand, PINNACLE can be generalised upon by simultaneously modelling dependencies between base-paired nucleotide sites and those between the triplet of nucleotides at each codon site. Both base-pairing and codon dependencies are expected to impact substitution patterns, particularly in viral genomes such as HIV, which code for protein and are known to contain evolutionary conserved secondary structures. However, the scope for such context-dependent models is much wider, with applications in modelling of antibody evolution (Hoehn *et al.*, 2017), CpG evolutionary modelling (Lunter and Hein, 2004), and implementing evolutionary models that obtain a target stationary distribution over sequences (Choi *et al.*, 2008), amongst many other applications.

Bibliography

Acland, A., Agarwala, R., Barrett, T., Beck, J., Benson, D. A., Bollin, C., Bolton, E., Bryant, S. H., Canese, K., Church, D. M., *et al.* 2014. Database resources of the national center for biotechnology information. *Nucleic acids research*, 42(Database issue): D7.

Anderson, J. 2014. Stochastic Context-Free Grammars and RNA Secondary Structure Prediction. In M. S. Poptsova, editor, *Genome Analysis: Current Procedures and Applications*, pages 339–66. Caister Academic Press: England.

Anderson, J. W., Tataru, P., Staines, J., Hein, J., and Lyngsø, R. 2012. Evolving stochastic context-free grammars for RNA secondary structure prediction. *BMC bioinformatics*, 13(1): 78.

Arnold, K., Bordoli, L., Kopp, J., and Schwede, T. 2006. The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2): 195–201.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., and Nielsen, H. 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5): 412–424.

Banerjee, A. R., Jaeger, J. A., and Turner, D. H. 1993. Thermal unfolding of a group I ribozyme: the low-temperature transition is primarily disruption of tertiary structure. *Biochemistry*, 32(1): 153–163.

Bernhart, S. H., Tafer, H., Mückstein, U., Flamm, C., Stadler, P. F., and Hofacker,

I. L. 2006. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms for Molecular Biology*, 1(1): 3.

Boomsma, W., Mardia, K. V., Taylor, C. C., Ferkinghoff-Borg, J., Krogh, A., and Hamelryck, T. 2008. A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences*, 105(26): 8932–8937.

Boomsma, W., Tian, P., Frellsen, J., Ferkinghoff-Borg, J., Hamelryck, T., Lindorff-Larsen, K., and Vendruscolo, M. 2014. Equilibrium simulations of proteins using molecular fragment replacement and NMR chemical shifts. *Proceedings of the National Academy of Sciences*, 111(38): 13852–13857.

Burge, S. W., Daub, J., Eberhardt, R., Tate, J., Barquist, L., Nawrocki, E. P., Eddy, S. R., Gardner, P. P., and Bateman, A. 2012. Rfam 11.0: 10 years of RNA families. *Nucleic acids research*, 41(D1): D226–D232.

Cavalli-Sforza, L. L. and Edwards, A. W. 1967. Phylogenetic analysis: models and estimation procedures. *Evolution*, 21(3): 550–570.

Challis, C. J. and Schmidler, S. C. 2012. A stochastic evolutionary model for protein structure alignment and phylogeny. *Molecular biology and evolution*, 29(11): 3575–3587.

Chamorro, M., Parkin, N., and Varmus, H. E. 1992. An RNA pseudoknot and an optimal heptameric shift site are required for highly efficient ribosomal frameshifting on a retroviral messenger RNA. *Proceedings of the National Academy of Sciences*, 89(2): 713–717.

Cheng, N., Mao, Y., Shi, Y., and Tao, S. 2012. Coevolution in RNA molecules driven by selective constraints: evidence from 5S rRNA. *PloS one*, 7(9): e44376.

Chib, S. and Greenberg, E. 1995. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4): 327–335.

Choi, S. C., Redelings, B. D., and Thorne, J. L. 2008. Basing population genetic inferences and models of molecular evolution upon desired stationary distributions of DNA or protein sequences. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1512): 3931–3939.

Damgaard, C. K., Andersen, E. S., Knudsen, B., Gorodkin, J., and Kjems, J. 2004. RNA interactions in the 5 region of the HIV-1 genome. *Journal of molecular biology*, 336(2): 369–379.

Darty, K., Denise, A., and Ponty, Y. 2009. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15): 1974.

Daugherty, M. D., Liu, B., and Frankel, A. D. 2010. Structural basis for cooperative RNA binding and export complex assembly by HIV Rev. *Nature structural & molecular biology*, 17(11): 1337–1342.

Deigan, K. E., Li, T. W., Mathews, D. H., and Weeks, K. M. 2009. Accurate SHAPE-directed RNA structure determination. *Proceedings of the National Academy of Sciences*, 106(1): 97–102.

Diaconis, P. and Zabell, S. 1982. Updating subjective probability. *J. American Stat. Assoc.*, 77: 822–830.

Downs, T. D. and Mardia, K. 2002. Circular regression. *Biometrika*, 89(3): 683–698.

Early, T. A., Olmsted III, J., Kearns, D. R., and Lezius, A. G. 1978. Base pairing structure in the poly d (GT) double helix: wobble base pairs. *Nucleic acids research*, 5(6): 1955–1970.

Echave, J. 2008. Evolutionary divergence of protein structure: the linearly forced elastic network model. *Chemical physics letters*, 457(4): 413–416.

Echave, J. and Fernández, F. M. 2010. A perturbative view of protein structural variation. *Proteins: Structure, Function, and Bioinformatics*, 78(1): 173–180.

Eddy, S. R. and Durbin, R. 1994. RNA sequence analysis using covariance models. *Nucleic acids research*, 22(11): 2079–2088.

Edgar, R. C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5): 1792–1797.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17(6): 368–376.

Felsenstein, J. 1985. Phylogenies and the comparative method. *American Naturalist*, 125(1): 1–15.

Frellsen, J., Mardia, K. V., Borg, M., Ferkinghoff-Borg, J., and Hamelryck, T. 2012. Towards a general probabilistic model of protein structure: the reference ratio method. In *Bayesian methods in structural bioinformatics*, pages 125–134. Springer.

Frühwirth-Schnatter, S. 1994. Data augmentation and dynamic linear models. *Journal of time series analysis*, 15(2): 183–202.

García-Portugués, E., Sørensen, M., Mardia, K. V., and Hamelryck, T. 2017. Langevin diffusions on the torus: estimation and applications. *Statistics and Computing*.

Gardner, P. P. and Giegerich, R. 2004. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC bioinformatics*, 5(1): 140.

