Multiple sequence analysis in the presence of alignment uncertainty

Joseph L Herman

St Cross College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2014

Multiple sequence analysis in the presence of alignment uncertainty

Joseph L Herman

St Cross College, University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Hilary 2014

Sequence alignment is one of the most intensely studied problems in bioinformatics, and is an important step in a wide range of analyses. An issue that has gained much attention in recent years is the fact that downstream analyses are often highly sensitive to the specific choice of alignment.

One way to address this is to jointly sample alignments along with other parameters of interest. In order to extend the range of applicability of this approach, the first chapter of this thesis introduces a probabilistic evolutionary model for protein structures on a phylogenetic tree; since protein structures typically diverge much more slowly than sequences, this allows for more reliable detection of remote homologies, improving the accuracy of the resulting alignments and trees, and reducing sensitivity of the results to the choice of dataset. In order to carry out inference under such a model, a number of new Markov chain Monte Carlo approaches are developed, allowing for more efficient convergence and mixing on the high-dimensional parameter space.

The second part of the thesis presents a directed acyclic graph (DAG)-based approach for representing a collection of sampled alignments. This DAG representation allows the initial collection of samples to be used to generate a larger set of alignments under the same approximate distribution, enabling posterior alignment probabilities to be estimated reliably from a reasonable number of samples. If desired, summary alignments can then be generated as maximum-weight paths through the DAG, under various types of loss or scoring functions.

The acyclic nature of the graph also permits various other types of algorithms to be easily adapted to operate on the entire set of alignments in the DAG. In the final part of this work, methodology is introduced for alignment-DAG-based sequence annotation using hidden Markov models, and RNA secondary structure prediction using stochastic context-free grammars. Results on test datasets indicate that the additional information contained within the DAG allows for improved predictions, resulting in substantial gains over simply analysing a set of alignments one by one.

Acknowledgements

The work described in this thesis has benefited greatly from the advice and guidance of a number of people. Particular thanks are due to Jotun Hein, Elspeth Garman, Willie Taylor, Geoff Nicholls, Ádám Novák, Chris Challis, Rune Lyngsø, Steffen Lauritzen, Scott Schmidler, and James Anderson. I would also like to extend thanks to Cyrus Rich, Diana-Elena Gratie and Maria Aştefănoaei for developing software implementing the RNA secondary structure prediction algorithms on alignment DAGs, and to the IT officers at the Department of Statistics, Stuart McRobert, Susan Hutchinson, David del Campo Hill and Saffron Greenwood, for support with maintaining and managing compute servers.

Contents

1 Overview **1**

1.1 Sequence alignment as a ubiquitous problem 1
1.2 Effect of alignment on downstream inference 2
1.3 Quantifying alignment uncertainty 2
1.4 Generating sets of alignments 3
1.5 A Bayesian approach 4
1.6 Filtering methods 5
1.7 Joint sampling approaches 6
1.8 Software engineering for inference with new joint sampling approaches . . 8
1.9 Graph-based approaches for analysing large sets of alignments 9
1.10 Alternatives to joint sampling 10
1.11 Statement of collaboration 11

2 Simultaneous Bayesian estimation of alignment and phylogeny under a joint model of protein sequence and structure **13**

2.1 Probabilistic evolutionary models 14
2.1.1 Sequence and structure data 14
2.1.2 Representation of a multiple alignment 15
2.1.3 Joint model for sequence and structure 15
2.1.4 Marginal posterior 17
2.1.5 Indel model 17
2.1.6 Substitution model 19
2.2 Structural drift model 19
2.2.1 Model specification 20
2.2.2 Structural diffusion on a tree 21
2.2.3 Linear relationship between structural deviation and evolutionary time in global-σ model 24
2.2.4 Branch-specific structural drift rates 25

2.2.5 Non-evolutionary sources of structural variability 26
2.2.6 Uncorrelated structural perturbations (non-phylogenetic structural model) 27
2.2.7 Rotations and translations 27
2.2.8 Priors 28
2.2.8.1 Alignment and tree parameters 28
2.2.8.2 Substitution parameters and indel model parameters . . . 30
2.2.8.3 Priors for structural parameters 30
2.2.8.4 Shrinkage prior for branch-specific diffusivity 31
2.3 MCMC inference 32
2.3.1 Sampling rotations and translations 36
2.3.2 Monitoring convergence 36
2.4 Results and model comparison 37
2.4.1 Structural information improves alignments 39
2.4.2 Structure reduces topological uncertainty 39
2.4.3 Structural information reduces tree errors 45
2.4.4 Structure helps select between alternative topologies 45
2.4.5 Inclusion of structural inforamtion facilitates analysis of more challenging datasets 50
2.4.6 Phylogenetic structural model improves fit 53
2.4.7 Parameter inference 54
2.5 Heterogeneity in structural diffusivity 59
2.5.1 Heterogeneous structural evolution among the globins 60
2.5.2 Patterns of structural divergence 62
2.5.3 Structural determinants of evolutionary drift rates 66
2.5.4 Independence of drift rates and branch lengths 67
2.6 Discussion 68
2.6.1 Future work 70

3 Improved MCMC techniques for joint sampling of alignments and trees 73
3.1 New proposals for continuous parameters 73
3.1.1 Multiplicative proposals 74
3.1.2 Automatic tuning of proposal variances 75
3.2 Joint moves on indel parameters 75
3.2.1 Reparameterisation 76
3.2.2 Transformed priors 76

3.2.3 Illustration . . . 76
3.3 Interdependence of topology and alignment sampling . . . 78
3.3.1 Nearest-neighbour interchange moves may invalidate alignments . . . 78
3.4 Alignment sampling . . . 79
3.4.1 Three-way alignment sampling . . . 82
3.5 Improvements to topology sampling . . . 84
3.5.1 Original StatAlign topology+alignment move . . . 84
3.5.2 Simultaneous changes to topology and branch lengths . . . 85
3.5.3 LOCAL topology move . . . 86
3.5.4 Fixed-column topology moves . . . 86
3.5.5 Block imputation . . . 87
3.5.6 Persistent silent indels . . . 89
3.6 Model extension framework . . . 90
3.7 Combination moves . . . 92
3.7.1 Combination moves for model extensions . . . 92
3.8 Future improvements . . . 93

4 Representation of alignment uncertainty using directed acyclic graphs 94
4.1 Representing the distribution of sampled alignments . . . 95
4.1.1 Mapping columns to dynamic programming tables . . . 95
4.1.2 Intersections between alignments . . . 97
4.1.3 Equivalence classes of columns . . . 97
4.1.4 The alignment column graph . . . 100
4.2 Probability distributions on alignment DAGs . . . 102
4.2.1 Alignment probabilities in terms of pair marginals . . . 102
4.2.2 Motivations for using factored approximations . . . 102
4.2.3 Kullback-Liebler divergence . . . 103
4.2.4 Mean-field approximation . . . 104
4.2.5 Motivations for using the mean-field approximation . . . 105
4.2.6 Estimating marginal probabilities . . . 106
4.2.7 Reconstructing alignment probabilities from marginals . . . 107
4.2.8 Approximate summation over all alignments . . . 113
4.3 Summarising the alignment distribution . . . 117
4.3.1 Loss function formulation . . . 118
4.3.2 Loss functions corresponding to common accuracy measures . . . 120
4.3.3 Pairwise loss functions . . . 121

4.3.4 Modeller scores . . . 123
4.3.5 Efficient algorithms . . . 123
4.4 Efficient data structures . . . 128
4.5 Example application: summary alignments for simulated data and BAliBASE 129
4.5.1 Comparison to other methods . . . 130
4.5.2 Accuracy metrics . . . 131
4.5.3 Results: simulated data . . . 132
4.5.4 Results: BAliBASE . . . 137
4.6 Other applications of the alignment DAG . . . 139
4.6.1 Combining the output of other alignment programs . . . 139
4.6.2 Alignment DAGs as generators of alignment samples . . . 140

5 Downstream analysis in the presence of alignment uncertainty 142
5.1 Propagating alignment uncertainty into downstream inference . . . 143
5.1.1 Sequential approach . . . 143
5.1.2 DAG-based approach . . . 144
5.1.3 Approximate marginalisation over alignments . . . 144
5.2 Sequence annotation in the presence of alignment uncertainty . . . 145
5.2.1 Annotation of a single alignment with an HMM . . . 146
5.2.2 Decoding the HMM . . . 147
5.2.3 Annotating alignment DAGs . . . 149
5.2.4 Simultaneous decoding of alignment and HMM . . . 151
5.2.5 Parameterising the model . . . 151
5.2.6 Example: annotation of binding sites . . . 153
5.2.7 Conclusions and future work . . . 155
5.3 RNA secondary structure prediction in the presence of alignment uncertainty 157
5.3.1 Stochastic context-free grammars . . . 158
5.3.2 SCFGs for RNA secondary structure . . . 158
5.3.3 Emission probabilities . . . 159
5.3.4 Computing probabilities of structures under a SCFG . . . 160
5.3.5 Marginal probabilities . . . 162
5.3.6 Generalising the Inside and Outside algorithms to alignment DAGs 163
5.3.7 Computational complexity . . . 165
5.3.8 Minimum-risk decodings . . . 166
5.3.9 Evaluation data . . . 167
5.3.10 Results . . . 168

5.4 Conclusions 173

Bibliography **173**

List of Figures

2.1 One-dimensional example of the OU drift process, illustrating broadening of the conditional distributions as a function of evolutionary time 21
2.2 A one-dimensional OU branching process 23
2.3 Ten samples from the structural drift model on a tree 23
2.4 Posterior distributions for branch lengths are typically not sensitive to the choice of exponential prior 29
2.5 The posterior distribution for total tree length is only very weakly influenced by the choice of prior 29
2.6 Trace plot illustrating switching between components when using the hierarchical spike mixture prior 35
2.7 Alignment accuracy on simulated data 40
2.8 The two most frequently sampled tree topologies for the 5-globin data set . 41
2.9 Consensus trees for the 5-globin dataset 42
2.10 Consensus tree for the 5-globin dataset, derived using BAli-Phy with default settings 43
2.11 Consensus topology for the cysteine proteinases, under different model variants 44
2.12 Posterior distribution of topology errors relative to the true tree for simulated data 46
2.13 Consensus trees for globin datasets, using the sequence-only model 49
2.14 The structurally derived trees have very low uncertainty, and exhibit less sensitivity to the choice of dataset 51
2.15 Inclusion of structural information allows for analysis of larger datasets that exhibit high uncertainty when analysed under a sequence-only model . 52
2.16 Highest posterior density intervals for structural model parameters estimated on simulated data, on a 4-leaf tree 55
2.17 Highest posterior density intervals for structural model parameters estimated on simulated data, on an 8-leaf tree 56

2.18 Highest posterior density intervals for structural model parameters estimated on simulated data, on a 10-leaf tree 57
2.19 Consensus tree with branches scaled by local diffusivity parameters for the 12-globin dataset 61
2.20 Distributions for structural diffusivity parameters for leaf branches in the 12-globin dataset 62
2.21 The consensus tree for the cysteine proteinase dataset, with branches scaled according to mean branch length, and structural diffusivity 64
2.22 The consensus tree for the protein kinase set, with branches scaled according to mean branch length, and structural diffusivity 65
2.23 Illustration of charge differences between cysteine proteinase structures . . 68
2.24 Lack of correlation between consensus branch lengths and diffusivity parameters indicates separability of information sources 69
2.25 Average pairwise mean squared deviation versus predicted variability derived from crystallographic *B*-factors 71

3.1 Improved mixing for the λ and μ parameters of the TKF92 indel model after switching to the alternatively parameterised moves 77
3.2 Tree showing the relationships between the nodes involved in the nearest-neighbour interchange (NNI) move 79
3.3 Illustration of some possible ways in which columns can become invalidated after an NNI move 80
3.4 The multiple-HMM *hmm3* used to propose realignments pairs of sequences to an unknown parent sequence 84
3.5 Breakdown of the different contributions to the log Metropolis-Hastings ratios for a set of topology proposals, using the fixed-column NNI move (without block imputation) 88
3.6 Breakdown of the different contributions to the log Metropolis-Hastings ratios for a set of topology proposals, using the block-imputation version of the fixed-column NNI move 89
3.7 Creation of silent indels after a realignment move 89
3.8 StatAlign 3 running in GUI mode, with the structural model extension enabled 91

4.1 Correspondence between alignment columns and edges connecting cells in a dynamic programming matrix 96
4.2 Interchanges between alignments can result in a multiplication of the number of possible paths through the DAG 98

4.3 Crossovers between two alignments containing no interchange columns . . 98
4.4 Predecessor and successor functions, and equivalence classes of columns . . 99
4.5 Increased complexity of algorithms when a gap-insensitive coding scheme is used 101
4.6 Pair-HMM used to sample pairwise alignments between the two globin sequences 108
4.7 Pairwise alignments between two globins, sampled from the pair-HMM . . 109
4.8 Mean squared error in the approximation to the true posterior, as a function of the number of alignment samples, for a pairwise example 110
4.9 Dominance of mean-field estimates when the underlying HMM is neighbour-independent 110
4.10 Convergence of the mean-field and pair-marginal posterior estimates for a 2-sequence example 111
4.11 Convergence of the mean-field and pair-marginal posterior estimates for a 10-sequence example 112
4.12 The number of paths through the alignment column graph as a function of the number of alignments used to generate the graph 114
4.13 The proportion of the posterior mass contained in paths through the DAG . 116
4.14 The probability mass contained within the individual samples increases slowly 116
4.15 Comparison of minimum-risk paths under C- and C^+-based loss functions . 122
4.16 Example illustrating a case where the MergeAlign algorithm does not yield the global optimum 124
4.17 A collection of alignment samples visualised as a DAG, illustrating the workflow involved in the minimum-risk summary algorithm 125
4.18 The tree used to generate simulated data with DAWG 130
4.19 Accuracy of summary alignments for simulated data 133
4.20 Accuracy of summary alignments for simulated data, under the AMA measure 134
4.21 Accuracy as a function of the g parameter 136
4.22 Accuracy of summary alignments for BAliBASE alignments 138

5.1 A Hidden Markov Model on an alignment DAG 149
5.2 Minimum-risk annotation and alignment for a section of the Drosophila genome, projected onto a sequence of interest 155

5.3 The derivations for variables replaced according to each of the three types of productions in DENF 161
5.4 Illustration of the quantities computed by the outside algorithm 162
5.5 Illustration of the generalised inside algorithm on an example DAG 164
5.6 Prediction accuracy on test RNA datasets 169
5.7 Distribution of F-score ranks for DAGs of varying sizes 170
5.8 Runtime for RNA secondary structure prediction on DAGs of varying sizes 172

Chapter 1

Overview

1.1 Sequence alignment as a ubiquitous problem

Sequence alignment is one of the most intensely studied problems in bioinformatics, and is an important step in a wide range of different analyses, including identification of conserved motifs (Siepel *et al.*, 2005), analysis of molecular coevolution (Altschuh *et al.*, 1988; Hopf *et al.*, 2012; Knudsen and Hein, 1999), estimation of phylogenies (Höhl and Ragan, 2007), and homology-based protein structure prediction (Blundell *et al.*, 1987; Sali and Blundell, 1993).

Many of the most popular alignment methods seek to compute a single optimal alignment, using dynamic programming algorithms (Gotoh, 1982; Needleman and Wunsch, 1970) as well as a variety of heuristic procedures (Edgar, 2004; Feng and Doolittle, 1987; Kim *et al.*, 1994; Lupyan *et al.*, 2005; Löytynoja and Goldman, 2008; Notredame and Higgins, 1996). Similar approaches can be used to find maximum likelihood alignments under certain probabilistic models of insertion, deletion and substitution events (Bradley *et al.*, 2009; Hein *et al.*, 2000; Miklós *et al.*, 2004; Thorne *et al.*, 1991, 1992).

1.2 Effect of alignment on downstream inference

It has become increasingly clear in recent years that downstream analyses are often highly sensitive to the specific choice of alignment. There may be many plausible but suboptimal alignments within the vicinity of the optimum, containing additional—often complementary—information regarding the evolutionary relationships between the sequences (Godzik, 1996); selecting a single point estimate results in the loss of this additional information, and fails to account for the statistical uncertainty associated with different regions of the alignment (Lunter *et al.*, 2008).

A number of studies have highlighted the impact of the choice of alignment on subsequent phylogenetic inference (Dessimoz and Gil, 2010; Lake, 1991; Liu *et al.*, 2009, 2012; Morrison and Ellis, 1997; Ogden and Rosenberg, 2006; Simmons *et al.*, 2010; Wang *et al.*, 2011); in many cases different alignment methods, or different guide trees, can give rise to very different phylogenies (Blackburne and Whelan, 2013; Capella-Gutiérrez and Gabaldón, 2013; Dwivedi and Gadagkar, 2009; Lake, 1991; Thorne and Kishino, 1992; Wong *et al.*, 2008). Sensitivity to the alignment is also observed in the context of many other types of downstream analysis, including homology modelling of protein structures (Chivian and Baker, 2006; Schwarzenbacher *et al.*, 2004; Tramontano *et al.*, 2001), detection of correlated evolution (Dickson and Gloor, 2012; Dickson *et al.*, 2010), prediction of RNA secondary structure (Gardner *et al.*, 2005), and inference of positive selection (Blackburne and Whelan, 2013; Fletcher and Yang, 2010; Jordan and Goldman, 2012; Privman *et al.*, 2012).

1.3 Quantifying alignment uncertainty

A number of different approaches have been developed for quantifying the uncertainty associated with a multiple sequence alignment. Many of these methods focus on the notion of alignment *reliability*, i.e. the degree to which a particular alignment (or regions thereof)

can be trusted as a prediction of the homology between the sequences.

One set of approaches involves computing scores or summary statistics on a single alignment of interest, using these as a measure of reliability of the alignment. Some of these approaches equate reliability of a particular alignment column with a high score under the model used to generate the alignment (Capella-Gutiérrez *et al.*, 2009), the justification being that low-scoring columns are harder to distinguish from random noise, and so are more likely to contain erroneous homology statements; others generate the alignment using one scoring scheme, and measure its 'reasonableness' based upon another set of criteria (Ahola *et al.*, 2008; DeBlasio *et al.*, 2012), which may involve looking at the deviation of summary statistics from their expected background distribution under the null hypothesis of no homology (Dress *et al.*, 2008; Misof and Misof, 2009). One potential issue with some of these approaches is that they introduce a bias towards highly conserved regions, since they do not distinguish between evolutionary variability and statistical uncertainty, often using the term *alignment quality* as a synonym for reliability.

An alternative approach, first mentioned by Gatesy *et al.* (1993), involves generating a set of plausible alignments, and assessing the alignment uncertainty by measuring the similarity between the alignments in this set. This type of *consistency*- or *congruence*-based approach has a more natural statistical interpretation, but requires a method of generating alternative alignments, as well as a measure of alignment similarity or distance; the interpretation of the resulting measures of uncertainty may depend heavily on these two factors.

1.4 Generating sets of alignments

A variety of heuristic methods have been developed in order to generate sets of alignments for the purposes of measuring uncertainty. Perhaps the simplest of these is to align after reversing the residue order in one or more of the sequences (Landan and Graur, 2007),

although the efficacy of this technique is questionable (Hall, 2008; Wise, 2010), and it has limited theoretical basis. Another class of methods generates alternative alignments by perturbing parameters such as the guide tree (Penn *et al.*, 2010a,b), gap opening and extension penalties (Löytynoja and Milinkovitch, 2001; Wheeler, 1995), and substitution matrices (Collingridge and Kelly, 2012), and recomputing the optimal alignment with these alternative parameters. However, in all these cases the types of perturbations applied to the parameters will affect the resulting estimates of uncertainty in an unpredictable fashion (Misof and Misof, 2009).

Another approach is to look at a set of suboptimal alignments under a particular scoring scheme, given fixed parameters (Vingron, 1996; Waterman and Byers, 1985; Zuker, 1991), using these to search for regions of consistency (Landan and Graur, 2008; Mevissen and Vingron, 1996; Vingron and Argos, 1990). The variability among these suboptimal alignments can then be converted into a measure of statistical uncertainty. In the context of likelihood-based approaches, consistency between alignments may have an interpretation as a confidence interval around the maximum likelihood estimator; for score-based approaches, approximations to the distribution of scores can be used to convert to a probabilistic measure of uncertainty (Karlin and Altschul, 1993). However, it may be unclear how to perturb around the initial alignment in order to obtain a representative set of alternative alignments. Such approaches therefore employ approximate techniques such as pairwise resampling of alignments, and a user-specified threshold may be required to determine the extent of perturbation (Kim and Ma, 2011).