Gautheret, D., Konings, D., and Gutell, R. R. 1995. GU base pairing motifs in ribosomal RNA. *Rna*, 1(8): 807–814.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. 1995. *Markov chain Monte Carlo in practice*. CRC press.

Golden, M. and Martin, D. 2012. DOOSS: a tool for visual analysis of data overlaid on secondary structures. *Bioinformatics*, 29(2): 271–272.

Golden, M., Muhire, B. M., Semegni, Y., and Martin, D. P. 2014. Patterns of recombination in HIV-1M are influenced by selection disfavouring the survival of recombinants with disrupted genomic RNA and protein structures. *PloS one*, 9(6): e100400.

Golden, M., García-Portugués, E., Sørensen, M., Mardia, K. V., Hamelryck, T., and Hein, J. 2017. A generative angular model of protein structure evolution. *Molecular Biology and Evolution*, page msx137.

Grishin, N. V. 1997. Estimation of evolutionary distances from protein spatial structures. *Journal of molecular evolution*, 45(4): 359–369.

Grishin, N. V. 2001. Fold change in evolution of protein structures. *Journal of structural biology*, 134(2): 167–185.

Gutin, A. M. and Badretdinov, A. Y. 1994. Evolution of protein 3D structures as diffusion in multidimensional conformational space. *Journal of molecular evolution*, 39(2): 206–209.

Halpern, A. L. and Bruno, W. J. 1998. Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies. *Molecular biology and evolution*, 15(7): 910–917.

Hamelryck, T. and Manderick, B. 2003. PDB file parser and structure class implemented in Python. *Bioinformatics*, 19(17): 2308–2310.

Hamelryck, T., Borg, M., Paluszewski, M., Paulsen, J., Frellsen, J., Andreetta, C., Boomsma, W., Bottaro, S., and Ferkinghoff-Borg, J. 2010. Potentials of mean force for protein structure prediction vindicated, formalized and generalized. *PloS one*, 5(11): e13714.

Hamelryck, T., Mardia, K., and Ferkinghoff-Borg, J., editors 2012. *Bayesian methods in structural bioinformatics*. Springer.

Heaphy, S., Dingwall, C., Ernberg, I., Gait, M. J., Green, S. M., Kern, J., Lowe, A. D., Singh, M., and Skinner, M. A. 1990. HIV-1 regulator of virion expression (Rev) protein binds to an RNA stem-loop structure located within the Rev response element region. *Cell*, 60(4): 685–693.

Herman, J. L., Challis, C. J., Novák, Á., Hein, J., and Schmidler, S. C. 2014. Simultaneous Bayesian estimation of alignment and phylogeny under a joint model of protein sequence and structure. *Molecular biology and evolution*, 31(9): 2251–2266.

Ho, P., Frederick, C., Quigley, G., Van der Marel, G., Van Boom, J., Wang, A., and Rich, A. 1985. GT wobble base-pairing in Z-DNA at 1.0 A atomic resolution: the crystal structure of d (CGCGTG). *The EMBO journal*, 4(13A): 3617.

Hoehn, K. B., Lunter, G., and Pybus, O. G. 2017. A phylogenetic codon substitution model for antibody lineages. *Genetics*, 206(1): 417–427.

Hofacker, I. L. 2009. RNA secondary structure analysis using the Vienna RNA package. *Current Protocols in Bioinformatics*, pages 12–2.

Hofacker, I. L., Schuster, P., and Stadler, P. F. 1998. Combinatorics of RNA secondary structures. *Discrete Applied Mathematics*, 88(1-3): 207–237.

Holm, L. and Rosenström, P. 2010. Dali server: conservation mapping in 3D. *Nucleic acids research*, page gkq366.

Holmes, I. 2004. A probabilistic model for the evolution of RNA structure. *Bmc Bioinformatics*, 5(1): 166.

Holmes, I. and Bruno, W. J. 2001. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, 17(9): 803–820.

Johnson, S. G. 2014. The NLopt nonlinear-optimization package.

Joosten, R. P., Te Beek, T. A., Krieger, E., Hekkelman, M. L., Hooft, R. W., Schneider, R., Sander, C., and Vriend, G. 2011. A series of PDB related databases for everyday needs. *Nucleic acids research*, 39(suppl 1): D411–D419.

Joyner, D., Čertík, O., Meurer, A., and Granger, B. E. 2012. Open source computer algebra systems: SymPy. *ACM Communications in Computer Algebra*, 45(3/4): 225–234.

Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, 30(14): 3059–3066.

Knudsen, B. and Hein, J. 1999. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6): 446–454.

Koshi, J. M. and Goldstein, R. A. 1998. Models of Natural Mutations Including Site Heterogeneity. *Proteins: Structure, Function, and Genetics*, 32: 289–295.

Lari, K. and Young, S. J. 1991. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 5(3): 237–257.

Lartillot, N. and Philippe, H. 2004. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Molecular biology and evolution*, 21(6): 1095–1109.

Lindahl, T. and Nyberg, B. 1974. Heat-induced deamination of cytosine residues in deoxyribonucleic acid. *Biochemistry*, 13(16): 3405–3410.

Lindgreen, S., Gardner, P. P., and Krogh, A. 2006. Measuring covariation in RNA alignments: physical realism improves information measures. *Bioinformatics*, 22(24): 2988–2995.

Liò, P., Goldman, N., Thorne, J. L., and Jones, D. T. 1998. PASSML: combining evolutionary inference and protein secondary structure prediction. *Bioinformatics*, 14(8): 726–733.

Lu, Z. J., Gloor, J. W., and Mathews, D. H. 2009. Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*, 15(10): 1805–1813.

Lukavsky, P. J., Otto, G. A., Lancaster, A. M., Sarnow, P., and Puglisi, J. D. 2000. Structures of two RNA domains essential for hepatitis C virus internal ribosome entry site function. *Nature Structural & Molecular Biology*, 7(12): 1105–1110.

Lunter, G. and Hein, J. 2004. A nucleotide substitution model with nearest-neighbour interactions. *Bioinformatics*, 20(suppl 1): i216–i223.

Mandal, M. and Breaker, R. R. 2004. Gene regulation by riboswitches. *Nature Reviews Molecular Cell Biology*, 5(6): 451–463.

Markham, N. R. and Zuker, M. 2008. UNAFold. In *Bioinformatics*, pages 3–31. Springer.

Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. 1999. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of molecular biology*, 288(5): 911–940.

Mattick, J. S. 2003. Challenging the dogma: the hidden layer of non-protein-coding RNAs in complex organisms. *Bioessays*, 25(10): 930–939.