1.5 A Bayesian approach

Within a Bayesian framework, the collection of plausible alignments can be identified with the *posterior distribution* of the alignment given the sequences and other model parameters; this leads to a probabilistic interpretation of alignment uncertainty, whereby the fraction

of alignments containing a particular homology statement is a measure of the posterior probability of that homology statement.

For the pairwise case, alignments can often be sampled exactly from their posterior distribution under a particular evolutionary model using a dynamic programming approach (Durbin *et al.*, 1999; Webb *et al.*, 2002; Zhu *et al.*, 1998). However, for multiple sequences such approaches rapidly become computationally infeasible, and other types of procedures must be used. A popular option is to use Markov chain Monte Carlo (MCMC) in order to sample from the posterior distribution of alignments (Churchill, 1997; Green and Mardia, 2006; Green *et al.*, 2010a; Lunter *et al.*, 2005b; Metzler, 2003; Metzler *et al.*, 2001; Novák *et al.*, 2008; Redelings and Suchard, 2005a; Ruffieux and Green, 2009; Satija *et al.*, 2009; Suchard and Redelings, 2006). The main advantage of the MCMC approach is that it is guaranteed to sample alignments from the correct probability distribution, provided that the simulation is run for long enough to ensure convergence, although this may require significant amounts of runtime.

1.6 Filtering methods

A common approach to tackling the issue of alignment uncertainty has been to attempt to annotate particular regions of the alignment as unreliable using one of the aforementioned techniques, and to remove these regions before carrying out subsequent analysis. Filtering methods have in some cases been observed to yield improved inference for phylogenies (Castresana, 2000; Talavera and Castresana, 2007; Wu *et al.*, 2012) and positive selection (Jordan and Goldman, 2012; Privman *et al.*, 2012).

However, the specific choice of filtering method may have a strong influence on the results (Gatesy *et al.*, 1993), and uncertain regions of the alignment may also contain important information that is lost through the use of such methods. For example, tree accuracy is not related in a straightforward fashion to alignment uncertainty (Dessimoz and Gil, 2010),

and seemingly unreliable regions may be important for accurately resolving phylogenies (Ajawatanawong *et al.*, 2012; Lee, 2001). Regions of high alignment uncertainty can also correspond to sites with higher indel rates (Lunter, 2007; Lunter *et al.*, 2008), as well as regions of structural variability (Miklós *et al.*, 2008) or intrinsic disorder (Thompson *et al.*, 2011) in protein structures, and filtering these out may lead to unpredictable biases in subsequent analysis. In essence, filtering approaches attempt to reduce uncertainty by artificially censoring the data, which may lead to difficulties separating real signals from artifacts introduced by the filtering process.

1.7 Joint sampling approaches

In contrast, the Bayesian approach lends itself to the construction of a joint distribution for all the unknown parameters of interest, including trees, alignments, and various other model parameters. This allows for uncertainty in each of the parameters to be incorporated into the estimation of the others. The last decade has seen the development of several fully Bayesian approaches for performing joint inference on alignments along with other objects of interest, such as mutation rates (Metzler *et al.*, 2001), phylogenetic trees (Novák *et al.*, 2008; Redelings and Suchard, 2005a; Suchard and Redelings, 2006), information about the evolution of protein structure (Dryden *et al.*, 2007; Green *et al.*, 2010a; Ruffieux and Green, 2009), and the locations of putative regulatory elements (Satija *et al.*, 2008, 2009; Sinha and He, 2007); inference on these quantities after accounting for alignment uncertainty can then be obtained by averaging over alignments according to their posterior probability under the joint model.

However, for sequences that are highly divergent there may be a significant degree of uncertainty associated with the resulting alignments and trees, such that it may be difficult to interpret the output. One way of addressing this issue is to combine multiple different types of data into a joint, or mixed, evolutionary model (Ronquist and Huelsenbeck, 2003).

As well as offering a way of reducing uncertainty, this type of approach has the potential to lead to more robust and reliable results, since the resulting inference is based on multiple independent sources of information (cf. Kumar *et al.* (2012)).

For protein-coding genes, additional information regarding evolutionary relationships can be obtained from protein structures. Tertiary structure is typically much more highly conserved than sequence, even over large evolutionary distances (Illergård *et al.*, 2009; Panchenko *et al.*, 2005); structural similarity is therefore a more reliable way to infer homology in the so-called *twilight zone* of low sequence identity, leading to more accurate alignments (Eidhammer *et al.*, 2000; Hasegawa and Holm, 2009; Katoh and Standley, 2013), and potentially also phylogenies (Bujnicki, 2000; Johnson *et al.*, 1990; Lundin *et al.*, 2012).

In Chapter 2 we introduce a new framework for making use of structural information in joint estimation of phylogenies and alignments. To do so, we extend a recently developed stochastic model of pairwise structural evolution (Challis and Schmidler, 2012) to multiple structures on a tree, analytically integrating over ancestral structures to permit efficient likelihood computations under the resulting joint sequence-structure model.

We observe that the inclusion of structural information significantly reduces alignment and topology uncertainty, and reduces the number of topology and alignment errors in cases where the true trees and alignments are known. In some cases the inclusion of structure results in changes to the consensus topology, indicating that structure may contain additional information beyond that which can be obtained from sequences.

We use the model to investigate the order of divergence of cytoglobins, myoglobins, and haemoglobins, and observe a stabilisation of phylogenetic inference: while sequence-based inference assigns significant posterior probability to several different topologies, the structural model strongly favours one of these over the others, and is more robust to the choice of dataset.

1.8 Software engineering for inference with new joint sampling approaches

In order to facilitate the implementation of new coestimation approaches such as the above, it is highly desirable to have a framework in which additional layers can easily be added on to the basic evolutionary model. In addition, it is also of great interest to explore ways in which other types of data besides sequences can be used to assist with inference of phylogenies and alignments, since this may increase the robustness of the conclusions to model assumptions (Kumar *et al.*, 2012).

The previous version of StatAlign (Novák *et al.*, 2008) was essentially hard-coded to use a single core model, with modifications only possible to the substitution models. As such, we developed a generic model extension framework for StatAlign, whereby plugins can provide an additional contribution to the joint likelihood, based upon specified distributions for parameters of interest. Model extension plugins that are able to compute a column-wise contribution to the likelihood are also used to improve the alignment proposals under the joint model.

This new software has been released as StatAlign 3 (Herman *et al.*, 2014e), and was used to implement the joint sequence-structure model described in Chapter 2.

As well as the model extension framework, StatAlign 3 also includes a number of improvements to the MCMC sampling functionality. The source code has been significantly restructured, allowing for easy addition of new MCMC moves, which can also be combined into joint moves to allow for more efficient mixing, for example combining topology changes and branch length proposals. As described in Chapter 3, we have introduced new types of moves for jointly resampling alignments and topologies, allowing for much more efficient mixing over the space of trees. Other changes include the ability to generate pairwise realignment proposals from the full posterior, and joint multiplicative moves on all branch lengths, as well as moves on the indel model parameters under different reparame-

terisations, to deal with high correlation between insertion and deletion rates.

As part of the improved MCMC framework, different prior and proposal distributions can be specified for each move, and we have implemented a selection of standard priors and proposal distributions for continuous parameters (Gaussian, Gamma etc.). In addition, there is now the option to keep the tree fixed throughout the analysis, to increase efficiency in cases where the phylogeny is already known.

With an increased number of different move types, it becomes essential to automate the adjustment of tuning parameters in order to ensure good exploration of the space. To this end, we have added mechanisms for automatically modifying proposal distributions during the burn-in period in order to ensure acceptance rates fall within the desired range.

1.9 Graph-based approaches for analysing large sets of alignments

Sampling-based approaches such as StatAlign generate as output a collection of alignments and trees, along with model parameters, sampled from the joint posterior distribution. Although a large number of techniques exist for analysing MCMC output for continuous parameters, and there is an increasingly large literature surrounding the analysis of sets of phylogenetic trees, there has been very little work looking at how to represent a posterior distribution over alignments. Since the space of alignments is very high-dimensional, a collection of samples from the posterior typically contains no duplicates, such that empirical frequencies cannot be used to estimate posterior probabilities.

In Chapter 4 we present a directed acyclic graph (DAG)-based approach for representing a collection of sampled alignments. As well as encoding the set of sampled alignments, this DAG representation allows an initial set of samples to be used to generate a larger set of alignments under the same approximate distribution. This is possible due to the fact that the number of possible paths through the DAG is typically much larger than the number

of samples used to generate the DAG. For a large class of models this formulation allows for posterior alignment probabilities to be estimated reliably from a reasonable number of samples.

The very large set of alignments contained within the DAG can then be used for approximate marginalisation over alignments, and summary alignments can be generated by selecting paths that minimise the expectation of various types of loss functions. More generally, the graph structure presents a probabilistically-weighted search space within which various types of analysis can be conducted.

1.10 Alternatives to joint sampling

The simple and computationally efficient nature of the representation described in Chapter 4 makes it practical to adopt a more principled, probabilistic approach to quantifying and making use of alignment uncertainty, even when joint sampling approaches are not feasible. Rather than analysing each sampled alignment individually, downstream analysis can be carried out on the entire set of alignments contained within the DAG, with each alignment weighted according to its empirical probability.

Two examples of algorithms that can easily be adapted to operate on the alignment DAG are phylogenetic inference, and sequence annotation using hidden Markov models (HMMs). A slightly more involved example is that of RNA secondary structure prediction using stochastic context-free grammars (SCFGs). The common feature to these algorithms is that they involve dynamic programming recursions, allowing the algorithms to be adapted to the DAG structure while maintaining efficiency.

In Chapter 5, we examine these approaches, and illustrate that the additional information contained within the DAG allows for improved predictions, resulting in substantial gains over simply analysing a set of alignments one by one.

1.11 Statement of collaboration

Bioinformatics is by its nature a collaborative field, and much of the work detailed in this thesis is the product of collaborations with various colleagues. In this thesis I have focused on including those parts of the work that correspond to my personal contributions; below I present specific details of areas which have benefitted from the input of collaborators.

Chapter 2 is the product of a close collaboration with Chris Challis, at Duke University, and much of the material in this Chapter was included in a paper that he and I submitted as joint first authors (Herman *et al.*, 2014d). Unless specifically stated, all the figures and data analysis in Chapter 2 represent my work; the majority of the text is also my work, although a number of the sections of this Chapter 2 are taken from the submitted manuscript, hence include contributions from the other co-authors. The main contributions of Chris Challis to the work presented in this Chapter include the initial extension of the pairwise structural model to a full tree, and the formulation of a branch-specific rates model, which we then developed together. After I proposed the inclusion of a parameter to account for non-evolutionary sources of variability, Chris formulated the variant of the model with a global baseline variance parameter; I then developed the extension to allow for heteroskedasticity by making use of *B*-factor information. Chris also worked extensively on the inclusion of rotations and translations in the model, and adapted his existing algorithms for sampling these parameters by MCMC, as well as working on the initial development of the code computing the structural contribution to the likelihood; Chris also simulated data under the model, as discussed in Chapter 2.

The software development detailed in Chapter 3 was carried out in conjunction with Ádám Novák, who also proposed the introduction of the model extension framework, and the fixed-column imputation algorithms were developed after discussions with Chris Challis. However, all the algorithms and analyses detailed in Chapter 3 represent my novel contribution. Much of this Chapter has been submitted as part of Herman *et al.* (2014e).

In Chapter 4, the simulated data, scripts for computing alignment accuracy, and soft-

ware for computing the minimum risk alignment were developed by Ádám Novák; the mapping of alignment columns to edges in the dynamic programming matrix was developed based on earlier work by István Miklós (Satija *et al.*, 2009). The rest of the material in the Chapter, including statistical methodology, algorithm development and data analysis, represents my own work. The work in this Chapter forms the basis of Herman *et al.* (2014a).

The software for inference with HMMs as described in Chapter 5 was developed by Ádám Novák, but the algorithm development and data analysis represent my own work. The material in this section forms the basis for a manuscript that is currently in preparation (Herman *et al.*, 2014c). The algorithms for RNA structure prediction on alignment DAGs were developed in collaboration with Rune Lyngsø with assistance from James Anderson; the simulated data were generated by James Anderson, and the Java implementation was developed with extensive assistance from Cyrus Rich, Diana-Elena Gratie and Maria Sinziana Aştefănoaei. The initial idea was developed as a result of conversations I had with Rune Lyngsø and Ádám Novák, following our successful implementation of the HMM version of the algorithm. We were encouraged to pursue the project after discovering some parallels with earlier work comparing an HMM with an SCFG (Jagota *et al.*, 2001). The RNA-related material in Chapter 5 is largely taken from a manuscript we are preparing for publication (Herman *et al.*, 2014b), hence some of the text represents the contributions of collaborators. Specifically, the background on SCFGs is largely due to James Anderson, and the description of the dynamic programming algorithms benefitted substantially from the input of Rune Lyngsø; several of the figures in this section were also generated by Rune Lyngsø, and are indicated as such in the relevant captions.

Chapter 2

Estimation of alignment and phylogeny with a joint sequence-structure model

Recently, Challis and Schmidler (2012) introduced a probabilistic evolutionary model of the joint evolution of protein sequence and structure. In contrast to structurally-constrained sequence models that modulate substitution rates based on a fixed structure (Choi *et al.*, 2007; Kleinman *et al.*, 2010; Robinson *et al.*, 2003; Rodrigue *et al.*, 2005), this approach includes an explicit model for the evolution of structure, allowing for structural information to be used to help infer evolutionary distances. Structural evolution is modelled according to a diffusion process with drift, which allows for tractable computation of the likelihood in the resulting joint sequence-structure model. Significant improvements were observed in the accuracy of inferred pairwise divergence times, especially for highly divergent sequences.

In this Chapter, we extend the model of Challis and Schmidler (2012) to a tree, and explore the utility of incorporating structural information into joint estimation of multiple alignments and phylogenies. Since relatively little is known about structural evolutionary

processes, we also introduce a model for heterogeneity in rates of structural evolution, which reduces the potential for conflict between structure- and sequence-based trees (Garau *et al.*, 2005).

We also add a model of background (non-evolutionary) variability in structures, making use of prior information obtained from the x-ray crystallography experimental data, and drawing on aspects of other earlier probabilistic models of protein structure (Green and Mardia, 2006; Green *et al.*, 2010a; Rodriguez and Schmidler, 2014; Schmidler, 2006; Wang and Schmidler, 2014).

2.1 Probabilistic evolutionary models

In what follows, we deal with classes of probabilistic models on binary trees. Biologically these trees define phylogenetic relationships between a set of organisms; probabilistically, given the sequence at a particular parent vertex, evolution along each of its child branches is assumed to proceed independently.

2.1.1 Sequence and structure data

We consider a sequence evolving on a tree, Υ, with vertices $\mathcal{V}_\Upsilon$ and edges $\mathcal{E}_\Upsilon$, according to an evolutionary model with parameters (Φ, Λ, Θ), which describe rates of substitution, insertion and deletion (*indel*) events, and structural evolution processes, respectively. Associated with the K tips of the tree is a set of K homologous sequences $\mathcal{S} = \{S^{(1)}, \ldots, S^{(K)}\}$, with $S^{(k)}$ of length $L^{(k)}$, and corresponding three-dimensional structures, $\mathcal{C} = \{C^{(1)}, \ldots, C^{(K)}\}$, where $C^{(k)}$ is an $L^{(k)} \times 3$ matrix containing the Euclidean coordinates of the C_α atoms of structure k. In order to make use of the tree structure to permit tractable inference, each of the internal nodes of the tree is augmented with an associated sequence and structure, the corresponding sets denoted by $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{C}}$ respectively. The structural coordinates and characters associated with these internal sequences will eventually be marginalised out an-

alytically.

2.1.2 Representation of a multiple alignment

A multiple alignment can be represented as a set of pairwise alignments along the branches of a tree, $\tilde{\mathcal{M}} = \{M^{(k,l)}\}$, with $(k,l) \in \mathcal{E}_\Upsilon$. Each pairwise alignment, of length $L^{(k,l)} \leq L^{(k)} + L^{(l)}$, can be thought of as a series of columns in a $2 \times L^{(k,l)}$ matrix, indicating homology between characters in $S^{(k)}$ and $S^{(l)}$, i.e. the parent and child sequences along the branch. Each such column can take one of three possible states:

$$M_i^{(k,l)} \in \left\{ \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x \\ - \end{pmatrix}, \begin{pmatrix} - \\ y \end{pmatrix} \right\} \tag{2.1}$$

where $x \in \{1, \ldots, L^{(k)}\}$ and $y \in \{1, \ldots, L^{(l)}\}$ indicate the index of the characters aligned in the column, and – indicates an insertion or deletion. We will also denote by $M^{(k)}$ the row corresponding to sequence k in $M^{(k,l)}$, with the zero elements removed, equal to the vector $(1, \ldots, L^{(k)})$; one of the requirements for a valid set of alignments, $\tilde{\mathcal{M}}$, is that all the pairwise alignments should be consistent in the sense that the mapping $M^{(k,l)} \mapsto M^{(k)}$ is the same for all l. Another requirement is that $L^{(k)}$ be equal to the length of $S^{(k)}$ when k is a leaf node. The full alignment, $\tilde{\mathcal{M}}$, can be projected down to a *leaf alignment* between the sequences at the leaves of the tree, $\mathcal{M}$, expressed in the familiar tabular format. We omit further notational details here for brevity.

2.1.3 Joint model for sequence and structure

The first phylogenetic evolutionary models to be developed allowed only for substitution events, assuming the alignment of the sequences to be known and fixed (Felsenstein, 1981; Kimura, 1980). However, work over the last two decades has shown that probabilistic modelling of insertion and deletion (indel) events can yield valuable additional information regarding evolutionary processes (Dessimoz and Gil, 2010; Löytynoja and Goldman,

2005), partly due to the rarity of such events (Lunter *et al.*, 2003a; Westesson *et al.*, 2012). In this work we build on these existing approaches, adding a probabilistic model of protein structure to yield a joint Bayesian model for substitutions, indels, and structural evolution on a tree.

For reasons of tractability, we focus attention on models where the joint posterior of the unknown parameters of interest, given the observed (leaf) and augmented (internal node) data, can be factored as the product of substitution and structural contributions, and a stochastic indel process:

$$p(\tilde{\mathcal{M}}, \Upsilon, \Phi, \Theta, \Lambda \mid \mathcal{S}, \tilde{\mathcal{S}}, C, \tilde{C}) \propto p(\Upsilon) \underbrace{p(\tilde{\mathcal{M}}, \Lambda \mid \Upsilon)}_{\text{indel}} \underbrace{p(\Phi, \Theta \mid \mathcal{S}, \tilde{\mathcal{S}}, C, \tilde{C}, \tilde{\mathcal{M}}, \Upsilon)}_{\text{substitution/structure}} \tag{2.2}$$

The above factorisation will generally only be possible for independent-site models of substitution and structural evolution; insertions and deletions can change neighbourhood relationships, such that substitution, structure and indel processes are in general not separable in neighbour-dependent models.

In this work we also make the further assumption of separability between the substitution and structural evolutionary processes, such that

$$p(\Phi, \Theta \mid \mathcal{S}, \tilde{\mathcal{S}}, C, \tilde{C}, \tilde{\mathcal{M}}, \Upsilon) = \underbrace{p(\Phi \mid \mathcal{S}, \tilde{\mathcal{S}}, \tilde{\mathcal{M}}, \Upsilon)}_{\text{substitution}} \underbrace{p(\Theta \mid C, \tilde{C}, \tilde{\mathcal{M}}, \Upsilon)}_{\text{structure}} \tag{2.3}$$

It should be noted that the branch lengths in the tree Υ are common to the substitution and structure components, such that the above separation still permits structural evolution to be expressed as a function of substitutions per site along each branch. Although it is also possible to formulate independent-sites models with a more explicit dependence between sequence and structure (for example by allowing for Θ to be a function of the amino acid content for a particular site), we leave such developments for future work.

2.1.4 Marginal posterior

Ultimately we are interested in the marginal posterior distribution over alignments, trees and model parameters obtained by integrating over the unobserved internal node data

$$p(\tilde{\mathcal{M}}, \Upsilon, \Phi, \Theta, \Lambda \mid \mathcal{S}, C) \propto p(\Upsilon)p(\tilde{\mathcal{M}}, \Lambda \mid \Upsilon) \times p(\Phi)p(\mathcal{S} \mid \Phi, \tilde{\mathcal{M}}, \Upsilon) \times p(\Theta)p(C \mid \Theta, \tilde{\mathcal{M}}, \Upsilon)$$

We focus on cases where the observed data likelihoods $p(\mathcal{S} \mid \Phi, \tilde{\mathcal{M}}, \Upsilon)$ and $p(C \mid \Theta, \tilde{\mathcal{M}}, \Upsilon)$ can be computed exactly by analytical summation and integration over ancestral characters and coordinates. Although, with some simplifying assumptions, certain indel models also allow for analytical summation over internal node alignments (Bouchard-Côté and Jordan, 2013; Lunter *et al.*, 2005b; Thorne *et al.*, 1991), for many models of interest this is not possible, yielding a problem of exponential complexity (Lunter *et al.*, 2005a), hence we focus on the general case of inference for the full alignment $\tilde{\mathcal{M}}$ rather than directly targeting the marginal posterior for the leaf alignment $\mathcal{M}$.

Beyond the factorisability in equation (2.2), the statistical alignment framework we present here is not dependent on particular model choices for substitution and indel processes, but we will briefly describe the specific choices used in this work for the purposes of illustrating how they combine with the structural model. We introduce the structural model in more detail in the subsequent section, but note here that one of the key features of the approach we will present is that it allows the integration over unknown ancestral structures to be carried out analytically, greatly increasing the tractability of the resulting model.

2.1.5 Indel model

In this work we focus on the TKF92 model (Thorne *et al.*, 1992) to generate the probability $p(\tilde{\mathcal{M}} \mid \Lambda, \Upsilon)$. This model is a birth/death process on fragments, each of which contains a contiguous run of characters (in our case amino acids). Fragments are inserted at rate λ and are deleted with rate μ; the length of each fragment is geometrically distributed according to

a probability r. We adopt the scheme discussed by Miklós *et al.* (2008), whereby fragments are not inherited from parent to child branches; the contribution to the posterior for $\tilde{\mathcal{M}}$ from the indel model can then be factored over the branches of the tree

$$p(\tilde{\mathcal{M}}, \Lambda \mid \Upsilon) = \prod_{j \in \mathcal{V}_\Upsilon} p(M^{(j)}, \Lambda) \prod_{(k,l) \in \mathcal{E}_\Upsilon} \frac{p(M^{(k,l)}, \Lambda \mid \Upsilon)}{p(M^{(k)}, \Lambda) p(M^{(l)}, \Lambda)}$$

$$= p(M^{(\text{root})}, \Lambda) \times \frac{\prod_{(k,l) \in \mathcal{E}_\Upsilon} p(M^{(k,l)}, \Lambda \mid \Upsilon)}{\prod_{j \in an(\Upsilon)} p(M^{(j)}, \Lambda)^2} \tag{2.4}$$

where $\mathcal{V}_\Upsilon$ and $\mathcal{E}_\Upsilon$ are the sets of vertices and, respectively, edges in the tree Υ, and $an(\Upsilon)$ is the set of ancestral (non-leaf) nodes of the tree. The vector $M^{(j)}$ is equal to one of the rows in the pairwise alignment $M^{(k,l)}$. The second line assumes that the tree is binary, which will be the case in all the examples we consider.

Each pair term in the numerator of equation (2.4) can be computed via dynamic programming using the pair-HMM representation of the indel model (Miklós *et al.*, 2008), allowing the augmented likelihood to be computed in time linearly proportional to the number of branches in the tree, and the square of the average sequence length. The stationary probabilities for individual nodes are derived by Thorne *et al.* (1992), and take the form

$$p(M^{(k)} \mid \Lambda) \equiv p(L^{(k)} \mid \lambda, \mu, r) \tag{2.5}$$

$$= (1 - m)\, m(1 - r)\, [m(1 - r) + r]^{L^{(k)} - 1} \tag{2.6}$$

where $L^{(k)}$ represents the length of the kth sequence, equivalent to the length of $M^{(k)}$, and $m = \lambda/\mu$.

2.1.6 Substitution model

Under the independent-sites assumption, the substitution process is modelled as a collection of independent processes on individual amino acids. This allows the marginal likelihood of the leaf sequences, given a particular alignment $\tilde{\mathcal{M}}$, to be calculated using the familiar sum-product algorithm of Felsenstein (1981), yielding the quantity

$$p(\mathcal{S} \mid \Phi, \tilde{\mathcal{M}}, \Upsilon) = \sum_{\tilde{\mathcal{S}}} (\mathcal{S}, \tilde{\mathcal{S}} \mid \Phi, \tilde{\mathcal{M}}, \Upsilon) \tag{2.7}$$

The analyses conducted here employ the Dayhoff *et al.* (1978) matrix of amino acid substitution to parameterise Φ, although other choices are possible. Algorithms used to evaluate likelihoods such as the above are discussed in more detail in Chapter 3.

2.2 Structural drift model

There is empirical evidence of correlation between evolutionary time and structural divergence, although the exact nature of this relationship has remained the source of much speculation (Illergård *et al.*, 2009). Chothia and Lesk (1986) famously observed an exponential relationship between structural divergence of core homologous residues as measured by RMSD and sequence divergence as measured by sequence identity. This original relationship was proposed based on a small dataset that was available at the time: 32 pairs of homologous proteins, as well as 5 instances of the same protein crystallised under different conditions. More recently, several authors have observed a linear relationship when sequence identity is converted to a measure of substitutions per site (Illergård *et al.*, 2009), or if sequence identity and RMSD are replaced by approximate measures of significance (Wood and Pearson, 1999), although in some families a non-linear relationship may still be observed (Panchenko *et al.*, 2005). In all cases structural divergence is observed to increase as sequence similarity decreases.

2.2.1 Model specification

In order to construct a model that allows for structural divergence to be a function of evolutionary distance, Challis and Schmidler (2012) introduced a diffusion-based model of structural drift. Whereas a probabilistic substitution model employs a continuous-time, finite-state Markov process, this structural model utilises a reversible diffusion process in 3D space, modelling fluctuations in the amino acid positions (represented by their C_α coordinates). As discussed earlier, independence between atoms is assumed to retain tractability.

Under this model, structural evolution is modelled using an Ornstein-Uhlenbeck (OU) process on each C_α atom. Unlike Brownian motion, the OU process has a well-defined stationary distribution and so is reversible, allowing the combined structural, indel, and substitution processes to form a reversible joint model.

With $C_{ij}(t)$ representing the jth coordinate of the ith C_α at time t, the structural drift model describes the change in coordinates over time according to the following stochastic differential equation

$$dC_{ij}(t) = -\theta\, C_{ij}(t)\, dt + \sigma dB \tag{2.8}$$

where dB is standard Brownian motion, and θ is the rate at which a structure loses memory of its previous configuration, which we term the *structural drift rate*. The equilibrium distribution and conditional distributions of this process are Gaussians

$$C_{ij}(\infty) \sim \mathsf{N}(0, \tau) \tag{2.9}$$

$$C_{ij}(t) \mid C_{ij}(s) \sim \mathsf{N}\left(C_{ij}(s)e^{-\theta(t-s)}, \tau(1 - e^{-2\theta(t-s)})\right) \tag{2.10}$$

with the marginal variance $\tau = \sigma^2/(2\theta)$ proportional to the expected radius of gyration multiplied by the length of the structure. The quantity $\sigma^2/2$ can be thought of as a diffusion

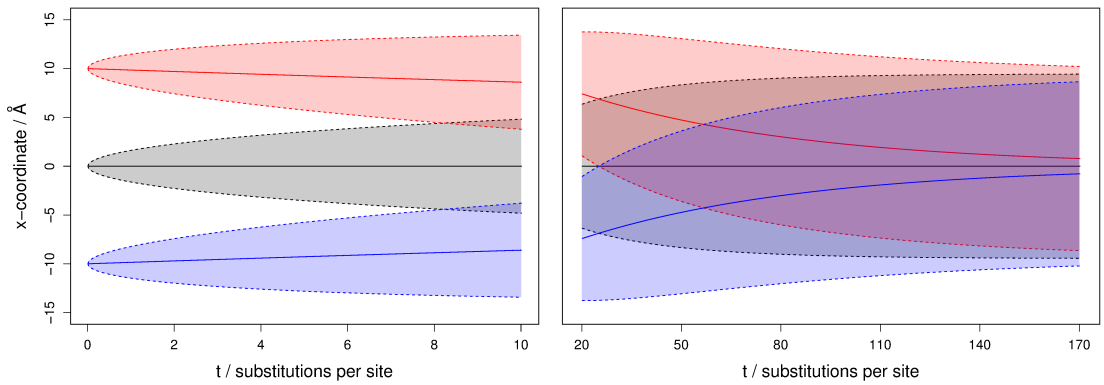


Figure 2.1: One-dimensional example of the OU drift process, illustrating broadening of the conditional distributions as a function of evolutionary time.

coefficient, with the expected mean square deviation after a time t approximately equal to $\sigma^2 t$ (*see Section 2.2.3*). As such, we will refer to σ^2 as the *structural diffusivity*.

Figure 2.1 shows the 95% confidence intervals for the conditional distribution in a one-dimensional version of this process, for three different starting coordinates, with $\theta = 0.015$ and $\sigma^2 = 0.7$. Within shorter timescales, the mean-reversion does not have a strong effect, and the key feature of the process is that the variance increases as a function of time. For longer times, the conditional distributions eventually lose memory of the starting conditions, converging to the same equilibrium distribution. However, parameter values typically inferred on real data place the model in the non-equilibrium regime.

2.2.2 Structural diffusion on a tree

When extending this process to a set of structures related by a phylogeny, we must contend with an unknown ancestral structure at each internal node. Fortunately, the OU process allows for analytical integration over the unknown ancestral structure coordinates, such that the joint likelihood of the observed structures at the tips of the tree, $p(C \mid \tilde{\mathcal{M}}, \Theta, \Upsilon)$, can be computed very efficiently. As discussed by Hansen and Martins (1996), for an OU process on a tree (cf. Figure 2.2), the joint distribution for the data at the leaves is a multivariate Gaussian, in our case with a zero mean. The Markovian nature of the OU

	Symbol	Domain	Meaning
Data structures	Υ	Binary tree	Phylogenetic tree
	$\mathcal{M}, \tilde{\mathcal{M}}$	*eqn (1)*	Observed (+ ancestral) alignments
	$\mathcal{S}, \tilde{\mathcal{S}}$	$S^{(k)} \in \chi^{L_k}$	Observed (+ ancestral) sequences
	$C, \tilde{C}$	$C^{(k)} \in \mathbb{R}^{3\times L_k}$	Observed (+ ancestral) coordinates
TKF92 indel model	$\Lambda\ \lambda$	$(0, \mu)$	Insertion rate
	$\Lambda\ \mu$	(λ, ∞)	Deletion rate
	$\Lambda\ r$	$(0, 1)$	Geometric rate for indel length
Structural model	$\Theta\ \tau$	$(0, \infty)$	Average structural radius of gyration
	$\Theta\ \epsilon_i$	$(0, \infty)$	Baseline variance for the ith column
	$\Theta\ \theta_k$	$(0, \infty)$	Rate of memory loss along the kth branch
	$\Theta\ \sigma_k^2$	$(0, \infty)$	Structural diffusivity of the kth branch
	$\Theta\ \sigma_g^2$	$(0, \infty)$	Mean structural diffusivity
	$\Theta\ \nu$	$(0, \infty)$	Variance of σ_k^2 parameters (on log scale)
	$\Theta\ \gamma$	$[0, 1]$	Proportion of σ_k^2 parameters fixed at σ_g^2
Substitution model	Φ	Unspecified	Substitution model parameters

Table 2.1: Mathematical notation used in defining the structural model.

process means that the elements of the covariance matrix can be computed analytically, with $\Sigma_{kl}[\tau, \theta, \Upsilon] = \tau e^{-\theta\, d_{kl}(\Upsilon)}$, where $d_{kl}(\Upsilon)$ is the distance between leaves k and l along branches of Υ.

Denoting by $C_j^{(\mathcal{M}_i)}$ the length-$|\mathcal{M}_i|$ vector obtained by taking the jth coordinate of each observed (leaf) structure containing a character at the ith column, the marginal likelihood of the observed structures is then given by a product over the L columns of the alignment and the three spatial dimensions:

$$p(C \mid \tilde{\mathcal{M}}, \Theta, \Upsilon) = \prod_{i=1}^{L} \prod_{j=1}^{3} \mathsf{N}_{|\mathcal{M}_i|}\left(C_j^{(\mathcal{M}_i)} \mid \mathbf{0}, \Sigma_{\mathcal{M}_i}[\tau, \theta, \Upsilon]\right) \tag{2.11}$$

where $\Sigma_{\mathcal{M}_i}$ is a submatrix of Σ of dimension $|\mathcal{M}_i|$ formed by selecting the columns and rows corresponding to ungapped positions in the alignment column $\mathcal{M}_i$.

Figure 2.3 illustrates a set of samples on a tree drawn from the structural drift model with $\sigma^2 = 0.7\text{Å}^2$/substitution per site, and $\tau = 70\text{Å}^2$, evolving from structure 2DN2 (human haemoglobin) at the root.

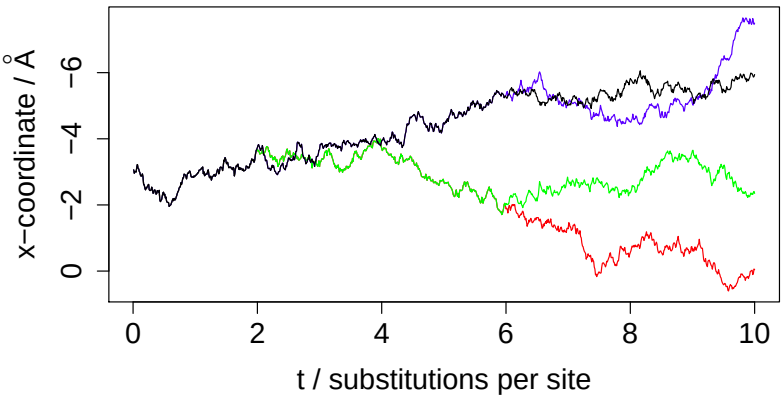


Figure 2.2: A one-dimensional OU process with branch points corresponding to three evolutionary divergence events. Here the parameters used were $\theta = 0.015$ and $\sigma^2 = 0.5$.

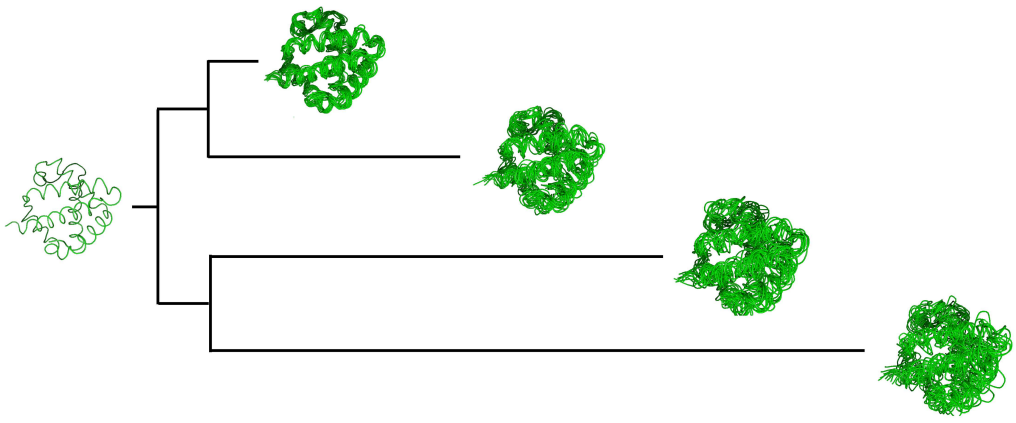


Figure 2.3: Ten samples from the structural drift model on a tree. The parameters used here were $\sigma^2 = 0.7\text{Å}^2$/substitution per site, and $\tau = 70\text{Å}^2$. With σ^2 set to zero we would see equal variability at each leaf, whereas the structural drift model proposes that structural divergence will be larger over greater evolutionary distances, in accordance with empirical observations. (*Figure generated by CJ Challis.*)

2.2.3 Linear relationship between structural deviation and evolutionary time in global-σ model

The model thus far assumes a constant structural diffusion coefficient, σ^2, throughout the phylogenetic tree. This assumes that structures respond to sequence mutations in a homogeneous fashion, leading to an approximately linear relationship between evolutionary time and mean-square-deviation. For a single σ^2 parameter over the whole tree, the expected mean-square-deviation (MSD) is

$$\begin{aligned}
\frac{1}{n}\sum_{ij}\mathbb{E}[(C_{ij}^{(t)} - C_{ij}^{(0)})^2 \mid C_{ij}^{(0)}] &= \frac{1}{n}\sum_{ij}\left((1-e^{-\theta t})C_{ij}^{(0)}\right)^2 + \frac{\sigma^2}{2\theta}(1-e^{-2\theta t}) \\
&\approx \frac{1}{n}\sum_{ij}(\theta t\, C_{ij}^{(0)})^2 + \sigma^2\left(t - \frac{\theta t^2}{2}\right) \\
&\approx \theta t^2\sigma^2 + \sigma^2 t \\
&\approx \sigma^2 t
\end{aligned}$$

where the first approximation results from $1-e^{-\theta t} \approx \theta t$, the second follows the relationships $\frac{1}{3n}\sum(C_{ij}^{(0)})^2 \approx \tau^2$ and $\tau^2 = \sigma^2/2\theta$, and the third from $\theta \ll \sigma^2$.

It should be noted that this expected linear relationship between MSD and branch length holds in a structure-only model; when combined with the sequence model, different relationships may be observed, since sequence information will also affect the estimation of the branch lengths.

With branch-specific diffusivity coefficients, it is possible to derive a similar relationship:

$$\frac{1}{n}\sum_{ij}\mathbb{E}[(C_{ij}^{(t)} - C_{ij}^{(0)})^2 \mid C_{ij}^{(0)}] \approx \sum_{k}\sigma_k^2 t_k \tag{2.12}$$

where k runs across the set of changepoints between time 0 and time t. Since each σ_k^2 is allowed to vary independently, this allows for a wide range of possible relationships between evolutionary time and structural divergence, reducing to a linear relationship when

$\sigma_k^2 = \sigma_g^2$ for all k.

2.2.4 Branch-specific structural drift rates

In order to allow for more general relationships between structural and sequence deviation, as well as reducing potential conflict between sequence- and structure-based trees, we relax this assumption and allow the structural diffusivity to vary over the tree. Following the approach of Thorne *et al.* (1998) and Aris-Brosou and Yang (2002) with regards to variable rates of sequence evolution, we allow σ^2 to vary by branch, which provides additional flexibility while allowing important properties such as infinite divisibility and reversibility to be maintained across the tree.

There are many ways in which this can be done; here we consider a model formulation that limits the number of additional parameters required. Let $\mathcal{E}_\Upsilon$ be the set of branches of tree Υ, with $\{\sigma_k^2, \theta_k \mid k \in \mathcal{E}_\Upsilon\}$ the associated set of structural parameters. Allowing both σ_k^2 and θ_k to vary by branch does not preserve a common stationary distribution at each node of the tree, making the joint distribution difficult to specify. To solve this issue, we instead consider the alternative parameterisation $\tau_k = \sigma_k^2/(2\theta_k)$ with $\tau_k = \tau$ for all k, such that τ represents the equilibrium variance common to all nodes of the tree, while σ_k^2 is the local structural diffusivity, which is allowed to vary by branch. Since $\sigma_k^2 = 2\tau\theta_k$, the diffusivity of a branch is proportional to its structural drift rate, hence when describing heterogeneity across the tree, we will refer to these quantities interchangeably. The joint distribution of leaf nodes under this model remains simple and easy to obtain. The marginal distribution for each coordinate is then $\mathsf{N}(0, \tau)$ as before, while the covariance between coordinates of leaves k and l becomes

$$\Sigma_{kl}[\tau, \theta, \Upsilon] = \tau \exp\left\{ \sum_{m \in \pi(k,l|\Upsilon)} t_m(\Upsilon)\, \frac{\sigma_m^2}{2\tau} \right\} \tag{2.13}$$

where $\pi(k, l \mid \Upsilon)$ represents the set of branches lying on the unique shortest path from leaf

k to leaf l, and $t_m(\Upsilon)$ is the length of branch m in tree Υ.