Mauger, D. M., Golden, M., Yamane, D., Williford, S., Lemon, S. M., Martin, D. P., and Weeks, K. M. 2015. Functionally conserved architecture of hepatitis C virus RNA genomes. *Proceedings of the National Academy of Sciences*, 112(12): 3692–3697.

Miklós, I., Lunter, G., and Holmes, I. 2004. A "long indel" model for evolutionary sequence alignment. *Molecular Biology and Evolution*, 21(3): 529–540.

Mizuguchi, K., Deane, C. M., Blundell, T. L., and Overington, J. P. 1998. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein science*, 7(11): 2469–2471.

Moler, C. and Van Loan, C. 2003. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1): 3–49.

Monjane, A. L., Harkins, G. W., Martin, D. P., Lemey, P., Lefeuvre, P., Shepherd, D. N., Oluwafemi, S., Simuyandi, M., Zinga, I., Komba, E. K., *et al.* 2011. Reconstructing the history of maize streak virus strain a dispersal to reveal diversification hot spots and its origin in southern Africa. *Journal of virology*, 85(18): 9623–9636.

Morrone, J. A., Perez, A., MacCallum, J., and Dill, K. A. 2017. Computed binding of peptides to proteins with MELD-accelerated molecular dynamics. *Journal of chemical theory and computation*, 13(2): 870–876.

Moulton, V., Zuker, M., Steel, M., Pointon, R., and Penny, D. 2000. Metrics on RNA secondary structures. *Journal of Computational Biology*, 7(1-2): 277–292.

Muhire, B. M., Golden, M., Murrell, B., Lefeuvre, P., Lett, J.-M., Gray, A., Poon, A. Y., Ngandu, N. K., Semegni, Y., Tanov, E. P., *et al.* 2014. Evidence of pervasive biologically functional secondary structures within the genomes of eukaryotic single-stranded DNA viruses. *Journal of virology*, 88(4): 1972–1989.

Murrell, B., Wertheim, J. O., Moola, S., Weighill, T., Scheffler, K., and Pond, S. L. K. 2012. Detecting individual sites subject to episodic diversifying selection. *PLoS genetics*, 8(7): e1002764.

Murrell, B., Moola, S., Mabona, A., Weighill, T., Sheward, D., Pond, S. L. K., and Scheffler, K. 2013. FUBAR: a fast, unconstrained bayesian approximation for inferring selection. *Molecular biology and evolution*, page mst030.

Muse, S. V. 1995. Evolutionary analyses of DNA sequences subject to constraints of secondary structure. *Genetics*, 139(3): 1429–1439.

Muse, S. V. and Gaut, B. S. 1994. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular Biology and Evolution*, 11(5): 715–724.

Nebel, M. E. and Schulz, A. 2014. Context-free Grammars and RNA Secondary Structure Prediction. *Genome Analysis: Current Procedures and Applications*, page 307.

Nodelman, U., Shelton, C. R., and Koller, D. 2002. Continuous time Bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 378–387. Morgan Kaufmann Publishers Inc.

Ortiz, A. R., Strauss, C. E., and Olmea, O. 2002. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Science*, 11(11): 2606–2621.

Pollom, E., Dang, K. K., Potter, E. L., Gorelick, R. J., Burch, C. L., Weeks, K. M., and Swanstrom, R. 2013. Comparison of SIV and HIV-1 genomic RNA structures reveals impact of sequence evolution on conserved and non-conserved structural motifs. *PLoS pathogens*, 9(4): e1003294.

Pond, S. L. K. and Muse, S. V. 2004. Column sorting: Rapid calculation of the phylogenetic likelihood function. *Systematic biology*, 53(5): 685–692.

Powell, M. J. D. 1994. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pages 51–67. Springer Netherlands.

Price, M. N., Dehal, P. S., and Arkin, A. P. 2010. FastTree 2–approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3): e9490.

Rao, V. and Teh, Y. W. 2013. Fast MCMC sampling for Markov jump processes and extensions. *The Journal of Machine Learning Research*, 14(1): 3295–3320.

Roberts, G. O. and Rosenthal, J. S. 2009. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2): 349–367.

Robinson, D. M., Jones, D. T., Kishino, H., Goldman, N., and Thorne, J. L. 2003. Protein evolution with dependence among codons due to tertiary structure. *Molecular Biology and Evolution*, 20(10): 1692–1704.

Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. 2004. Protein structure prediction using Rosetta. *Methods in enzymology*, 383: 66–93.

Rousset, F., Pelandakis, M., and Solignac, M. 1991. Evolution of compensatory substitutions through GU intermediate state in Drosophila rRNA. *Proceedings of the National Academy of Sciences*, 88(22): 10032–10036.

Roy, S., Delling, U., Chen, C., Rosen, C., and Sonenberg, N. 1990. A bulge structure in HIV-1 TAR RNA is required for Tat binding and Tat-mediated trans-activation. *Genes & development*, 4(8): 1365–1373.

Ruscio, J. 2008. Constructing confidence intervals for Spearman's rank correlation with ordinal data: A simulation study comparing analytic and bootstrap methods. *Journal of Modern Applied Statistical Methods*, 7(2): 7.

Rzhetsky, A. and Nei, M. 1993. Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Molecular biology and evolution*, 10(5): 1073–1095.

Satija, R., Novák, Á., Miklós, I., Lyngsø, R., and Hein, J. 2009. BigFoot: Bayesian alignment and phylogenetic footprinting with MCMC. *BMC evolutionary biology*, 9(1): 217.

Schwartz, A. S., Myers, E. W., and Pachter, L. 2005. Alignment metric accuracy. *arXiv preprint q-bio/0510052*.

Shindyalov, I. N. and Bourne, P. E. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein engineering*, 11(9): 739–747.

Siddharthan, R., Siggia, E. D., and Van Nimwegen, E. 2005. PhyloGibbs: a Gibbs sampling motif finder that incorporates phylogeny. *PLoS computational biology*, 1(7): e67.

Siegfried, N. A., Busan, S., Rice, G. M., Nelson, J. A., and Weeks, K. M. 2014. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nature methods*, 11(9): 959–965.

Siepel, A. and Haussler, D. 2004. Combining phylogenetic and hidden Markov models in biosequence analysis. *Journal of Computational Biology*, 11(2-3): 413–428.

Spearman, C. 1904. The proof and measurement of association between two things. *The American journal of psychology*, 15(1): 72–101.

Sükösd, Z., Knudsen, B., Værum, M., Kjems, J., and Andersen, E. S. 2011. Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars. *BMC bioinformatics*, 12(1): 103.

Sükösd, Z., Knudsen, B., Kjems, J., and Pedersen, C. N. 2012. PPfold 3.0: fast RNA secondary structure prediction using phylogeny and auxiliary data. *Bioinformatics*, 28(20): 2691–2692.