2.2.5 Non-evolutionary sources of structural variability

With sequence data, sequencing errors are relatively rare, such that any differences between sequences can generally be attributed to mutation events. However, for structural data, other sources of variability in the coordinates arise from factors such as flexibility of polypeptide chains, variable conformations, and measurement error (Grishin, 1997; Gutin and Badretdinov, 1994; Illergård *et al.*, 2009). Moreover, this uncertainty may vary across the protein, with surface residues and loops exhibiting increased flexibility over buried core positions.

Information about this uncertainty for high-resolution structures solved by x-ray diffraction is contained in crystallographic B-factors for each atomic coordinate. These values, reported by the crystallographer, are intended to summarise a combination of experimental uncertainty and thermal fluctuations, and are often strongly correlated with intrinsic structural flexibility measured by nuclear magnetic resonance and molecular dynamics simulations (Rueda *et al.*, 2007). B-factors can be converted to units of coordinate uncertainty using approximate formulae such as the *diffraction-component precision index* (Cruickshank, 1960, 1999). This can be combined with additional assumptions (Schneider, 2000) to obtain a linear relationship between the B-factor and the standard deviation of the coordinates for each atom. We therefore model the variance for the ith atom of structure k (with B-factor B_{ki}) as

$$\epsilon_{ki} = \epsilon \frac{B_{ki}^2}{\left(\sum_j B_{kj}\right)^2} \tag{2.14}$$

where ϵ is a global scale parameter for background variance, to be estimated from the data. For the ith column, we compute the expected variance for the column as the average over

the atoms aligned to the column

$$\epsilon_i = \frac{1}{|\mathcal{M}_i|} \sum_{k \in \mathcal{M}_i} \epsilon_{k\mathcal{M}_{ik}} \tag{2.15}$$

Incorporating this into the structural drift model leads to a variance components model, with column i having covariance $\Sigma^{(i)} = \Sigma_{\mathcal{M}_i} + \epsilon_i I_{|\mathcal{M}_i|}$.

2.2.6 Uncorrelated structural perturbations (non-phylogenetic structural model)

In the limiting case as $\sigma_k^2,\ \theta_k \to 0$, keeping the ratio $\frac{\sigma_k^2}{2\theta_k} = \tau$ fixed, all structural deviation is explained via ϵ, and the marginal distribution of the observed data in the ith column is

$$C_{ij}^{(\mathcal{M}_i)} \mid \mathcal{M}, \tau, \epsilon, \Upsilon \sim \mathsf{N}_{|\mathcal{M}_i|}(0, \Sigma^{(i)}) \tag{2.16}$$

where $\Sigma_{kl}^{(i)} = \tau$ if $k \neq l$, and $\Sigma_{kk}^{(i)} = \tau + \epsilon_i$. This is similar to the non-evolutionary Bayesian structure alignment models described above (Wang and Schmidler, 2014), where structural perturbations are independent of evolutionary distance. In this limiting model, the structural likelihood does not depend on the tree nor on the evolutionary parameters, and structural information only indirectly affects the distribution over trees via the effect on the alignment.

2.2.7 Rotations and translations

Up to this point we have assumed that the data consist simply of a set of three-dimensional coordinates. However, the coordinates of each structure are recorded with respect to an arbitrary reference frame, and the likelihood is not invariant to transformations of the coordinate system. This can be addressed without compromising the reversibility of the model by introduction of auxiliary rotation and translation random variables for each structure, as

discussed in Challis and Schmidler (2012). Since the OU process is symmetric and hence invariant to rotations of the coordinate system, we can omit the rotation for an arbitrarily chosen reference protein; this reference protein still has an associated translation, such that the likelihood is independent of the choice of reference. With independent uniform priors over rotations and translations, reversibility is maintained (Challis, 2013), and the resulting posterior is proper.

2.2.8 Priors

In order to complete the specification of the full Bayesian model, it is necessary to assign prior distributions to each unknown parameter. In general, we opt for diffuse priors, reflecting our lack of strong prior knowledge regarding the parameters, using standard conjugate priors where available.

2.2.8.1 Alignment and tree parameters

We assume a uniform prior on tree topologies, since we typically have no data-independent information about the topology. The prior on alignments is induced by the indel model parameters and their priors. For branch lengths, we use a diffuse $\mathsf{Exp}(0.01)$ prior. Although previous studies have occasionally observed slow convergence resulting from diffuse branch length priors due to the flatness of the likelihood surface at large distances (Challis, 2013), we did not observe such issues on any of the datasets used in this work, and the posterior distributions for branch lengths showed very little dependence on the choice of prior (*see Figure 2.4*). Furthermore, we do not observe any issues relating to the induced Gamma prior on total tree length as discussed by (Rannala *et al.*, 2012), with the posterior tree length also showing very little dependence on the choice of prior (*see Figure 2.5*).

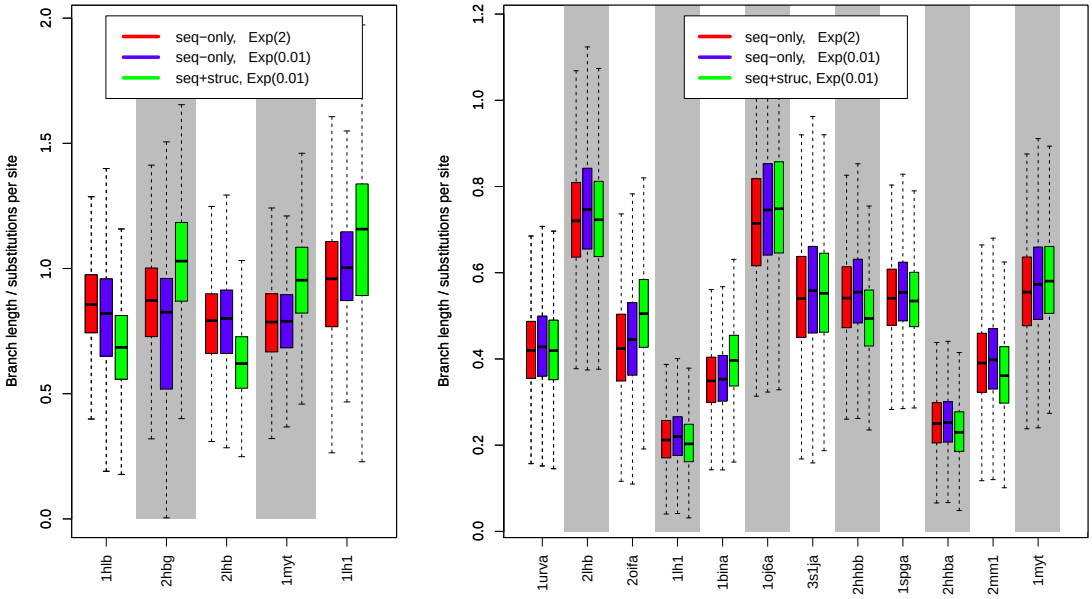


Figure 2.4: Posterior distributions for branch lengths are typically not sensitive to the choice of exponential prior. This is especially true for larger datasets. Shown here are branch lengths for the tips of the tree for the 5-globin (left) and 12-globin (right) datasets.

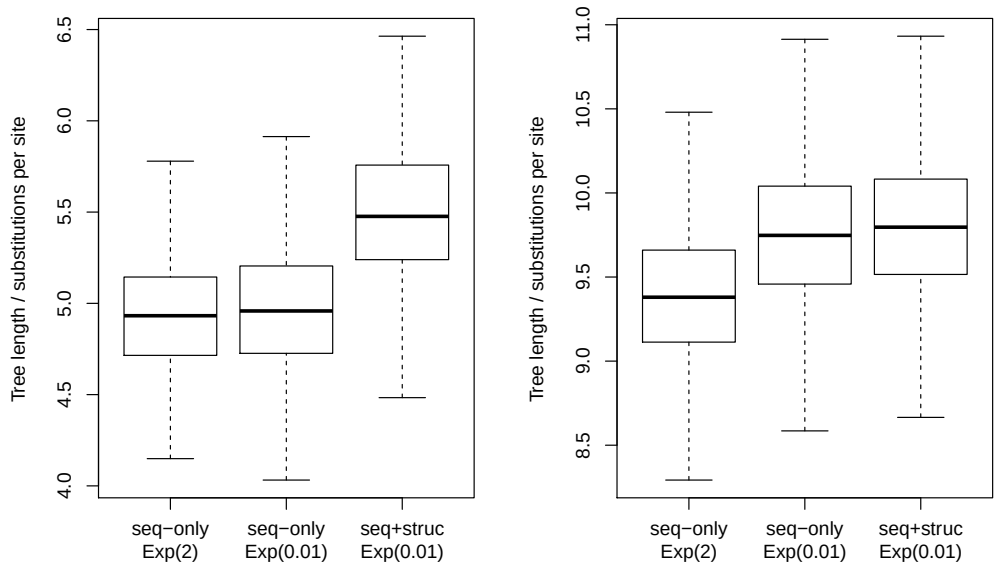


Figure 2.5: The posterior distribution for total tree length is only very weakly influenced by the choice of prior. Shown here are tree lengths for the 5-globin (left) and 12-globin (right) datasets.

2.2.8.2 Substitution parameters and indel model parameters

In the analysis considered here use Dayhoff substitution rate matrix (Dayhoff *et al.*, 1978). It is possible to estimate parameters of a more general substitution model during inference, but in the current analysis we keep these parameters fixed for reasons of computational efficiency.

The TKF92 model parameters are assigned the following prior specification

$$\lambda \sim \mathsf{Gamma}(a_\lambda, b_\lambda)$$

$$\mu \sim \mathsf{Gamma}(a_\mu, b_\mu)$$

$$r \sim \mathsf{Beta}(a_r, b_r)$$

In the analyses conducted here, the hyperparameters are set to $a_\lambda = b_\lambda = a_\mu = b_\mu = 1$, resulting in $\mathsf{Exp}(1)$ priors for λ and μ, and $a_r = b_r = 1$, resulting in a $\mathsf{Unif}(0, 1)$ prior for r. Although λ and μ will typically have a value somewhat lower than 1, we favour the $\mathsf{Exp}(1)$ prior over a prior more concentrated around zero in order to ensure that the effect of the prior be more similar across the range of probable values for λ and μ.

2.2.8.3 Priors for structural parameters

Rotations and translations are given uniform priors, as no rotation or translation is favoured *a priori*. Since the likelihood is not invariant to overall translations of the coordinates, the posterior remains proper despite the improper prior on translations. For the other structural parameters we use

$$\tau \sim \mathsf{InvGamma}(a_\tau, b_\tau)$$

$$\epsilon \sim \mathsf{Gamma}(a_\mu, b_\mu)$$

$$\sigma^2 \sim \mathsf{Gamma}(a_\sigma, b_\sigma)$$

with hyperparameters $a_\tau = b_\tau = 0.001$, $a_\epsilon = b_\epsilon = 2$, and $a_\sigma = b_\sigma = 1$ yielding weakly informative priors reflecting our knowledge about the expected magnitude of structural fluctuations.

2.2.8.4 Shrinkage prior for branch-specific diffusivity

With a separate drift rate for each branch, there might be concern that the structural drift model could be overparameterised (Dutheil *et al.*, 2012; Groussin *et al.*, 2013). To address this possibility, we adopt a shrinkage-favouring mixture prior for the branch-specific σ_k^2 parameters:

$$\sigma_k^2 \mid \sigma_g^2, \nu \sim \gamma\delta(\sigma_k^2 - \sigma_g^2) + (1-\gamma)\mathsf{LogN}(\log \sigma_g^2, \nu) \tag{2.17}$$

with $\sigma_g^2 \sim \mathsf{Gamma}(a_g, b_g)$ and $\nu \sim \mathsf{Gamma}(a_\nu, b_\nu)$. This setup allows for pooling of information about σ_g^2 from all branches, while maintaining the flexibility of individual rates for each branch, as well as allowing for some degree of variable selection when appropriate. We set $a_g = 1, b_g = 2$, and $a_\nu = 1, b_\nu = 6$.

When $\gamma = 1$, all σ_k parameters are shrunk to the global mean, whereas $\gamma = 0$ yields the fully branch-specific model. For $0 < \gamma < 1$, the σ_k parameters that lie close to the global mean are shrunk strongly to σ_g. This additional shrinkage beyond the basic hierarchical prior is useful in larger trees where the internal branch drift parameters may have high uncertainty, particularly when the corresponding branches are very short. In the limit of $\nu \to 0$ (i.e. very low variance among the σ_k^2 parameters), the two components of the mixture become essentially identical. For larger ν, diffusivity parameters close to the global σ_g^2 are shrunk more strongly to the spike density.

For smaller trees we fix $\gamma = 0$; for larger trees γ is inferred from the data, using a $\mathsf{Beta}(a_\gamma, b_\gamma)$ prior. When high levels of shrinkage are desired, we use $a_\gamma = 1.35$ and $b_\gamma = 1.1$; this leads to a prior mode of $\gamma = 0.78$, favouring shrinkage of most of the

σ_k^2 parameters to the global σ_g^2, but still places around 42% of the prior density for γ below 0.5, and 5% below 0.1, allowing for the prior to be overruled when strong evidence exists for heterogeneity among the branch-specific diffusivity parameters.

To carry out inference under this prior for γ, we employ a standard data augmentation scheme, with indicator variables z_k for inclusion of σ_k^2. In this augmented model, we then have

$$z_k \sim \mathsf{Bernoulli}(\gamma)$$
$$\sigma_k^2 \mid \sigma_g^2, \nu, z_k \sim z_k \delta(\sigma_k^2 - \sigma_g^2) + (1 - z_k)\mathsf{LogN}(\log \sigma_g^2, \nu)$$

Integrating over z_k this yields the original mixture model (Diebolt and Robert, 1994). To improve mixing, we can integrate out γ from this augmented model, yielding a Beta-Binomial prior for z

$$p(z \mid a_\gamma, b_\gamma) = {}^n\mathbb{C}_m \frac{B(a_\gamma + m, b_\gamma + n - m)}{B(a_\gamma, b_\gamma)}$$

where n is the number of branches in the tree, $m = \sum_k z_k$ is the number of free σ_k^2 parameters, $a_\gamma, b_\gamma > 1$, and $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$ is the Beta function.

2.3 MCMC inference

Calculations of posterior distributions are performed by MCMC sampling. Since the joint posterior over alignments, topology, and parameters can be complicated, careful design of the MCMC algorithm is essential, and we have developed a number of specialised moves to increase the efficiency of convergence and mixing.

Continuous parameters, i.e. (Θ, Φ, Λ) plus the branch lengths of the tree, are updated using random walk Metropolis updates after appropriate transformations, and tree topologies are proposed using a combination of stochastic nearest-neighbour interchanges and

the LOCAL move of Larget and Simon (1999) with the acceptance ratio given in Holder *et al.* (2005). Alignments are resampled using a window-based progressive dynamic programming scheme to generate proposals, correcting the acceptance ratio by the ratio of likelihoods under the full model. The scheme is similar to the approach outlined in Miklós *et al.* (2008), augmented to include the structural likelihood.

Under the shrinkage prior described in equation 2.17, the z_k and σ_k^2 variables are sampled together according to the scheme shown in Algorithm 1. This formulation retains the same model dimensionality at each step, since σ_k^2 still remains in the model when $z_k = 1$, albeit with its domain restricted to a point mass at σ_g^2. It is also possible to formulate the same scheme in terms of the reversible jump formalism (Green, 1995), whereby setting $z_k = 1$ corresponds to removing a parameter from the model. In the latter formulation, an auxiliary variable, u_k, is used in place of the absent variable σ_k^2 when $z_k = 1$, but leads to the same acceptance ratios (*cf.* §*3.1 of Hastie and Green (2012)*). However, since the fixed σ_k^2 parameters still contribute an independent Gamma prior density, it is more appropriate to regard them as fixed rather than absent, although the effective size of the model will decrease when the number of free parameters is decreased.

The z_k parameters are held fixed to 1 during the first half of the burn-in, while the tree topology is still converging, and allowed to switch thereafter. Similarly, sampling of ν is omitted during the first half of the burn-in, and also during the second half when $n - m \leq 1$ (i.e. when the number of non-fixed parameters is 1 or 0). This helps to avoid ν becoming very close to zero in the period where the proposal variance is being automatically tuned (which can lead to it potentially getting stuck near zero). When $n - m \leq 1$ during the second half of the burn-in, with probability 0.2, z_k is also allowed to switch to 0 without resampling σ_k^2, to allow for parameters to be shifted to the log-normal component even when ν has become small enough that the non-spike density at the mode is higher than the spike density.

It is possible to use a scheme whereby $q(z_k' \mid z_k) = \delta(z_k + z_k' - 1)$, such that an indicator

Algorithm 1 Scheme for sampling indicator variables, z_k, and branch-specific diffusivity parameters, σ_k^2, under the shrinkage prior described in equation 2.17.

Sample k uniformly at random
Propose $z_k' \sim q(z_k' \mid z_k)$
if $z_k' = z_k = 1$ **then**
  Accept the move with no change to σ_k^2
else
  Propose $\sigma_k^{2\prime} \sim \begin{cases} \delta(\sigma_k^{2\prime} - \sigma_g^2) & (z_k' = 1) \\ q(\sigma_k^{2\prime} \mid \sigma_k^2) & (z_k' = 0) \end{cases}$
  Accept the move with probability

$$\min\left\{1, \frac{p(\,\cdot \mid \sigma_k^{2\prime}, \ldots)}{p(\,\cdot \mid \sigma_k^2, \ldots)} \frac{p(z' \mid a_\gamma, b_\gamma)}{p(z \mid a_\gamma, b_\gamma)} \times \frac{q(z_k \mid z_k')}{q(z_k' \mid z_k)} \frac{q(\sigma_k^2 \mid \sigma_k^{2\prime})^{(1-z_k)}}{q(\sigma_k^{2\prime} \mid \sigma_k^2)^{(1-z_k')}}\right\}$$

where $\dfrac{p(z' \mid a_\gamma, b_\gamma)}{p(z \mid a_\gamma, b_\gamma)} = \begin{cases} 1 & (z_k = 0, z_k' = 0) \\ \frac{n-m}{m+1} \frac{a_\gamma + m}{b_\gamma + n - m - 1} & (z_k = 0, z_k' = 1) \\ \frac{m}{n-m+1} \frac{b_\gamma + n - m}{a_\gamma + m - 1} & (z_k = 1, z_k' = 0) \end{cases}$

and $p(\,\cdot \mid \sigma_k^2, \ldots)$ is the likelihood of the data as a function of the kth diffusivity parameter.

flip is attempted for every move. However, in order to improve mixing on σ_k^2 we favour a stochastic scheme whereby σ_k^2 can be resampled without flipping the indicator when $z_k = 0$. In our applications we use an independence proposal for the indicator variables, of the form $z_k \sim \text{Bernoulli}(a_\gamma/(a_\gamma + b_\gamma))$, such that $q(z_k \mid z_k')/q(z_k' \mid z_k) = (a_\gamma/b_\gamma)^{(z_k - z_k')}$. In the examples used here, $q(\sigma_k^{2\prime} \mid \sigma_k^2)$ is set to be a log-normal density, with variance v_k as automatically determined during the burn-in (*see Section 3.1.2*).

Figure 2.6 shows an example trace for the z parameter for a 12-taxon tree (analysed in more detail in Section 2.4), with instances of $z_k = 1$ indicated in red, for a set of 50,000 MCMC samples generated using the scheme described above. In this case, most of the internal branches end up strongly shrunk to the global σ_g^2 parameter, and there is good mixing across all components of the z vector aside from those branches for which σ_k^2 is significantly less than or greater than σ_g^2, as expected.

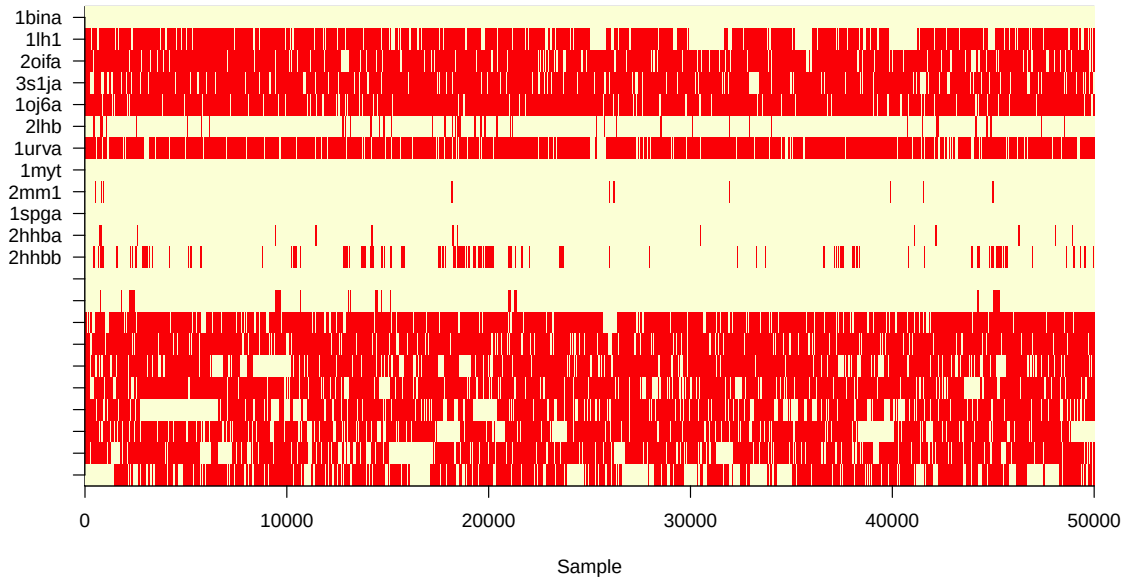


Figure 2.6: Example trace plot illustrating switching between components when using the hierarchical spike mixture prior. Figure generated using the scheme in Algorithm 1, for the 12-taxon globin dataset described in Table 2.2. Each row corresponds to a branch in the tree, with labelled rows denoting leaf branches, and unlabelled rows denoting internal branches.

2.3.1 Sampling rotations and translations

Although the rotations and translations would ideally be integrated out of the model analytically, this typically leads to marginal likelihoods that are complicated functions of the unknown ancestral structures, even for uncorrelated Gaussian noise models (Goodall and Mardia, 1993). Hence we sample rotations and translations using the scheme described by Challis and Schmidler (2012).

An MCMC scheme in which rotations are proposed only to individual proteins may become stuck in a state whereby subsets of proteins are well-aligned and rotated to each other, but with poor alignment and superposition between groups. Preliminary observations using such a scheme showed poor convergence in some cases. To remedy this, we introduced a sampling move that proposes a rotation to an entire subtree and simultaneously realigns the root of the subtree to its parent. Without the alignment proposal, the subtree rotation may be rejected due to movement of amino acids that are currently aligned to proteins in the rest of the tree. The combination of subtree rotation and realignment of the subtree root to its parent successfully alleviates this problem, and leads to faster average convergence times for the alignment and rotations/translations. We also make use of joint proposals that combine the various moves mentioned above, in order to help improve convergence.

2.3.2 Monitoring convergence

All MCMC simulations reported used four independent chains with randomised initial conditions. The overall likelihood and all scalar parameters were monitored for convergence using Gelman-Rubin potential scale reduction factors. For tree topologies, we monitored the stability of clade probabilities in the consensus tree, computing the average standard deviation of split frequencies (ASDSF) as an overall measure; for alignments, we monitored convergence of alignment length and stabilisation of the minimum-risk summary alignment (*see Chapter 4*), along with the associated marginal probabilities for each column.

2.4 Results and model comparison

To investigate the benefits of the structural model, we focused on datasets with highly divergent sequences, for which sequence-based analysis leaves significant uncertainty. We devote particular attention to the well-studied globins as a test case (*Table 2.2*); previous attempts to reconstruct the evolutionary history for this family using sequence data have yielded trees with high uncertainty. We also examine a set of cysteine proteinases (*Table 2.3*), which further demonstrate the utility of structural information in reducing uncertainty in alignments and topologies, while also providing insight into patterns of structural divergence.

To assess the accuracy of parameter estimation (including topologies and alignments), data were simulated from the structural drift model, with the modification that inserted residues were placed at the midpoint of their two neighbours, in order to avoid unrealistic bond lengths. The structure at the root was set to be equal to the human haemoglobin 2DN2, and model parameters were chosen based upon typical values observed on test runs on small globin datasets: $\sigma_k^2 = 0.7$, $\lambda = 0.03$, $\mu = 0.0305$, $r = 0.67$. All B-factors were set to be equal to 1 for simplicity, and ϵ was varied over the set $\{0, 0.5, 1.0, 2.0\}$. Three different tree topologies were used, with 6, 8, and 10 leaves respectively, and for each topology, branch lengths were multiplied by two different scale factors (1.0 and 2.0) in order to yield varying levels of divergence. Each parameter combination was simulated ten independent times, and results averaged over the ten replications.

For each dataset, we perform analysis using the sequence-only model, and the phylogenetic (ϵ, σ^2) and non-phylogenetic (ϵ-only) variants of the structural model, in order to assess the effect of including structural information.

Structure	Protein	Organism	Resolution	*R*-value	Length*
2oif	NsGb	H. vulgare (barley)	1.80	20.2	153
1bin	Lhb	G. max (soybean)	2.20	19.8	143
1lh1 ⋆	Lhb	L. luteus (lupin bean)	2.00	27.3	153
1oj6 ‡	Ngb	H. sapiens (human)	1.95	17.8	147
3s1j	HGbI	M. infernorum (thermophile)	1.80	21.0	131
1urv ‡	Cygb	H. sapiens (human)	2.00	22.2	154
2lhb ‡, ⋆	CycHb	P. marinus (lamprey)	2.00	14.2	149
1myt ‡, ⋆	Mb	T. albacares (tuna)	1.74	17.7	146
2mm1 ‡	Mb	H. sapiens (human)	2.80	15.8	153
1spga ‡	α-Hb	L. xanthrus (spot croaker)	1.95	19.1	143
2hhba ‡	α-Hb	H. sapiens (human)	1.74	16.0	141
2hhbb ‡	β-Hb	H. sapiens (human)	1.74	16.0	146
2hbg ⋆	Hb	G. dibranchiata (bloodworm)	1.50	12.7	147
1hlb ⋆	Hb	C. aurenicola (sea cucumber)	2.50	15.0	157

Table 2.2: The 5-, 8- and 12-globin datasets, grouped according to observed clades. Sequences marked with a ‡ are present in the 8-globin dataset, those with a ⋆ in the 5-globin set, and all except 2hbg and 1hlb are present in the 12-globin set. NsGb = non-symbiotic plant globin; Lhb = leghaemoglobin; Ngb = neuroglobin; HGbI = bacterial Hell's gate globin I; Cygb = cytoglobin; CycHb = cyclostome haemoglobin; Hb = haemoglobin; Mb = myoglobin. $*$ - length shown for the portion present in the PDB file.

Structure	Protein	Organism	Resolution	*R*-value	Length*
1aim	Cruzain	T. cruzi (trypanosome)	2.00	18.8	216
8pcha	Cathepsin H	S. scrofa (wild boar)	2.10	NA	221
1mema	Cathepsin K	H. sapiens (human)	1.80	18.3	216
2acta	Actinidin	A. chinensis (kiwi fruit)	1.70	16.5	219
1cqda	Proteinase II	Z. officinale (ginger)	2.10	21.3	217
1yal	Chymopapain	C. papaya	1.70	19.2	217
1ppn	Monoclinic papain	C. papaya	1.60	16.0	213
1gece	Glycyl peptidase	C. papaya	2.10	19.6	217
1ppo	Protease omega	C. papaya	1.80	15.5	217

Table 2.3: The cysteine proteinase dataset. Average pairwise identity using the HOMSTRAD alignment is 42%. $*$ - length shown for the portion present in the PDB file.

2.4.1 Structural information improves alignments

For the simulated datasets the true multiple alignment is known, and we can measure the distance of the posterior alignment samples to this known alignment using the *column score* (proportion of correct columns), and the *sum-of-pairs score* (proportion of correct pairwise homology statements (Thompson *et al.*, 1999)). The alignment accuracy metrics are averaged over the ten repetitions for each tree. Under the sequence-only model alignment accuracy decreases markedly as branch lengths increase; in contrast, with the structural models, alignment accuracy remains high (*Figure 2.7*).

On the 5-globin and cysteine proteinase datasets, alignment accuracy was measured with respect to the alignments contained in the HOMSTRAD database (Mizuguchi *et al.*, 1998), which were constructed using 48 (globin) and 13 (cysteine proteinase) structures. In each case, the addition of structural information results in a consistent improvement in alignment accuracy and decreased variability (Figure 2.7), as with the simulated data.

2.4.2 Structure reduces topological uncertainty

The 5-globin dataset was chosen as a simple test case to explore the effect of structural information on topology uncertainty. Results were generated from four independent runs of $100,000$ samples, thinned from $10m$ iterations, after a $5m$ burn-in. For sequence-only, on average around $80,000$ topology switches were observed during the $10m$ iterations. With the non-phylogenetic structural model included, around 2200 switches were observed, and with the phylogenetic structural drift model. around 700. The ASDSF values for the consensus trees were 0.009, 0.000 and 0.000 respectively (*Figure 2.9*).

The sequence-only model visits the most probable tree only 60.1% of the time, with 27.7% of the samples coming from a second topology (*Figure 2.9*). We also ran BAli-Phy (Suchard and Redelings, 2006) on this dataset, and the consensus tree yields a polytomy between 1lh1, 1hlb and 2hbg, indicating even higher posterior tree uncertainty under the BAli-Phy sequence-only evolutionary model (*Figure 2.10*).

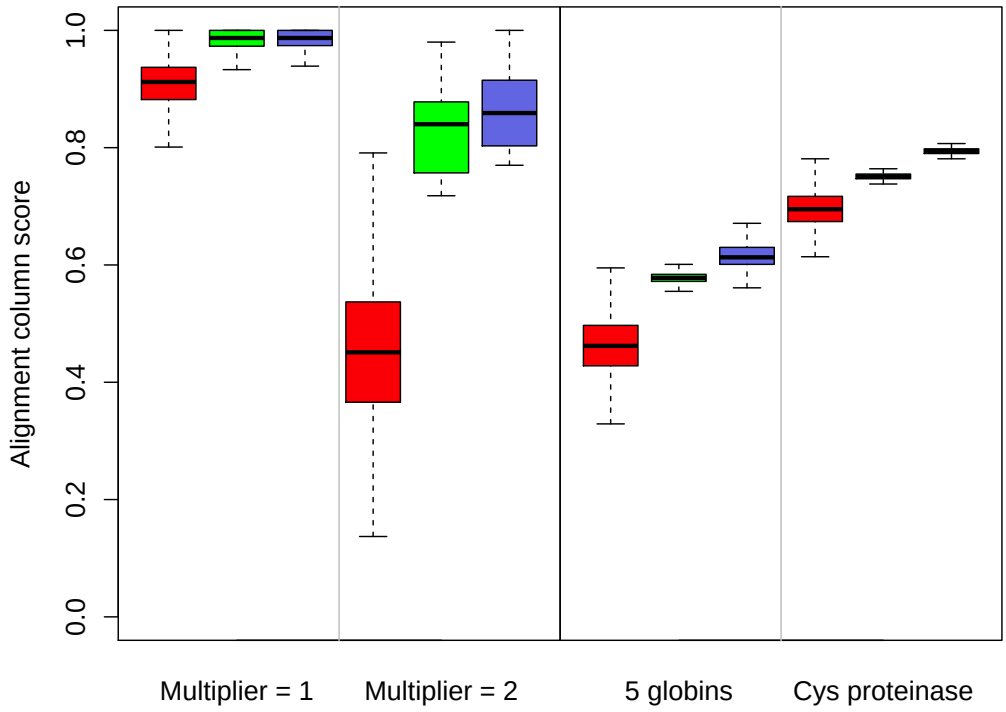
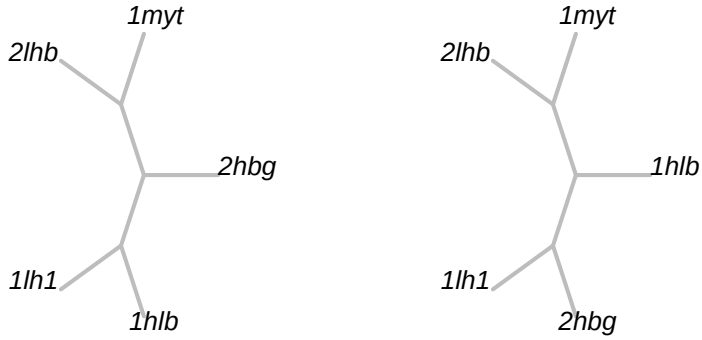


Figure 2.7: Alignment accuracy on simulated data (left two panels) for short branches (multiplier = 1) and long branches (multiplier = 2), and on the 5-globin and cysteine proteinase datasets (right panels). Shown are posterior distributions of distance to true alignment (simulated data) or HOMSTRAD alignment (globins and cysteine proteinases) obtained under the sequence-based model alone (red), and after combining with the non-phylogenetic (green) and phylogenetic (blue) structural models. In all cases alignments are more accurate with structural information than under the sequence-only model, with a much narrower range of accuracy values. In many cases the phylogenetic structural model also offers an additional improvement in alignment accuracy over the non-phylogenetic model. Simulated data results shown for ten realisations on an 8-taxon tree with $\sigma_k^2 = 0.7$ and $\epsilon = 0.5$, with branch lengths multiplied by the multiplier indicated. Similar results were seen with the sum-of-pairs alignment accuracy metric (not shown).



seq-only	0.606	0.277
seq+struc (*non-phylo*)	0.997	0.000
seq+struc (*phylo*)	0.999	0.000

Figure 2.8: The two most frequently sampled tree topologies for the 5-globin data set, with posterior probabilities shown under sequence-only and structural models. Trees shown with equal branch lengths for each branch for illustrative purposes. Posterior probabilities were computed using the program `trees-consensus`, written by Benjamin Redelings.

In contrast, under both structural model variants there is virtually no uncertainty in the topology, with more than 99% of the samples coming from the most probable topology, placing 2hbg (G. dibranchiata haemoglobin) in between the other four structures. Acceptance for nearest-neighbour topology moves was 4% for sequence-only, and less than 0.1% for the structural models, the latter reflecting the very low uncertainty in the topology when structural information is included.

These results clearly illustrate the ability of the joint sequence-structure model to concentrate the posterior around the most likely topology, indicating that additional information is contained within the structural portion of the model. This extra information can be incorporated with little additional computational cost in this case: the three model variants required the same number of iterations to achieve convergence, with the runtime of the structural models around 1.2-1.5 times that of the sequence-only model (*see Table 4.1*).

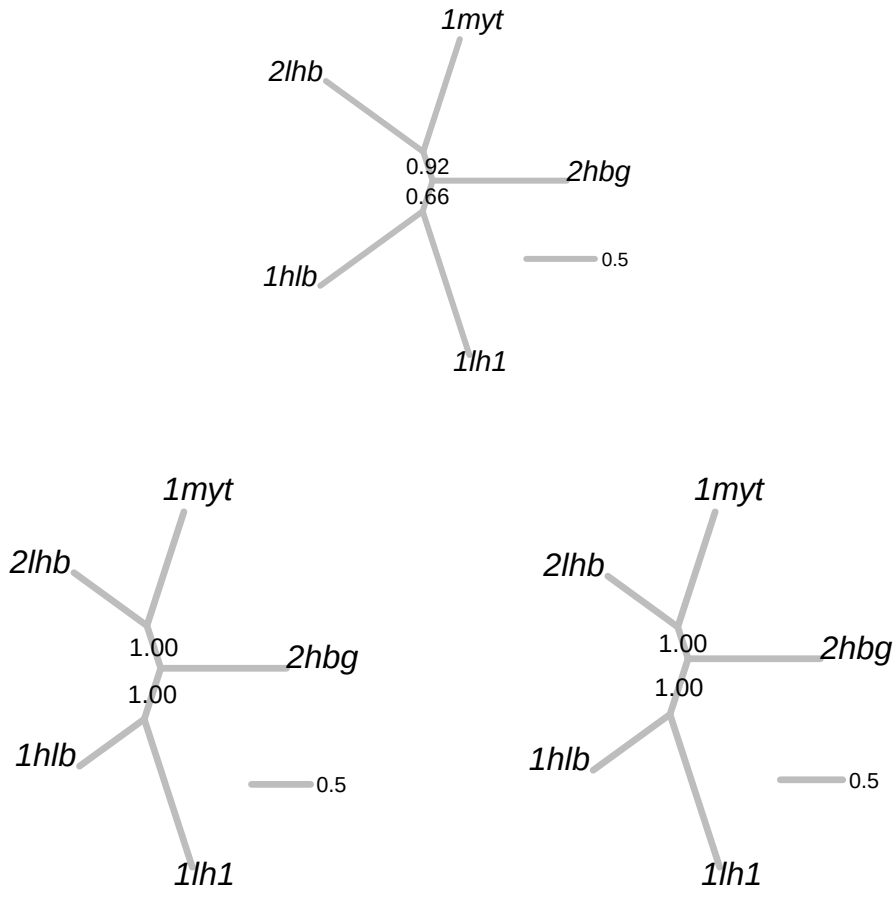


Figure 2.9: Consensus trees for the 5**-globin dataset.** Results shown computed under the sequence-only model (top), and with the non-phylogenetic (bottom left) and phylogenetic (bottom right) structural models. Results were generated from 10m MCMC iterations after a burn-in of 5m, sampling every 200 iterations. ASDSF values are 0.009, 0.000 and 0.000 for the three trees.

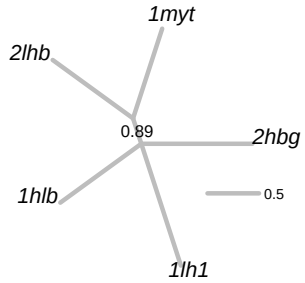


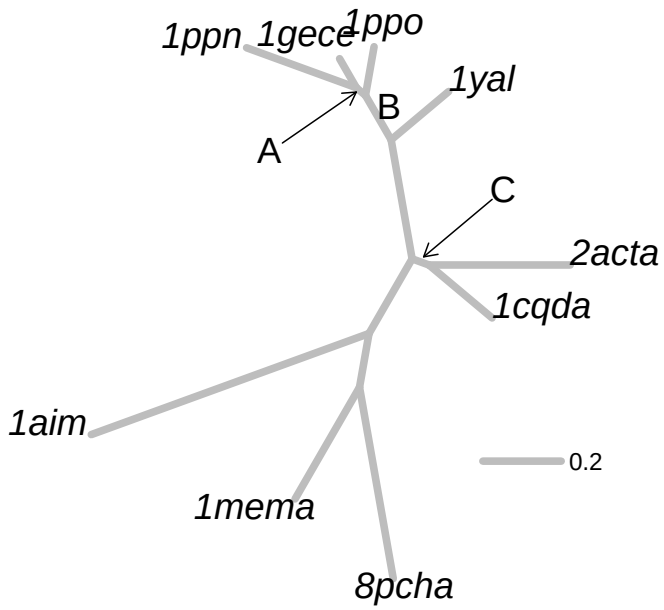
Figure 2.10: Consensus tree for the 5-globin dataset, derived using BAli-Phy with default settings. Convergence required 10,000 iterations, roughly 30 minutes' runtime on a 2.13Ghz Intel core, with burn-in set to 365 as recommended by the `statreport` utility.

		Opteron 2.3GHz	*Intel i3 3.3GHz*
5 globins	seq-only	1.0	1.8
	seq + struc	0.7	1.4
8 globins	seq-only	0.8	1.4
	seq + struc	0.6	1.2
12 globins	seq-only	0.7	1.2
	seq + struc	0.4	0.8

Table 2.4: Average number of MCMC iterations per hour (in millions) on different sized datasets, computed on two different CPU types.

Similar results are observed with the larger cysteine proteinase dataset (Figure 2.11). Again the structural consensus trees do not differ topologically from the sequence tree, and consensus branch lengths are very similar, but uncertain splits in the consensus tree are more highly resolved when structure is included. ASDSF = 0.000, 0.015, 0.019 for sequence-only, non-phylogenetic (ϵ-only), and phylogenetic (ϵ and σ^2) structural models respectively.

As discussed earlier, structural information can reduce topology uncertainty in at least three ways: by increasing alignment accuracy, by reducing alignment uncertainty, and by providing direct information regarding the topology and branch lengths. In the above cases, a decrease in topology uncertainty is also observed when the non-phylogenetic structural



	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 2.11: For the cysteine proteinases the consensus topology is the same under all model variants. The labelled edges correspond to splits with significant uncertainty under the sequence-only model (the other three splits have posterior probability 1.00 in all cases). The table below the figure shows the posterior probability of each of these labelled splits under the different model variants.

model is used, suggesting that alignment inaccuracy and/or uncertainty is a principal cause of topology uncertainty in these examples. Nevertheless, additional reductions in alignment and topology uncertainty are also seen from adding the phylogenetic drift component to the model (*Figures 2.7 and 2.11*).