Sükösd, Z., Knudsen, B., Anderson, J. W., Novák, Á., Kjems, J., and Pedersen, C. N. 2013. Characterising RNA secondary structure space using information entropy. *BMC bioinformatics*, 14(2): S22.

Tavaré, S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on mathematics in the life sciences*, 17: 57–86.

The Jmol team 2013. An open-source Java viewer for chemical structures in 3D. *Jmol web page: http://www.jmol.org/*, 15.

Thorne, J. L., Kishino, H., and Felsenstein, J. 1992. Inching toward reality: an improved likelihood model of sequence evolution. *Journal of molecular evolution*, 34(1): 3–16.

Tinoco Jr, I. and Bustamante, C. 1999. How RNA folds. *Journal of molecular biology*, 293(2): 271–281.

Touw, W. G., Baakman, C., Black, J., te Beek, T. A., Krieger, E., Joosten, R. P., and Vriend, G. 2015. A series of PDB-related databanks for everyday needs. *Nucleic acids research*, 43(D1): D364–D368.

Tuplin, A., Wood, J., Evans, D. J., Patel, A. H., and Simmonds, P. 2002. Thermodynamic and phylogenetic prediction of RNA secondary structures in the coding region of hepatitis C virus. *Rna*, 8(6): 824–841.

Tuplin, A., Evans, D., and Simmonds, P. 2004. Detailed mapping of RNA secondary structures in core and NS5B-encoding region sequences of hepatitis C virus by RNase cleavage and novel bioinformatic prediction methods. *Journal of General Virology*, 85(10): 3037–3047.

UniProt Consortium and others 2017. UniProt: the universal protein knowledgebase. *Nucleic acids research*, 45(D1): D158–D169.

Valentin, J. B., Andreetta, C., Boomsma, W., Bottaro, S., Ferkinghoff-Borg, J., Frellsen, J., Mardia, K. V., Tian, P., and Hamelryck, T. 2014. Formulation of probabilistic models of protein structure in atomic detail using the reference ratio method. *Proteins: Structure, Function, and Bioinformatics*, 82(2): 288–299.

Varani, G. and McClain, W. H. 2000. The G·U wobble base pair. *EMBO reports*, 1(1): 18–23.

Watson, J. D. and Crick, F. H. 1953. Molecular structure of nucleic acids. *Nature*, 171(4356): 737–738.

Watts, J. M., Dang, K. K., Gorelick, R. J., Leonard, C. W., Bess Jr, J. W., Swanstrom, R., Burch, C. L., and Weeks, K. M. 2009. Architecture and secondary structure of an entire HIV-1 RNA genome. *Nature*, 460(7256): 711–716.

Whelan, S. and Goldman, N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular biology and evolution*, 18(5): 691–699.

Wilkinson, K. A., Merino, E. J., and Weeks, K. M. 2006. Selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE): quantitative RNA structure analysis at single nucleotide resolution. *Nature protocols*, 1(3): 1610–1616.

Wilson, C., Agafonov, R., Hoemberger, M., Kutter, S., Zorba, A., Halpin, J., Buosi, V., Otten, R., Waterman, D., Theobald, D., *et al.* 2015. Using ancient protein kinases to unravel a modern cancer drug's mechanism. *Science*, 347(6224): 882–886.

Woese, C., Magrum, L., Gupta, R., Siegel, R., Stahl, D., Kop, J., Crawford, N., Brosius, R., Gutell, R., Hogan, J., *et al.* 1980. Secondary structure model for bacterial 16S ribosomal RNA: phylogenetic, enzymatic and chemical evidence. *Nucleic acids research*, 8(10): 2275–2294.

Woese, C. R. and Fox, G. E. 1977. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences*, 74(11): 5088–5090.

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

Yang, Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular evolution*, 39(3): 306–314.

Yang, Z. 1996. Among-site rate variation and its impact on phylogenetic analyses. *Trends in Ecology & Evolution*, 11(9): 367–372.

Yang, Z. 2014. *Molecular Evolution: A Statistical Approach*. Oxford University Press.

Yu, J. and Thorne, J. L. 2006. Dependence among sites in RNA evolution. *Molecular biology and evolution*, 23(8): 1525–1537.

Zhang, Y. and Skolnick, J. 2005. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic acids research*, 33(7): 2302–2309.