2.4.3 Structural information reduces tree errors

For the simulated datasets where the true tree is known, we can also assess whether the structural model concentrates the tree posterior around the correct topology, using the Robinson-Foulds topology distance (Robinson and Foulds, 1981). For trees with smaller branch lengths, the sequence-only and sequence + structure models performed similarly, with the structural model only slightly more accurate. However, when branch lengths are doubled, the structural information not only reduces uncertainty, but also improves accuracy of the sampled topologies (*Figure 2.12*).

2.4.4 Structure helps select between alternative topologies

In cases where the majority of the tree is well resolved, the structural model often favours the same consensus tree as sequence. However, for trees with higher uncertainty, structure can also help to select between alternative hypotheses in regions that are difficult to resolve. Here we illustrate this by analysing a larger set of globins (*Table 2.2*).

The known set of vertebrate globin types was expanded relatively recently with the discovery of two additional globins: the neuroglobin (Burmester *et al.*, 2000) and cytoglobin (Burmester *et al.*, 2002). Neuroglobin tends to occur in neurons and endocrine cells, while cytoglobins appear in fibroblast-related cell types, and have been observed to be present in all vertebrates. The function of both proteins is still somewhat unclear, although high levels of sequence conservation suggest a vital physiological function for cytoglobin (Hoffmann *et al.*, 2012b).

Since these discoveries, there has been a surge of interest in establishing the likely

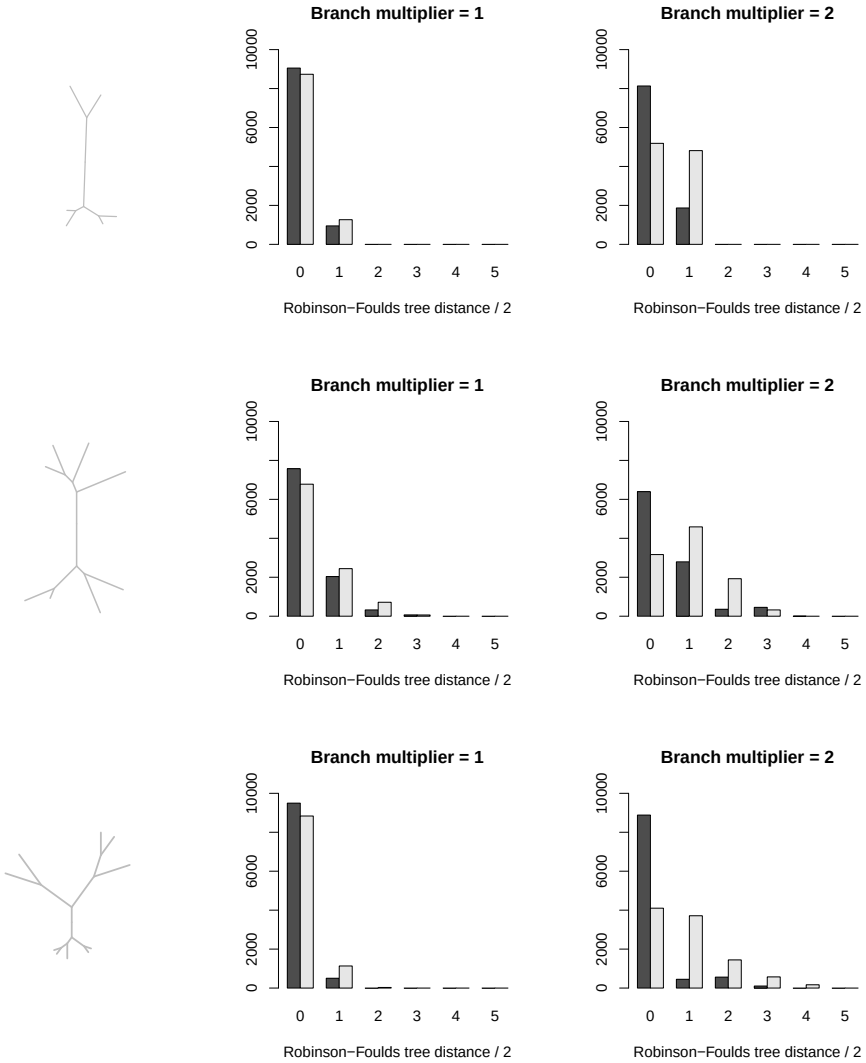


Figure 2.12: Posterior distribution of topology errors relative to the true tree for simulated data. Results shown for analysis under the phylogenetic structural model (black) and the sequence-only model (grey), as branch lengths are doubled (left to right). The inclusion of structural information allows the tree to be accurately inferred even for large evolutionary distances, whereas the trees inferred by the sequence-only model become much less accurate. Frequencies shown for the trees on the left, with 6 (top), 8 (middle), and 10 (bottom) leaves, aggregated from 10 independent samples from the model; the maximal half Robinson-Foulds distance for a tree with n leaves is $2(n-3)$, i.e. 3, 5 and 7 for the three trees above.

evolutionary history of the four vertebrate globin types: haemoglobin (Hb), myoglobin (Mb), neuroglobin (Ngb), and cytoglobin (Cygb). All previous analyses have found Ngb to be the most distant outgroup, so we focus here on the order in which the other vertebrate globins split after diverging from the neuroglobins.

Initial phylogenetic studies of Cygb using maximum likelihood approaches suggested the topology (Ngb, (Hb, (Mb, Cygb))) (Burmester *et al.*, 2002), although the support for this arrangement was found to be low. This topology may have initially appeared more plausible, since it requires O_2 transport to have evolved only once, along the branch to Hb. However, close homology was subsequently discovered between Cygb and the Hbs found in the jawless fishes known as cyclostomes (abbreviated as CycHbs). Accounting for this relationship requires either double evolution of O_2 transport function, or double loss of this functionality, as discussed by Hoffmann *et al.* (2010). Based on Bayesian phylogenetic analysis, the authors proposed the same phylogeny as Burmester *et al.* (2002), but with CycHb splitting from Cygb, i.e. (Ngb, (Hb, (Mb, (Cygb,CycHb)))), as shown in the top-left tree in Figure 2.13. Under this scenario, oxygen transport functionality is proposed to have developed independently in the cyclostome Cygb, the ancestor of the current CycHb, with the orthologues of the Mb and Hb genes subsequently lost (Hoffmann *et al.*, 2010, 2012b; Storz *et al.*, 2013).

More recently Hoffmann *et al.* (2012a) conducted a Bayesian analysis on a larger dataset including globins from plants, and in this case reported a three-way split, i.e. (Ngb, (Hb,Mb,(Cygb,CycHb))) (as shown in the bottom left tree in Figure 2.13, which contains a polytomy at the centre). Using a similar dataset including plant globins (without CycHb), Ebner *et al.* (2010) were also unable to resolve this three-way split, reporting the same polytomy.

Here we compare the results obtained by Hoffmann *et al.* (2010, 2012a) with those from our structural model, as well as the sequence-only indel model. To do so, we construct smaller versions of the two datasets, containing one or two representatives from each of the

clades of interest (*details in Table 2.2*). The first dataset is the 8-globin set containing only Hb, Mb, Cygb, Ngb and CycHb, and the second dataset contains an additional four proteins, namely three plant globins and a recently-crystallised bacterial globin known as *Hell's gate*, which has been observed to show high structural homology with human neuroglobin (Teh *et al.*, 2011; Vázquez-Limón *et al.*, 2012).

Although the original analyses of Hoffmann *et al.* (2010, 2012a) used 68 and 110 sequences respectively, we obtain the same consensus tree from just 8 and 12 sequences using our sequence-only statistical alignment model (*see Figure 2.13*). However, as with the results of Hoffmann *et al.* (2012a), the addition of the plant globins appears to destabilise the consensus tree, favouring other topologies in the posterior.

Specifically, our sequence-only model shifts from having 94% posterior probability on the split (Cygb,CycHb), Mb | Hb in the 8-globin case, to favouring this less than 50% of the time when the plant globins are added. In the 12-globin case, the sequence-only model visits the following three topologies between the clades of interest:

1. (Mb,((Cygb,CycHb),Hb))
2. ((Cygb,CycHb),(Mb,Hb))
3. (Hb,((Cygb,CycHb),Mb))

with relative frequency 2:1:1. The third topology is the same as the consensus topology on the 8-globin set.

As noted by Hoffmann *et al.* (2012a), globins are relatively short proteins and thus limited in the information that can be provided about evolutionary history. Hence, there is good reason to believe that more accurate inference can be obtained by including other sources of information such as structure.

Indeed, as shown in Figure 2.14, the structural model favours topology 2 with almost 100% certainty regardless of whether the plants globins are added. This demonstrates that inference under the structural model is more robust to the choice of dataset. Moreover, we

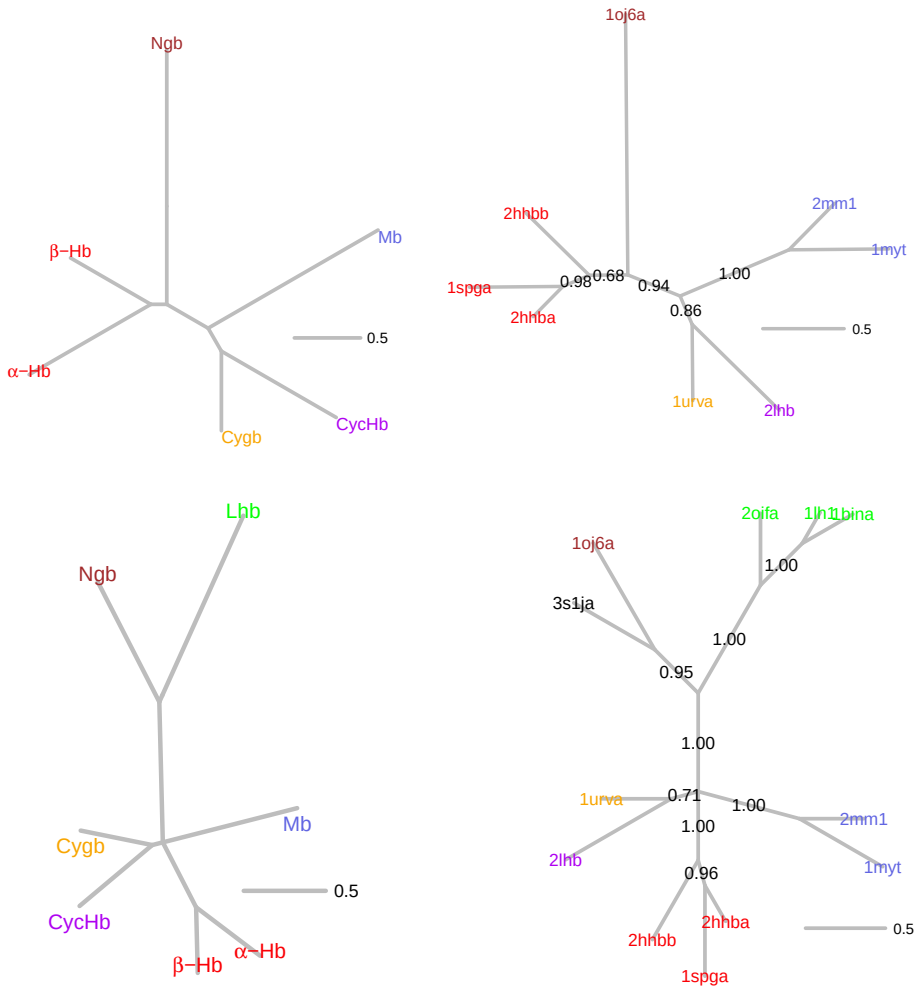


Figure 2.13: Consensus trees for globin datasets. These datasets are taken from Hoffmann *et al.* (2010) and Hoffmann *et al.* (2012a) (top left and bottom left respectively), and inferred using the sequence-only evolutionary model of Miklós *et al.* (2008) (top right and bottom right, ASDSF=0.011, 0.008 respectively). The bottom row features an augmented dataset containing plant globins, as well as a bacterial globin in our analysis. In both cases we obtain the same consensus tree as Hoffmann *et al.*, including the four-way polytomy in the 12-globin case.

can see that the sequence-only model is shifting to increasingly favour the structural tree as more sequences are included, illustrating the fact that structures can contain additional evolutionary information beyond what can be obtained from sequences alone.

Both structural models favour (CycHb,Cygb) as the first split from the root. It should be emphasised that in the non-phylogenetic (ϵ-only) structural model, only the alignment is directly informed by structural information (rather than evolutionary distance), which reiterates the fact that the alignment can have a large impact on the resulting phylogenetic inference. When phylogenetic structural drift is also included in the model, the posterior probability of (CycHb,Cygb) diverging before the Mb-Hb split increases further (from 0.72 to 1.00), demonstrating that the phylogenetic structural drift model does indeed allow for additional structural information to be used in estimating tree topologies.

2.4.5 Inclusion of structural inforamtion facilitates analysis of more challenging datasets

As a further example, we examined a larger set of globins, consisting of 17 structures, including four nerve globins, and two non-haem bacterial globins (*cf.* *Figure 2.15*). The evolutionary relationships between these structures are of great interest, since some of the splits may represent very ancient events, taking place before the divergence of plants and animals (Herman *et al.*, 2014f).

As shown in Figure 2.15, when analysed under the sequence-only model, the uncertainty associated with the phylogeny is very high, with many clades forming polytomies. In contrast, when structural information is included in the model, the uncertainty is reduced dramatically, allowing for more robust tree inference. The implications of the inferred phylogeny under the structural model on this dataset are currently being investigated in more detail (Herman *et al.*, 2014f), but the example shown here illustrates how the structural model has the power to extend the range of datasets that can be analysed.

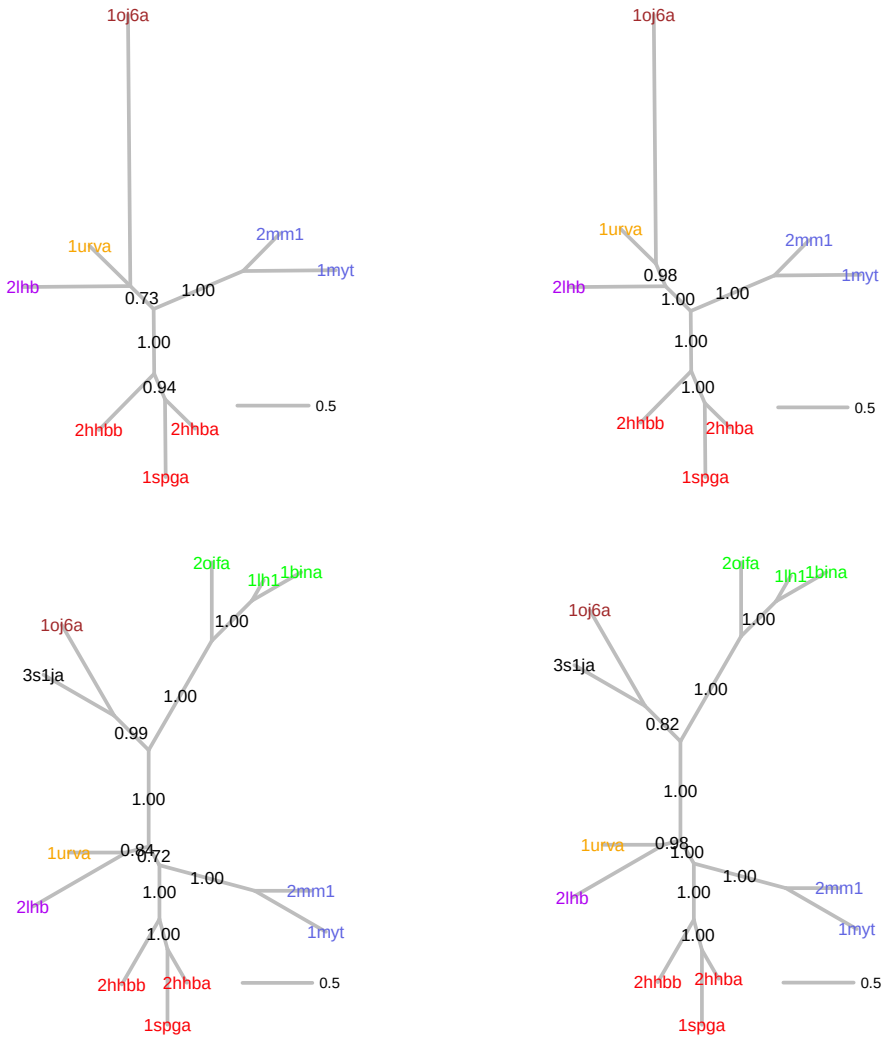


Figure 2.14: The structurally derived trees have very low uncertainty, and the order of the splits of interest is unchanged by the inclusion of additional sequences. Consensus trees derived under the non-phylogenetic (ϵ-only) structural model (top left and bottom left, ASDSF=0.010, 0.026 respectively), and the phylogenetic structural drift model (top right and bottom right, ASDSF=0.002, 0.016 respectively).

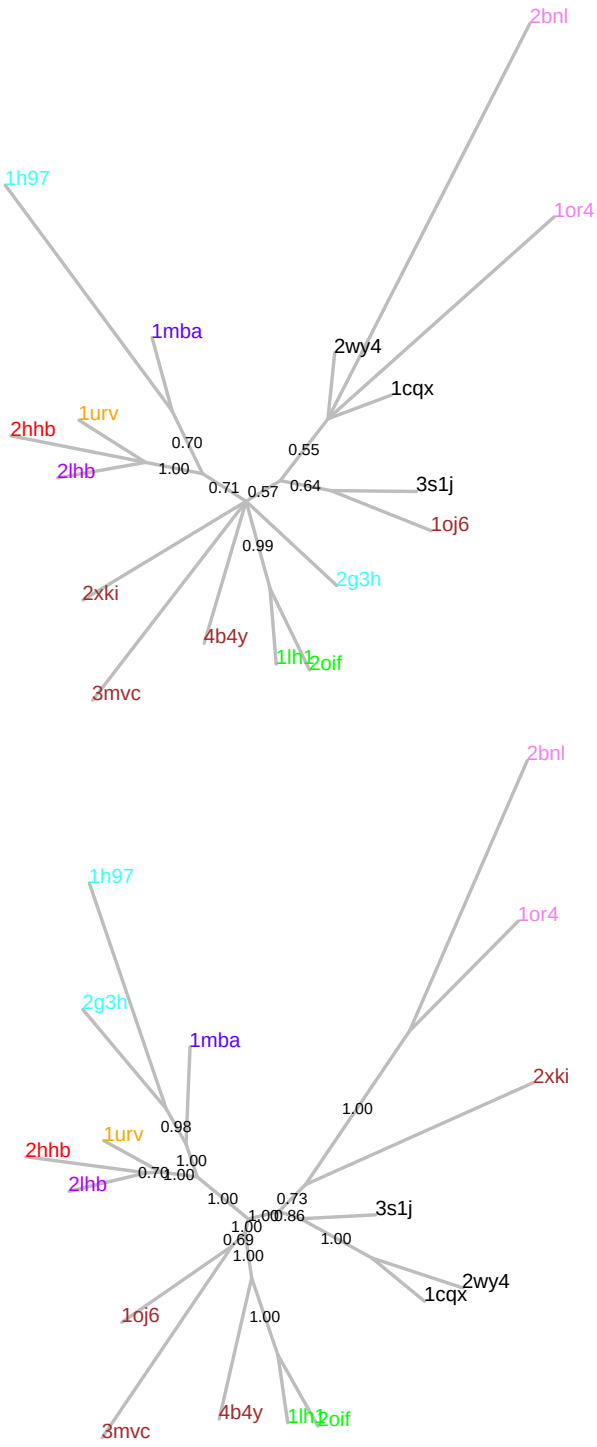


Figure 2.15: Inclusion of structural information allows for analysis of larger datasets that exhibit high uncertainty when analysed under a sequence-only model. On this larger globin dataset, the sequence-only model (top) results in a tree with a large number of polytomies, indicating high uncertainty. When structural information is included (bottom), all the splits are fully resolved, with much lower uncertainty. *Red = vertebrate Hb; purple = cyclostome Hb; orange = Cygb; cyan = protostome Hb; blue = vertebrate Mb; brown = nerve globin; green = plant globin; black = bacterial Hb; pink = bacterial non-haem globin.*

	8-globins		12-globins		Cys proteinase	
	non-phylo	phylo	non-phylo	phylo	non-phylo	phylo
P_V	150	140	258	229	226	213
DIC	16759	15959	25110	23743	18739	17075

Table 2.5: Effective number of parameters, P_V, and model fit as measured by *DIC* for structural models with and without a phylogenetic drift component. Results averaged over four independent repetitions for each dataset.