Appendix A

PINNACLE

Table A.1: List of dataset short and long names

Short name	Full name
BCTV	Beet Curly Top Virus
FMDV	Foot-and-mouth-disease Virus
HIV	Human Immunodeficiency Virus
HPV1	Human Poliovirus 1
MSV	Maise Streak Virus
SIV	Simian Immunodeficiency Virus
TYCLV	Tomato Yellow Leaf Curl Virus
WDV	Wheat Dwarf Virus

```
from sympy import *

""" Construct Q matrix """
rates = [Symbol('r1'), Symbol('r2'), Symbol('r3'),
         Symbol('r4'), Symbol('r5'), Symbol('r6')]
freqs = [Symbol('piA'), Symbol('piC'), Symbol('piG'), Symbol('piT')]
Qfreqs = Matrix([freqs[0], freqs[1], freqs[2], freqs[3]])
Q = zeros(4,4)
index = 0
for i in xrange(4):
    for j in xrange(i+1,4):
        if i != j:
            Q[i,j] = freqs[j]*rates[index]
            Q[j,i] = freqs[i]*rates[index]
            index += 1
            if index == 6:
                break
    if index == 6:
            break

""" Calculate Q matrix diagonal entries """
for i in xrange(4):
    Q[i,i]  = 0.0
    for j in xrange(4):
        if i != j:
            Q[i,i] -= Q[i,j]

""" Construct Muse matrix """
Muse = zeros(16,16)
Musefreqs = Matrix([0.0 for i in xrange(16)]) # dinucleotide frequencies

lambdaGC = Symbol('lambdaGC')
lambdaAU = Symbol('lambdaAU')
lambdaGU = Symbol('lambdaGU')
GCindices = [2*4+1,1*4+2] # GC, CG
```

```
AUindices = [0*4+3,3*4+0] # AU, UA
GUindices = [2*4+3,3*4+2] # GU, UG

index = 0
kappa = 0 # dinucleotide frequencies normalising constant
for i in xrange(0,4):
    for j in xrange(0,4):
        Musefreqs[index] = Qfreqs[i]*Qfreqs[j]
        dinuc = i*4+j
        if dinuc in GCindices:
            Musefreqs[index] = lambdaGC*lambdaGC*Musefreqs[index]
        if dinuc in AUindices:
            Musefreqs[index] = lambdaAU*lambdaAU*Musefreqs[index]
        if dinuc in GUindices:
            Musefreqs[index] = lambdaGU*lambdaGU*Musefreqs[index]
        kappa += Musefreqs[index]
        index += 1

rho1 = Symbol('rho1') # rate for first position
rho2 = Symbol('rho2')  # rate for second position
Musefreqs /= kappa
for i in xrange(16):
    for j in xrange(16):
        if i != j:
            s1 = i / 4
            s2 = i % 4
            e1 = j / 4
            e2 = j % 4
            if s1 != e1 and s2 != e2:
                Muse[i,j] = 0.0
            else:
                if s1 != e1:
                    Muse[i,j] = rho1*Q[s1,e1]
                if s2 != e2:
                    Muse[i,j] = rho2*Q[s2,e2]
```

```
71
72              if j in GCindices:
73                  Muse[i,j] = lambdaGC*Muse[i,j]
74              if j in AUindices:
75                  Muse[i,j] = lambdaAU*Muse[i,j]
76              if j in GUindices:
77                  Muse[i,j] = lambdaGU*Muse[i,j]
78              if i in GCindices:
79                  Muse[i,j] = Muse[i,j]/lambdaGC
80              if i in AUindices:
81                  Muse[i,j] = Muse[i,j]/lambdaAU
82              if i in GUindices:
83                  Muse[i,j] = Muse[i,j]/lambdaGU
84
85  """ Calculate Muse matrix diagonal entries """
86  for i in xrange(0,16):
87      Muse[i,i]  = 0.0
88      for j in xrange(0,16):
89          if i != j:
90              Muse[i,i] -= Muse[i,j]
91
92  v = simplify(transpose(Musefreqs)*Muse)
93  print "Stationarity (pi*M = 0)?", (sum([q == 0 for q in v]) == 16)
94
95  timereversible = True
96  for i in xrange(0,16):
97      for j in xrange(0,16):
98          v = simplify(Musefreqs[i]*Muse[i,j] - Musefreqs[j]*Muse[j,i])
99          if v != 0:
100             timereversible = False
101         #print "pi%d*M%d,%d = pi%d*M%d,%d" % (i,i,j,j,j,i), v == 0
102 print "Time-reversible (pi_i*M_ij = pi_j*M_ji)?", timereversible
103
104 """
105 Output:
```

```
Stationarity (pi*M = 0)? True
Time-reversible (pi_i*M_ij = pi_j*M_ji)? True
"""
```

Listing A.1: Python script (`musesymbolic.py`) testing stationarity and time-reversibility of the extended Muse model.

```
function samplestructurehelper(rng::AbstractRNG, inside::Array{Float64
    ,3}, pairedlogprobs::Array{Float64,2}, unpairedlogprobs::Array{
    Float64,1}, parentsymbol::Char, x::Int, y::Int, paired::Array{Int
    ,1}, grammar::KH99, B::Float64)
  type1rules = grammar.type1rules
  type2rules = grammar.type2rules
  type3rules = grammar.type3rules
  stack = Tuple{Char,Int,Int}[]
  push!(stack, (parentsymbol, x, y))
  while length(stack) > 0
    parentsymbol, x, y = pop!(stack)
    rulearr = Rule[]
    if x == y  # unpaired
    elseif x < y
      j = x
      b = y-j+1

      logliks = Float64[]
      rule = Tuple[]

      for type3rule in type3rules  # bifurcations / branching
        if type3rule.left == parentsymbol
          for h=j:j+b-2
            prob1 = inside[type3rule.rightindices[1],j,h]
            prob2 = inside[type3rule.rightindices[2],h+1,j+b-1]
            push!(logliks, type3rule.logprob + prob1 + prob2)
            push!(rule, (3,j,h,h+1,j+b-1,type3rule.right[1],type3rule.
  right[2]))
```

```
25            push!(rulearr, type3rule)
26          end
27        end
28      end
29
30      for type2rule in type2rules # base-pairings
31        if type2rule.left == parentsymbol
32          push!(logliks, (type2rule.logprob + pairedlogprobs[j,j+b-1]) +
 inside[type2rule.rightindices[2],j+1,j+b-2])
33          push!(rule, (2,j,j+b-1,type2rule.right[2]))
34          push!(rulearr, type2rule)
35        end
36      end
37
38      logliks *= B
39      s = -Inf
40      for v in logliks
41        s = logsumexp(s,v)
42      end
43      r = sample(rng, exp.(logliks-s)) # sample rule used to generate
bifurcation or base-pairing
44
45      if rule[r][1] == 2 # base-pairing, recursively sample the sub-
alignment
46        a = rule[r][2]
47        b = rule[r][3]
48        paired[a] = b # store pairing
49        paired[b] = a # store pairing
50        push!(stack, (rule[r][4], a+1, b-1))
51      elseif rule[r][1] == 3 # bifurcation, recursively sample the two
sub-alignments
52        push!(stack, (rule[r][6], rule[r][2], rule[r][3]))
53        push!(stack, (rule[r][7], rule[r][4], rule[r][5]))
54      end
55    end
```

```
56    end
57
58    return calculateKH99prior(paired)
59  end
```

Listing A.2: Julia recursion (`samplestructurehelper`) for sampling secondary structures from the inside probabilities, phylogenetic unpaired probabilities, and phylogenetic paired probabilities.

```
function logsumexp(a::Float64, b::Float64)
    d = a-b
    if b == -Inf
      return a
    elseif a == -Inf
        return b
    elseif a < b
      return b + log1p(exp(a-b))
    else
      return a + log1p(exp(b-a))
    end
end

function logcountstructures(n::Int, mindist::Int=2, cache::Dict{Int,
    Float64}=Dict{Int,Float64}())
  for j=1:n
    count = logS(j,mindist,cache)
  end
  return cache[n]
end

function logS(n::Int, mindist::Int=2, cache::Dict{Int,Float64}=Dict{Int,
    Float64}())
  if haskey(cache,n)
    return cache[n]
  else
    v = 0.0
    if n >= mindist + 1
      v = logS(n-1,mindist,cache)
      for k=1:n-(mindist+1)
          v = logsumexp(v, logS(k-1,mindist, cache) + logS(n-k-1,mindist
    , cache))
      end
    end
    cache[n] = v
```

```
33      return v
34    end
35  end
36
37  function countstructures(n::Int, mindist::Int=2, cache::Dict{Int,BigInt
      }=Dict{Int,BigInt}())
38    for j=1:n
39      count = S(j,mindist,cache)
40    end
41    return cache[n]
42  end
43
44  function S(n::Int, mindist::Int=2, cache::Dict{Int,BigInt}=Dict{Int,
      BigInt}())
45    if haskey(cache,n)
46      return cache[n]
47    else
48      v = BigInt(1)
49      if n >= mindist + 1
50        v = S(n-1,mindist,cache)
51        for k=1:n-(mindist+1)
52            v += S(k-1,mindist, cache)*S(n-k-1,mindist, cache)
53        end
54      end
55      cache[n] = v
56      return v
57    end
58  end
```

Listing A.3: Julia code for efficient counting of secondary structures.