2.4.6 Phylogenetic structural model improves fit

As shown by the results in the previous sections, structural information is able to reduce topology uncertainty, concentrating the topology distribution around the posterior mode, as well as offering improvements in alignment accuracy. These improvements are often greater with the phylogenetic structural drift model than with the non-phylogenetic (ϵ-only) model.

In order to measure whether the phylogenetic model also achieves a better model fit to the data, we make use of the *deviance information criterion* (DIC) (Spiegelhalter *et al.*, 2002), given by $DIC = \mathbb{E}[D] + P_V$, where $D = -2 \log L$ is the *deviance*, and $P_V = Var[D]/2$ is a measure of the effective number of parameters in the model (Gelman *et al.*, 2003). Smaller values of DIC indicate a better model fit. The DIC measure is particularly suited to analysing the output of MCMC inference in hierarchical models when Bayes factors are not easily available (Spiegelhalter *et al.*, 2002). It should be noted that the effective number of parameters includes a contribution from the alignment and the tree, such that lower posterior uncertainty in these parameters will reduce the effective dimensionality of the model.

As shown in Table 2.5, despite increasing the actual number of parameters, the addition of phylogenetic drift rates for each branch reduces the overall uncertainty associated with the model, hence decreasing the *effective* number of parameters, P_V, and resulting in a substantial improvement in model fit, as measured by the DIC.

With complete shrinkage ($\gamma = 1$), the model retains only a single global σ_g^2, decrease the

effective number of parameters (on the 5-globin set this results in a reduction in average P_V from 148 to 140). However, the model fit generally suffers as a result (average DIC increases from 13,640 to 13,700 on the 5-globin dataset), and tends to result in trees with very different branch lengths from those obtained with sequence-only data. In contrast, the heterogeneous diffusivity model ($\gamma < 1$) results in a better model fit, and estimates branch lengths similar to those in the sequence-only trees. This suggests that branch-specific drift rates are indeed needed to explain the heterogeneity in the data. We examine this in more detail in Section 2.5.

2.4.7 Parameter inference

In addition to alignments and phylogenies, the model also provides the ability to estimate several scalar parameters of interest in the evolutionary process, such as indel rates and structural diffusivity coefficients.

On simulated data, the structural parameters are recovered to a high degree of accuracy, lying within the 95% highest posterior density interval in all cases, with the posterior median usually very close to the true value (*see Figures 2.16, 2.17 and 2.18*). Importantly, we are able to clearly resolve the different contributions from ϵ and σ even without repeated observations at the leaves.

Table 2.6 shows posterior quantiles for ϵ and σ_g^2 (the global diffusivity) on two globin datasets (with 8 and 12 taxons), and the cysteine proteinase dataset, under the non-phylogenetic (ϵ-only) and phylogenetic structural models. The phylogenetic drift model estimates $\sigma_g^2 > 0$ even with ϵ in the model, indicating that there is always a time-dependent component to the structural variation. ϵ is a multiplicative scale factor (in units of Å^2) for the site-specific variance parameters, which in our case are proportional to normalised B-factors. Hence, $\epsilon = 1$ signifies that an atom with B-factor equal to the mean has baseline variance equal to 1Å^2. The parameter σ_g^2 has units of Å^2 per substitution per site. For example, from the 12-globin set we expect phylogenetic drift to lead to an increase in mean square deviation

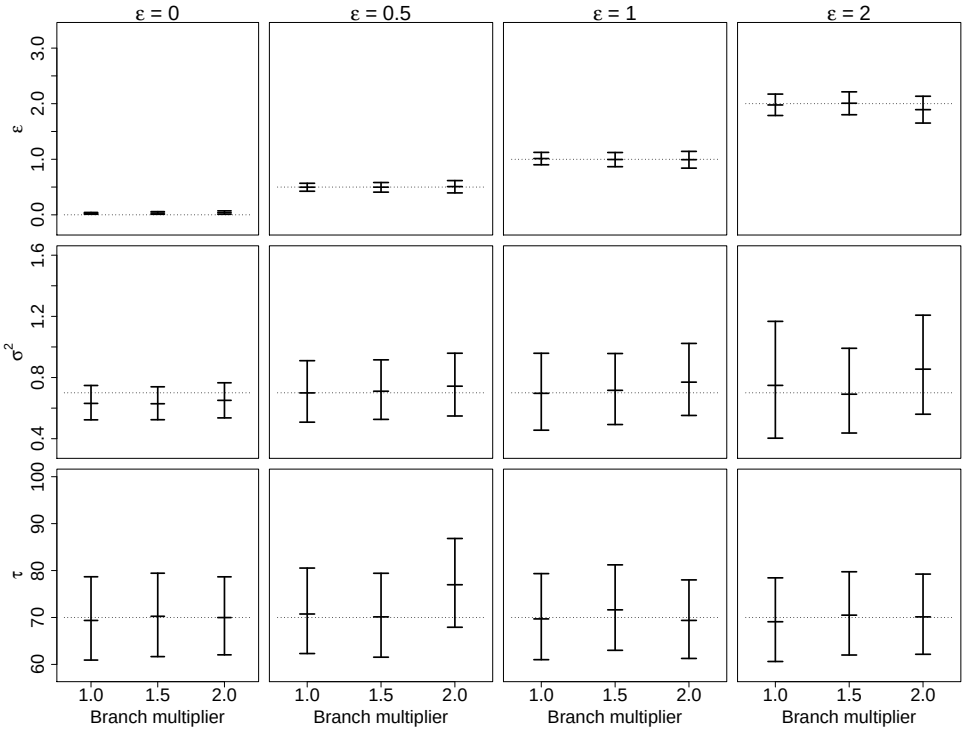


Figure 2.16: 95% highest posterior density intervals for structural model parameters estimated on simulated data, on a 4-leaf tree.

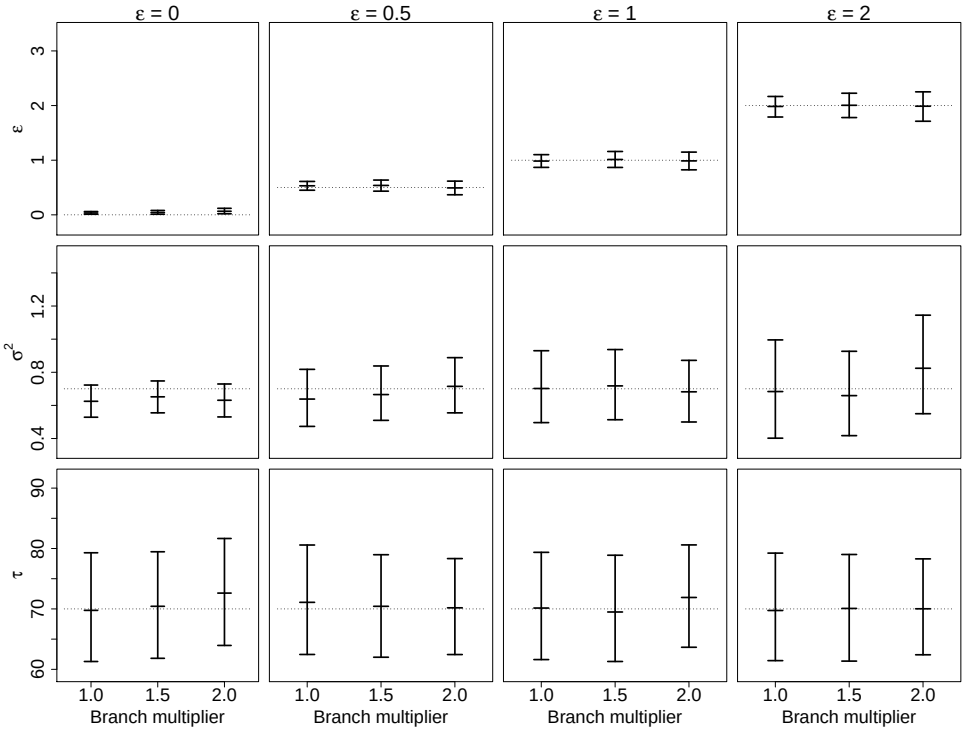


Figure 2.17: 95% highest posterior density intervals for structural model parameters estimated on simulated data, on an 8-leaf tree.

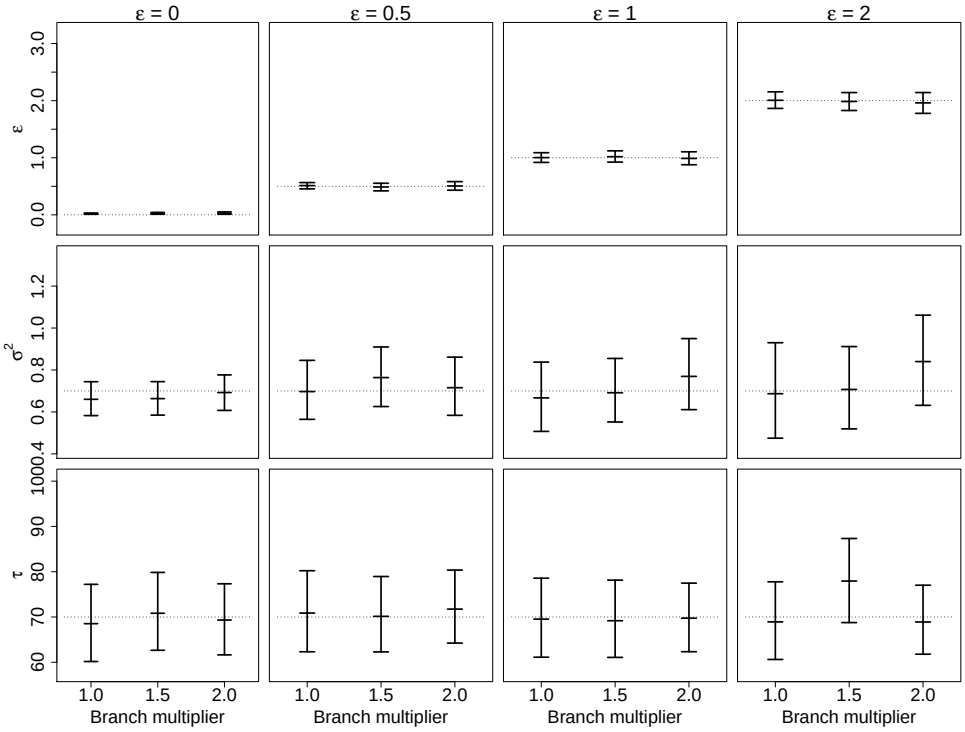


Figure 2.18: 95% highest posterior density intervals for structural model parameters estimated on simulated data, on a 10**-leaf tree.**

		8-globins		12-globins		Cys proteinase	
		non-phylo	phylo	non-phylo	phylo	non-phylo	phylo
$\widehat{\epsilon}$	5%	3.23	0.762	5.16	1.54	1.03	0.239
	50%	**3.53**	**0.902**	**5.78**	**1.76**	**1.09**	**0.275**
	95%	3.81	1.046	6.37	1.99	1.14	0.310
	GR	*1.02*	*1.00*	*1.49*	*1.00*	*1.01*	*1.04*
$\widehat{\sigma_g^2}$	5%	0	0.085	0	0.112	0	0.032
	50%	**0**	**0.192**	**0**	**0.232**	**0**	**0.049**
	95%	0	0.336	0	0.386	0	0.069
	GR	-	*1.00*	-	*1.00*	-	*1.01*

Table 2.6: Comparison of inference for global structural parameters on three datasets under the phylogenetic and non-phylogenetic variants of the model, averaged over four repetitions from independent starting points. Gelman-Rubin potential scale reduction factors (GR) are shown below each column. In the cysteine proteinase case, most of the variability is explained by baseline variance rather than evolutionary drift, although drift coefficients are significantly higher in certain regions of the tree (not shown).

of approximately 0.23Å^2 per substitution per site (*see Table 2.6*), although there are also noticeable hetereogeneities in drift rates across the tree.

In all cases Gelman-Rubin (GR) potential scale reduction factors were very close to 1, except for the non-phylogenetic (ϵ-only) model on the 12-globin dataset, since a single ϵ parameter struggles to explain the variability in this dataset, leading to slow convergence. In the cysteine proteinase case, although the global σ_g^2 is estimated to be very low (around 0.05), some branch-specific diffusivity coefficients are estimated to be substantially higher, hence there is still a substantial improvement in model fit using the phylogenetic structural drift model in this case (*Table 2.5*).

Table 2.7 also shows posterior distributions of the TKF92 parameters with and without (phylogenetic) structural information. Increasing the dataset from 8 to 12 sequences reduces the uncertainty associated with the parameter estimates in all cases, but a similar reduction in uncertainty in the alignment length and r is also observed when structural information is included. Alignments are typically slightly longer with the structural model, and the indel rate parameters, λ and μ, are estimated slightly higher. This shows the estimation of these parameters can also be affected by alignment uncertainty, hence the inclusion

		8-globins		12-globins	
		seq-only	seq+struc	seq-only	seq+struc
L	5%	167	177	174	184
	50%	**173**	**182**	**184**	**188**
	95%	183	186	194	194
	GR	*1.00*	*1.06*	*1.01*	*1.02*
r	5%	0.669	0.700	0.644	0.681
	50%	**0.787**	**0.796**	**0.742**	**0.761**
	95%	0.887	0.880	0.833	0.832
	GR	*1.00*	*1.04*	*1.00*	*1.02*
λ	5%	0.021	0.035	0.028	0.045
	50%	**0.049**	**0.071**	**0.050**	**0.073**
	95%	0.092	0.121	0.079	0.109
	GR	*1.00*	*1.00*	*1.00*	*1.00*
μ	5%	0.021	0.037	0.029	0.047
	50%	**0.053**	**0.077**	**0.053**	**0.080**
	95%	0.103	0.137	0.087	0.123
	GR	*1.00*	*1.00*	*1.00*	*1.00*

Table 2.7: Posterior quantiles for alignment lengths (L), and TKF92 indel model parameters for globin datasets, aggregated from four independent MCMC chains in each case. All runs used a burn-in of $10m$ iterations, followed by a sampling period of $20m$ (sequence-only) and $40m$ (sequence + phylogenetic structural drift), with samples for all parameters recorded every 200 iterations, hence $100,000$ samples were taken for the sequence-only runs, and $200,000$ for the structural variants. Gelman-Rubin potential scale reduction factors (GR) are shown in each column.

of structural information also has the potential to improve estimates of insertion and deletion rates by improving alignment accuracy.

2.5 Heterogeneity in structural diffusivity

The structural drift model also enables the estimation of separate structural diffusion rates for different branches. As mentioned earlier, we observe strong heterogeneity in the rates of structural drift within several families, suggesting that structure is subject to varying degrees of selective pressure. In particular, higher structural drift appears to be associated with changes in function, with certain patterns suggestive of evolution by duplication and divergence.

2.5.1 Heterogeneous structural evolution among the globins

On the 12-globin dataset, there are some striking examples of heterogeneity in the structural drift rates across the tree, consistent with the observations of Illergård *et al.* (2009). As shown in Figure 2.19, diffusivity is often higher along internal branches between proteins or clades that perform different functions (*see Section 2.5.2 for further discussion*). There is also strong evidence for purifying selection (low σ^2) in six out of the 12 structures, corresponding to the haemoglobins and myoglobins (*see Figure 2.20*), implying a high degree of selective pressure to preserve structure in these proteins.

Equally notable are the highly increased rates of structural drift among the plant globins, particularly along the internal branch between the type-I non-symbiotic globin (nsGb) 2oif and the symbiotic leghaemoglobins (Lhb) 1bin and 1lh1. Although it was first hypothesised that Lhbs may have evolved from a bacterial ancestor, it is now thought that the Lhbs evolved from the nsGbs around 200mya, acquiring O_2 transport capability through the stabilisation of the open pentacoordinate haem configuration as opposed to the original, more stable hexacoordinate configuration (Garrocho-Villegas *et al.*, 2007; Hoy *et al.*, 2007; Landsmann *et al.*, 1986; Vinogradov *et al.*, 2005).

Although our structurally-based results support this same topology, there is a noticeable acceleration in the rate of structural evolution between the nsGbs and Lhbs. Previous studies have also uncovered a high rate of sequence variation in Lhbs than type-I nsHbs during the evolution of land plants, suggesting that different types of evolutionary pressures may have been involved along these two separate lineages (Vázquez-Limón *et al.*, 2012). Since the purpose of O_2 transport functionality in Lhbs is to sustain the symbiotic bacteria living in the root nodules of leguminous plants, it is conceivable that this increased rate of structural divergence may be related to the emergence of symbiosis in legumes. Analysis of intermediate structures along this transition may help to uncover more of the mechanisms responsible for this major structural transition (Gopalasubramaniam *et al.*, 2008).

With the cysteine proteinases, the drift rates are generally much smaller, as might be

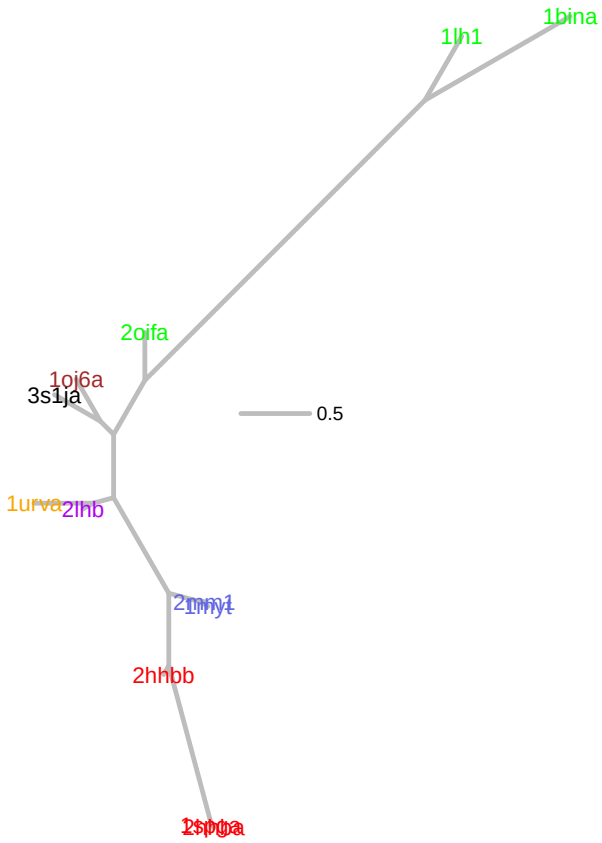


Figure 2.19: Consensus tree with branches scaled by local σ_k^2 parameters for the 12-globin dataset.

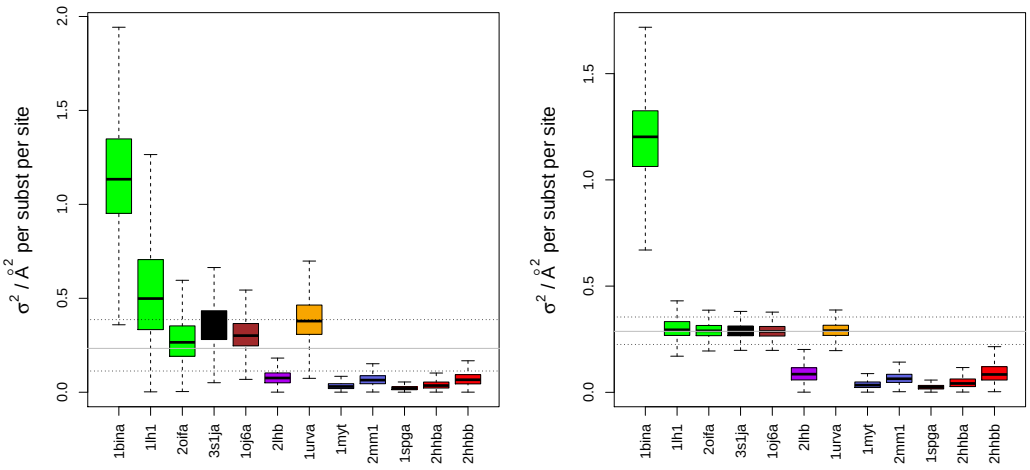


Figure 2.20: Distributions for σ_k^2 for leaf branches in the 12-globin dataset. Parameters shown as estimated with low ($\gamma = 0$, left) and high ($\gamma > 0$, right) shrinkage to the global σ_k^2, using the shrinkage mixture prior described in Section 2.2.8. In the high shrinkage case, the posterior median for γ was 0.50, with a 95% highest posterior density interval of $[0.41, 0.64]$, and effective sample size of 2390, indicating good mixing on the z_k parameters.

expected given that function is largely conserved across most members of the datasets, with $\widehat{\sigma_g^2} = 0.05$. However, several of the branches have much larger diffusivity, for example for the porcine cathepsin (PDB code 8pch) we have $\widehat{\sigma^2}_k = 0.43$ (Gunčar *et al.*, 1998) (*see Figure 2.21*).