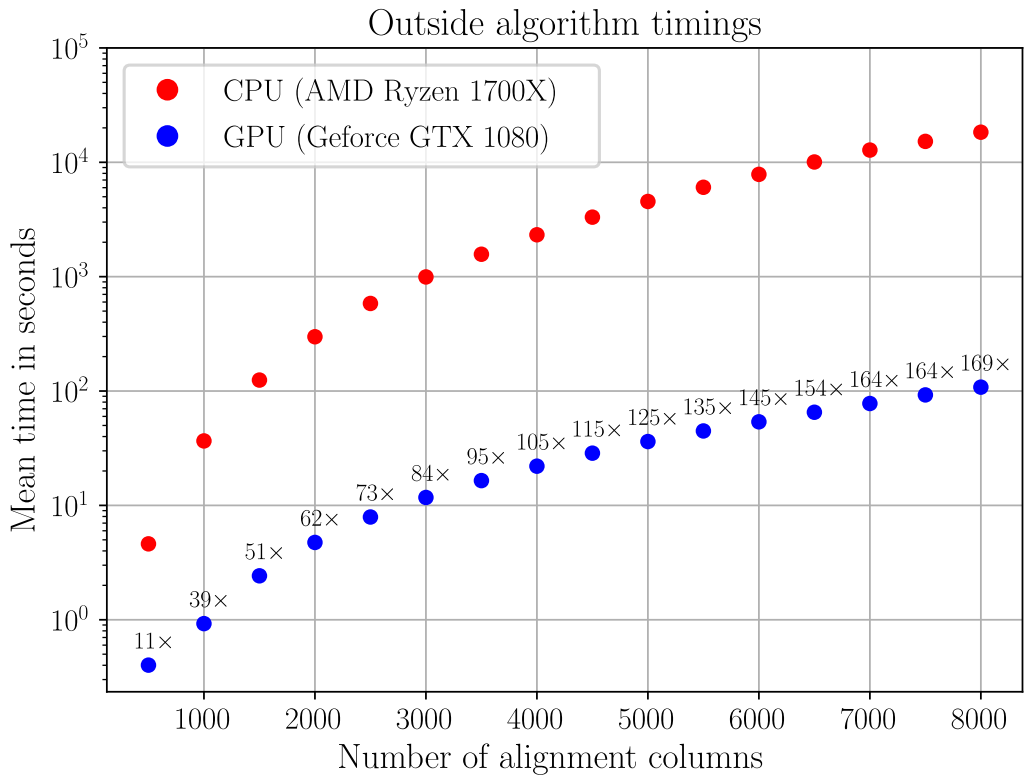


Figure A.1: Outside algorithm timings in seconds ($\log_{10}$ axis) as a function of the number of alignment columns, with the fold speed-up of the GPU version over the CPU version indicated above the GPU timings.

Table A.2: Secondary structure prediction benchmarks of 22 RFAM datasets.

Dataset	Method	Precision	Recall	F-score	MCC	Weighted mountain similarity	Mountain similarity (p=1)
	Our method	0.871	**0.670**	**0.758**	**0.781**	**0.856**	**0.936**
RF01854	PPfold	0.833	0.275	0.413	0.524	0.611	0.687
	RNAalifold	**1.000**	0.319	0.483	0.564	0.605	0.692
	Our method	0.856	**0.975**	**0.911**	**0.987**	**0.958**	0.961
RF01846	PPfold	0.889	0.608	0.722	0.779	0.907	**0.978**
	RNAalifold	**0.943**	0.418	0.579	0.646	0.857	0.967
	Our method	0.792	**0.760**	**0.776**	**0.872**	**0.907**	**0.991**
RF00002	PPfold	0.679	0.760	0.717	0.872	0.898	0.980
	RNAalifold	**1.000**	0.480	0.649	0.693	0.874	0.952
	Our method	0.872	**0.854**	0.863	**0.924**	**0.966**	**0.963**
RF02001	PPfold	0.930	0.833	**0.879**	0.913	0.962	0.947
	RNAalifold	**0.968**	0.625	0.759	0.790	0.907	0.914

Dataset	Method	Precision	Recall	F-score	MCC	Weighted mountain similarity	Mountain similarity (p=1)
RF00379	Our method	0.424	**0.875**	0.571	0.825	0.898	0.868
	PPfold	**0.519**	0.875	**0.651**	**0.825**	**0.934**	0.893
	RNAalifold	0.519	0.875	0.651	0.825	0.934	**0.904**
RF00380	Our method	0.469	0.920	0.622	0.693	0.826	0.742
	PPfold	0.533	**0.960**	0.686	**0.732**	0.855	0.774
	RNAalifold	**0.548**	0.920	**0.687**	0.718	**0.877**	**0.789**
RF00012	Our method	0.886	**0.984**	**0.932**	**0.992**	**0.955**	**0.982**
	PPfold	0.784	0.635	0.702	0.797	0.890	0.977
	RNAalifold	**1.000**	0.603	0.752	0.776	0.841	0.932
RF00100	Our method	0.902	0.632	0.743	0.795	0.854	0.964
	PPfold	0.903	**0.644**	**0.752**	**0.802**	**0.862**	**0.969**
	RNAalifold	**0.921**	0.402	0.560	0.634	0.744	0.933

Dataset	Method	Precision	Recall	F-score	MCC	Weighted mountain similarity	Mountain similarity (p=1)
RF00020	Our method	0.811	**1.000**	0.896	**1.000**	0.926	0.957
	PPfold	0.879	0.967	**0.921**	0.983	**0.947**	**0.988**
	RNAalifold	**1.000**	0.700	0.824	0.836	0.905	0.980
RF00174	Our method	0.544	0.861	**0.667**	**0.817**	0.903	0.901
	PPfold	0.507	**0.972**	0.667	0.786	0.850	0.801
	RNAalifold	**0.750**	0.583	0.656	0.661	**0.945**	**0.960**
RF00001	Our method	0.707	**0.853**	**0.773**	**0.923**	0.933	0.973
	PPfold	0.714	0.735	0.725	0.857	**0.964**	**0.991**
	RNAalifold	**0.840**	0.618	0.712	0.786	0.910	0.971
RF00004	Our method	0.830	**0.978**	0.898	**0.989**	0.928	0.982
	PPfold	0.913	0.933	0.923	0.966	0.949	**0.996**
	RNAalifold	**1.000**	0.889	**0.941**	0.943	**0.964**	0.994

Dataset	Method	Precision	Recall	F-score	MCC	Weighted mountain similarity	Mountain similarity (p=1)
RF00030	Our method	0.653	**0.721**	**0.685**	**0.849**	**0.984**	0.990
	PPfold	0.642	0.632	0.637	0.795	0.978	**0.991**
	RNAalifold	**0.706**	0.529	0.605	0.728	0.970	0.989
RF02542	Our method	0.641	**0.863**	**0.735**	**0.929**	0.863	0.957
	PPfold	0.637	0.758	0.692	0.871	0.875	0.964
	RNAalifold	**0.698**	0.698	0.698	0.836	**0.889**	**0.970**
RF00026	Our method	0.250	**1.000**	0.400	0.913	0.860	0.976
	PPfold	0.385	1.000	0.556	**1.000**	0.925	0.987
	RNAalifold	**1.000**	1.000	**1.000**	1.000	**1.000**	**1.000**
RF00017	Our method	0.962	**0.850**	**0.903**	**0.922**	**0.902**	**0.938**
	PPfold	0.959	0.783	0.862	0.885	0.883	0.908
	RNAalifold	**0.987**	0.650	0.784	0.806	0.800	0.861

Dataset	Method	Precision	Recall	F-score	MCC	Weighted mountain similarity	Mountain similarity (p=1)
RF00011	Our method	0.664	**0.987**	0.794	**0.974**	0.877	0.909
	PPfold	0.634	0.934	0.755	0.947	0.877	0.918
	RNAalifold	**0.767**	0.908	**0.831**	0.953	**0.948**	**0.945**
RF00010	Our method	0.617	**0.987**	**0.759**	**0.891**	0.909	0.912
	PPfold	0.591	0.907	0.716	0.847	0.918	0.912
	RNAalifold	**0.667**	0.720	0.692	0.740	**0.960**	**0.935**
RF00209	Our method	0.857	0.934	0.894	0.967	**0.950**	**0.993**
	PPfold	0.844	**0.975**	**0.905**	**0.988**	0.933	0.981
	RNAalifold	**0.877**	0.877	0.877	0.936	0.939	0.988
RF02540	Our method	0.650	**0.933**	0.766	**0.958**	0.858	0.971
	PPfold	0.677	0.908	0.775	0.946	0.882	0.973
	RNAalifold	**0.768**	0.890	**0.825**	0.940	**0.927**	**0.982**

Dataset	Method	Precision	Recall	F-score	MCC	Weighted mountain similarity	Mountain similarity (p=1)
RF02541	Our method	0.677	**0.897**	0.772	**0.941**	0.902	0.978
	PPfold	0.707	0.884	0.786	0.934	0.915	0.980
	RNAalifold	**0.803**	0.852	**0.827**	0.917	**0.939**	**0.986**
RF00003	Our method	0.800	**0.900**	0.847	0.949	0.911	0.976
	PPfold	**0.900**	0.900	**0.900**	**0.949**	**0.948**	**0.986**
	RNAalifold	0.897	0.875	0.886	0.935	0.931	0.982

Table A.3: 103 non-overlapping HIV NL4-3 substructures ranked lowest to highest z-score based SHAPE reactivities within them.

Rank	Position	Length	Name and reference	Median	z-score
1	7276 - 7574	299	Rev Response element (RRE, Heaphy *et al.* (1990); Mandal and Breaker (2004))	0.05	-9.88
2	9070 - 9138	69	3' Untranslated region (3'UTR, Siegfried *et al.* (2014))	0.01	-7.82
3	4797 - 4899	103		0.04	-6.90
4	2618 - 2917	300		0.08	-4.99
5	5793 - 6013	221		0.08	-4.15
6	1561 - 1711	151	Gag-pol frameshift (Chamorro *et al.*, 1992)	0.08	-4.04
7	2145 - 2171	27	Longest continuous helix (Siegfried *et al.*, 2014)	0.04	-3.39
8	58 - 104	47	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.08	-3.21
9	4938 - 4999	62		0.06	-3.21
10	4754 - 4770	17		0.06	-2.89
11	2015 - 2121	107		0.10	-2.71
12	1075 - 1101	27		0.06	-2.60
13	2268 - 2310	43		0.09	-2.37
14	2378 - 2429	52		0.07	-2.11

Rank	Position	Length	Name and reference	Median	z-score
15	7705 - 7770	66		0.08	-2.11
16	1810 - 1918	109		0.08	-2.09
17	2 - 56	55	Trans-activation response element (TAR, Roy *et al.* (1990))	-	-1.94
18	3269 - 3290	22		0.05	-1.91
19	105 - 344	240	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.12	-1.87
20	1728 - 1758	31		0.12	-1.79
21	1026 - 1068	43		0.14	-1.42
22	997 - 1014	18		0.06	-1.24
23	2245 - 2260	16		0.08	-1.14
24	8797 - 8917	121		0.13	-1.10
25	1760 - 1785	26		0.08	-0.99
26	3074 - 3089	16		0.09	-0.91
27	3392 - 3427	36		0.15	-0.85
28	7647 - 7692	46		0.09	-0.73
29	4158 - 4175	18		0.13	-0.72
30	398 - 485	88	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.13	-0.68

Rank	Position	Length	Name and reference	Median	z-score
31	1103 - 1141	39		0.14	-0.66
32	6706 - 6767	62		0.16	-0.57
33	9042 - 9057	16		0.07	-0.54
34	6544 - 6589	46		0.11	-0.54
35	3205 - 3223	19		0.12	-0.50
36	8500 - 8783	284		0.16	-0.41
37	6861 - 7155	295		0.15	-0.40
38	8226 - 8268	43		0.16	-0.36
39	8354 - 8377	24		0.12	-0.33
40	1375 - 1387	13		0.18	-0.21
41	501 - 714	214	5' Untranslated region (5'UTR, Siegfried *et al.* (2014))	0.15	-0.15
42	3040 - 3058	19		0.13	0.00
43	4099 - 4129	31		0.23	0.03
44	9059 - 9067	9		0.22	0.10
45	6777 - 6802	26		0.20	0.23
46	6273 - 6287	15		0.21	0.27

Rank	Position	Length	Name and reference	Median	z-score
47	2135 - 2144	10		0.22	0.28
48	795 - 849	55		0.16	0.33
49	6048 - 6066	19		0.16	0.34
50	3242 - 3257	16		0.15	0.38
51	6680 - 6702	23		0.18	0.55
52	759 - 775	17		0.27	0.59
53	871 - 913	43		0.21	0.63
54	921 - 930	10		0.23	0.65
55	8440 - 8454	15		0.36	0.66
56	3178 - 3190	13		0.26	0.68
57	7602 - 7613	12		0.30	0.70
58	4614 - 4647	34		0.20	0.71
59	3092 - 3176	85		0.16	0.74
60	5117 - 5158	42		-	0.91
61	3307 - 3351	45		0.20	0.97
62	3797 - 4058	262		0.17	0.99

Rank	Position	Length	Name and reference	Median	z-score
63	946 - 957	12		0.30	1.06
64	961 - 996	36		0.25	1.10
65	7833 - 7840	8		0.37	1.11
66	8275 - 8348	74		0.22	1.21
67	6616 - 6650	35		0.22	1.31
68	5010 - 5022	13		0.32	1.33
69	2476 - 2532	57		0.21	1.42
70	8461 - 8498	38		0.26	1.45
71	6667 - 6678	12		0.42	1.48
72	7614 - 7645	32		0.24	1.52
73	2972 - 3038	67		0.22	1.54
74	5081 - 5096	16		0.34	1.58
75	2537 - 2586	50		0.24	1.59
76	8982 - 9039	58		0.31	1.59
77	6518 - 6540	23		0.30	1.60
78	1395 - 1559	165		0.19	1.61

Rank	Position	Length	Name and reference	Median	z-score
79	2201 - 2238	38		0.24	1.72
80	8971 - 8980	10		0.26	1.80
81	7896 - 8152	257		-	1.95
82	4068 - 4098	31		0.29	2.02
83	1177 - 1312	136		0.20	2.02
84	4475 - 4491	17		0.28	2.03
85	4573 - 4586	14		0.36	2.04
86	4786 - 4795	10		0.50	2.06
87	8169 - 8205	37		0.29	2.11
88	5167 - 5188	22		0.40	2.31
89	5471 - 5770	300		-	2.34
90	6490 - 6501	12		0.43	2.41
91	4520 - 4529	10		0.62	2.61
92	1971 - 1986	16		0.40	2.64
93	7852 - 7894	43		0.29	2.66
94	4678 - 4722	45		0.26	2.81

Rank	Position	Length	Name and reference	Median	z-score
95	4723 - 4752	30		0.40	2.94
96	6078 - 6096	19		0.44	3.00
97	5192 - 5427	236		0.21	3.28
98	3429 - 3699	271		0.26	3.29
99	6302 - 6488	187		0.24	3.66
100	1316 - 1348	33		0.48	3.70
101	4176 - 4473	298	Central polypurine tract (CPPT, Siegfried *et al.* (2014))	0.28	4.47
102	9141 - 9170	30		-	6.08
103	6133 - 6272	140		0.42	6.42

Appendix B

ETDBN

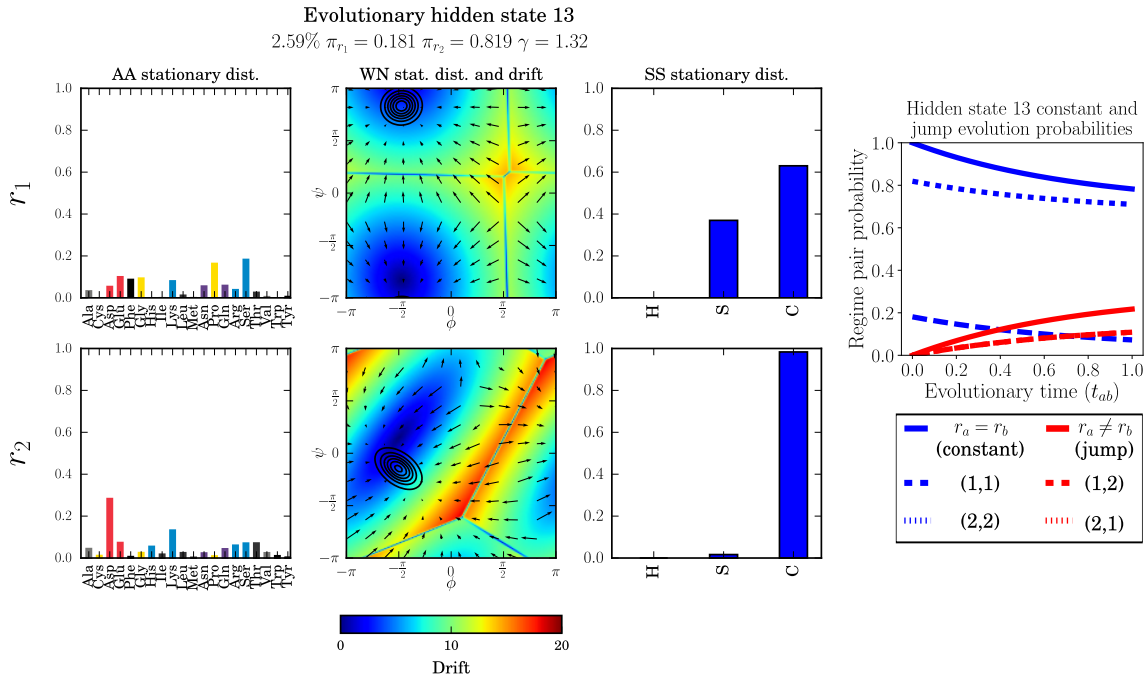


Figure B.1: Depiction of evolutionary hidden state 13. This hidden state was sampled at 2.59% of sites (the average was 1.56%). The equilibrium frequencies of r_1 and r_2 were $\pi_1 = 0.181$ and $\pi_2 = 0.819$, respectively. The jump rate was $\gamma = 1.32$. The corresponding site-class pair probabilities are depicted to the right as a function of evolutionary time. Note that the dashed red lines depicting the probabilities for $(1, 2)$ and $(2, 1)$ superimpose exactly, because the probabilities are equal – this holds for the jump probabilities of all hidden states as it is required for time-reversibility. In the main figure, the two rows depict the parameters encoded by the two site-classes, respectively. Columns 1 and 3 depict the parameters governing the the amino acid and secondary structure stochastic processes, respectively. The secondary structure classes correspond to H=helix, S=sheet and C=coil. Column 2 depicts the WN diffusions. The stationary distributions of the WN diffusions are shown using black contour lines, the direction of the drifts are indicated by the arrows and the magnitude of the drifts at each position indicated using the colour gradient.