2.5.2 Patterns of structural divergence

Further intriguing patterns of heterogeneity in the structural evolution rates can be seen on larger datasets. One of the largest sets we have examined so far with this methodology is a 28-structure protein kinase dataset, listed in Table 2.8. This dataset was constructed in order to contain several structural representatives from each of the clades in the sequence-only analysis of Manning *et al.* (2002). For this dataset, convergence required $60m$ iterations, after which the MCMC chains were run for a further $40m$ iterations, sampling every 200 iterations, yielding a total of $200,000$ samples.

Figure 2.22 shows the consensus tree for the protein kinase dataset, as well as the same

Structure	Protein
1gz8	Cell division protein kinase 2
2gfs	Mitogen-activated protein kinase 14
1q5k	Glycogen synthase kinase-3 beta
1o61	Aminotransferase
1uu3	3-phosphoinositide dependent protein kinase 1
1tki	Titin
1jks	Death-associated protein kinase
2oza	MAP kinase-activated protein kinase 2
1yhw	Serine/threonine-protein kinase PAK-1
2j4z	Serine/threonine-protein kinase 6
1qpc	LCK kinase
1mp8	Focal adhesion kinase 1
1t46	Tyrosine kinase
1p4o	Insulin-like growth factor I receptor
1r0p	Hepatocyte growth factor receptor
1t4h	Serine/threonine-protein kinase WNK1
1u46	Activated CDC42 kinase 1
1xbb	Tyrosine-protein kinase SYK
1xws	Serine/threonine-protein kinase PIM-1
2jav	Serine/threonine-protein kinase NEK-2
3blh	Cell division protein kinase 9
2jfl	STE20-like serine/threonine-protein kinase
1vjy	TGF-beta receptor type I
1fvr	Tyrosine protein kinase TIE-2
1nvr	Serine/threonine-protein kinase CHK-2
1uwh	B-RAF Serine/threonine-protein kinase
1xkk	Epidermal growth factor receptor
1s9j	Mitogen-activated protein kinase 1

Table 2.8: The human protein kinase dataset.

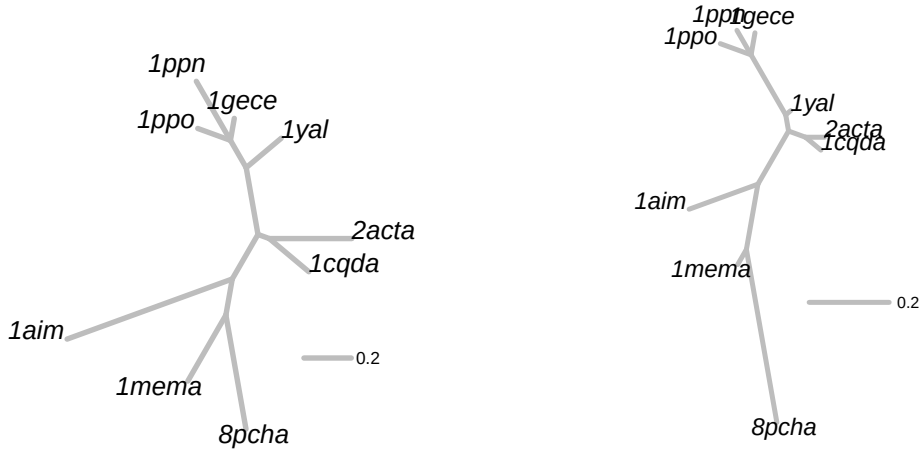


Figure 2.21: The consensus tree for the cysteine proteinase dataset, with branches scaled according to mean branch length (left), and mean σ_k^2 (right). The variability in lengths in the right-hand panel illustrates the heterogeneity in structural diffusivity coefficients across the tree.

tree after scaling the branch lengths according to the structural diffusivity. For the most part, the structures cluster according to the functional groupings discussed by Manning *et al.* (2002), as indicated by the colours for each taxon. Several clades contain branches with very low as well as very high drift rates. Overall, just over half of the branches end up with diffusivity parameters shrunk to the global σ_g^2, with the posterior median for γ at 0.56 (95% intervals spanning from 0.48 to 0.59, and an effective sample size of 845).

In general, we observe some recurring patterns of heterogeneity that can be divided into four categories of particular interest, as shown in Table 2.9. We also see patterns of the type $(\boldsymbol{A} + \boldsymbol{B})^n$, i.e. repeated bifurcations where one of the children of the pair has a very low diffusivity, and no descendants, for example in the top right of Figure 2.22, and between the α and β globins, as shown in Figure 2.19. This may be a signature of a series of duplication and neofunctionalisation events, whereby the ancestral protein retained its original function, and the new duplicate was either free, or perhaps under selective pressure to evolve a new functionality.

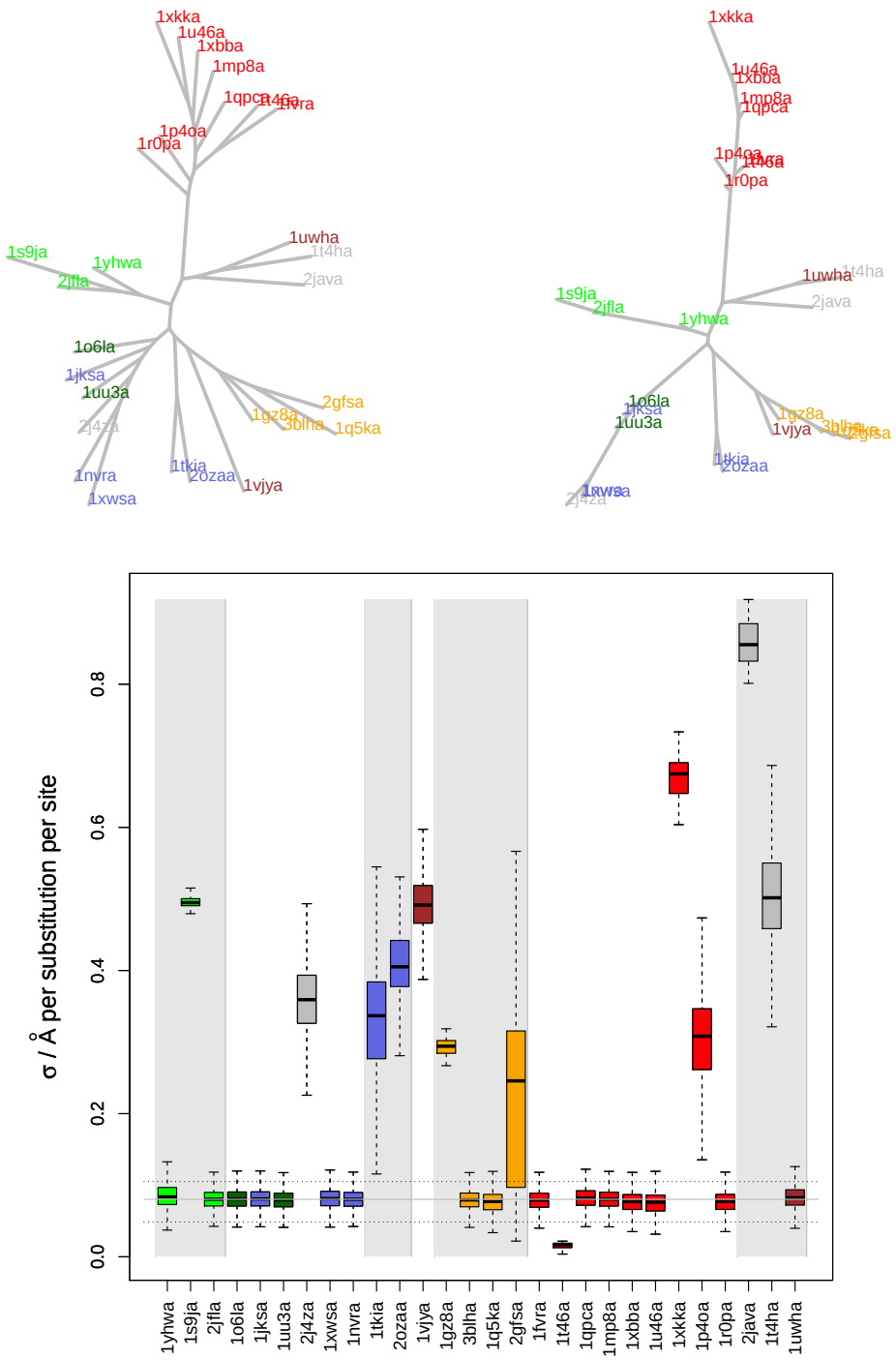


Figure 2.22: The consensus tree for the protein kinase set, with branches scaled according to mean branch length (top left), and mean σ_k (top right). Note that σ_k rather than σ_k^2 is used for plotting the tree for ease of visualisation. The distributions for diffusivity coefficients at the leaf branches are summarised below. Taxons are colour-coded according to the scheme in Manning *et al.* (2002): red = tyrosine kinases, blue = calmodulin-dependent kinases, light green = yeast sterile kinases, dark green = (PKA,PKC,PKG), orange = (CDK,MAPK,GSK3,CLK), brown = tyrosine kinase-like, grey = uncategorised. Grey boxes in the background indicate boundaries between clades based on the consensus tree. Median and highest posterior density interval for the global σ_g^2 is shown by the dotted lines running across the boxplot.

Type	*Magnitude of* σ_k^2	*Suggested explanation*
A	small	high structural constraint (e.g. Hb and Mb)
B	large	accelerated rate of structural drift (e.g. when developing new functionality, for example in symbiotic Lhb)
A+B	small + large	possible duplication event; the branch with the smaller diffusivity is closer to the ancestral structure, allowing the other structure to diverge since there is some redundancy (several examples, including α and β Hb); a form of *neofunctionalisation* (Hughes, 1994; Rastogi and Liberles, 2005).
A+A	small + small	strong selective pressure to preserve structure (e.g. α Hb in human versus fish); may be a form of *subfunctionalisation* (Rastogi and Liberles, 2005)

Table 2.9: Patterns of heterogeneity among branch-specific structural diffusivity parameters, as observed in several different datasets.

2.5.3 Structural determinants of evolutionary drift rates

There is theoretical and empirical evidence to suggest that more designable proteins (those with a higher contact density) may evolve faster on the sequence level, since destabilising mutations are more easily tolerated in such cases (Bloom *et al.*, 2006; England and Shakhnovich, 2003; Tiana *et al.*, 2004). Equivalently, in our framework these cases correspond to branches for which σ_k^2 is small, meaning that mutations to the sequence result in a smaller change to the structure along these branches.

On the other hand, Lukatsky *et al.* (2007) provided evidence to suggest that structurally similar proteins may exhibit a propensity to interact with each other; indeed, the globin family provides a particularly rich set of examples of oligomer formation, ranging from the familiar α-β Hb heterotetramer, to the large extracellular homo-oligomers found in insects (Lamy *et al.*, 1996; Terwilliger, 1992). Although this may present a mechanism for the evolution of new binding partners (Levy *et al.*, 2008), it also poses a risk of unintentional homodimerisation. The need to avoid homodimer formation may give rise to what has been termed *negative design*, whereby a structure accumulates mutations that reduce its potential for self interaction Lukatsky *et al.* (2007). Such negative design may explain local

accelerations in structural drift at certain branches in the tree, particularly after a duplication event, when the presence of two copies of a particular protein is likely to further increase the propensity for unwanted self-oligomerisation.

As discussed by Hughes (1994), one possible mechanism by which functional diversification can occur in enzyme families is to evolve new binding capabilities through modulating the charge distribution on the surface of the protein. Among the cysteine proteinases, Hughes (1994) observed several regions of major charge difference among cathepsin B sequences, and devised a statistical test that suggested shifts of charge in certain regions of the structure were likely to have arisen as a result of selective pressure to diversify. The elevated rates of structural drift we observe in certain regions of the tree may be a signature of a similar mechanism of structural diversification.

In our case, we also see an elevated number of charge differences between the human cathepsin K, (PDB code 1mema) and the other sequences, including its nearest neighbour, 8pch (*see Figure 2.23*). When combined with the observation of an unusually high structural drift rate, this might suggest that charge modulation could play a role in the functional diversification of the cysteine kinase family.

2.5.4 Independence of drift rates and branch lengths

Since the default substitution model we use here posits a single subtitution rate for the whole tree, local variations in substitution rate will be encoded as longer or shorter branch lengths, which may lead to non-clock-like trees. In such a case, even if structural evolution were clock-like, this could still lead to the estimation of heterogeneity in the diffusivity parameters, as an indirect reflection of heterogeneity in the substitution rates. In this case σ_k^2 would be approximately inversely correlated with t_k.

However, on all the datasets we examined there was essentially zero correlation between σ_k^2 and the branch length for all k, showing that these quantities contain separable sources of information (cf. Figure 2.24). In addition, the branch lengths estimated under

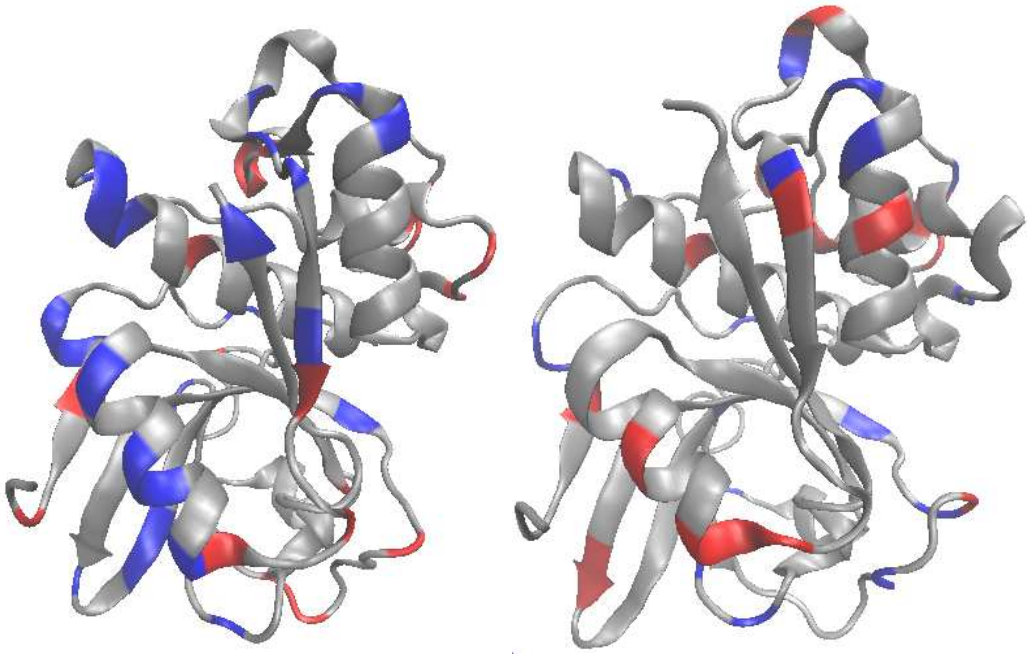


Figure 2.23: Illustration of charge differences between cysteine proteinase structures. Structures for 1mem (left) and 8pch (right), with charged residues highlighted in red (positive) and blue (negative), showing a large number of differences between the two proteins.

the sequence-only and sequence + structure model variants are very similar (Figure 2.24), suggesting that any effect of variation in substitution rate on branch lengths would still present in the joint sequence + structure model. These observations suggests that similar patterns of heterogeneity in diffusivity parameters would be seen using a substitution model that allows for branch-specific substitution rates. Further investigation with other types of substitution models will help to reveal to what extent these patterns are robust to model choice.

2.6 Discussion

The main achievement of the work presented in this Chapter is the development of a tractable probabilistic model for joint evolution of sequences and structures on a phylogenetic tree. Our results demonstrate that inclusion of structural information reduces posterior uncertainty over alignments and topologies, improves alignment accuracy and reduces

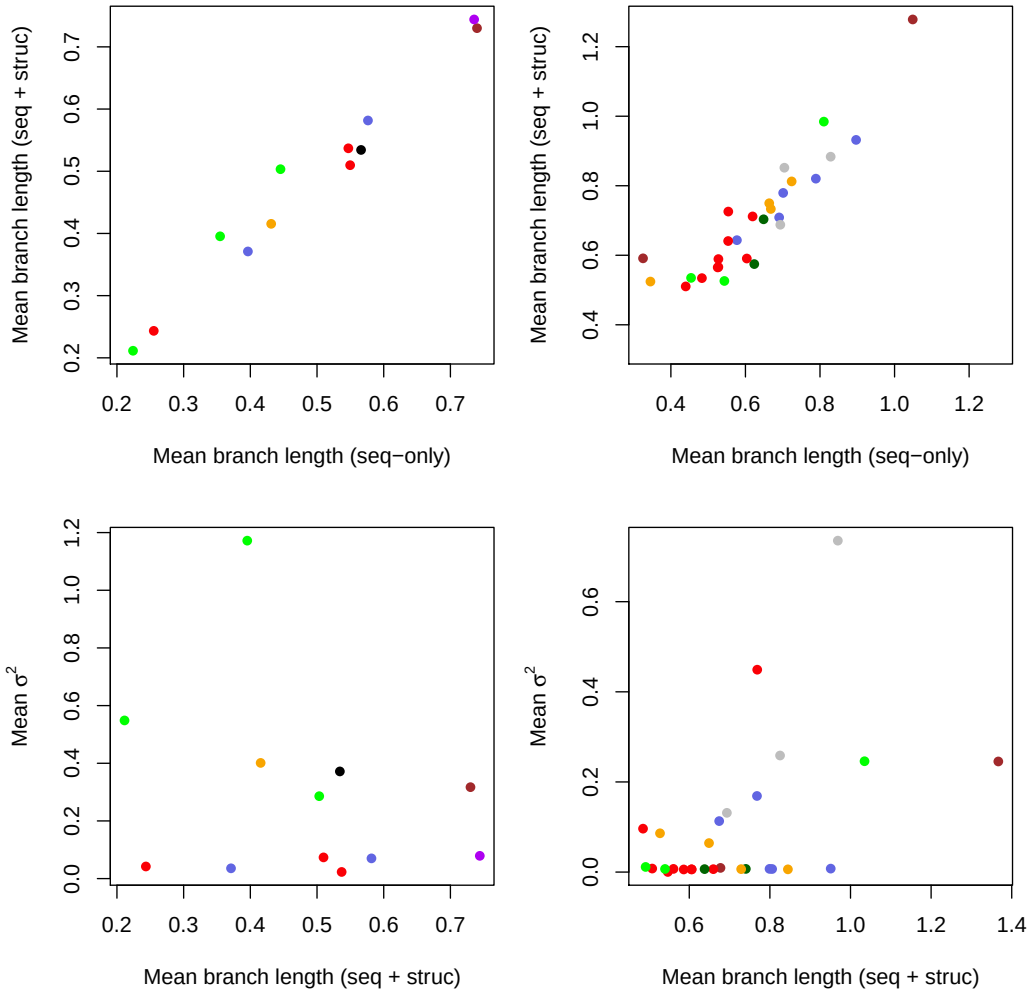


Figure 2.24: ***Top*: Comparison of consensus branch lengths for trees computed under the sequence-only, and phylogenetic structural drift models.** ***Bottom*: Branch-specific σ_k^2 parameters plotted against consensus branch length.** The lack of correlation shows that the drift rates do not depend on branch lengths in a predictable fashion. Data shown for the 12-globin dataset (left), and the rotein-kinase dataset (right), coloured according to the same schemes used in Figures 2.19 and 2.22.

the number of tree errors, allowing for more reliable inference over larger evolutionary distances. The structural model is also more robust to the particular dataset chosen for analysis, whereas sequence-only models can be highly sensitive to this choice.

Using this approach, we are able to provide structural insights into the evolutionary history of the globin family, whereas sequence-only methods encounter high uncertainty and sensitivity to choice of dataset, making it difficult to confidently characterise deep splits in the tree.

Structural information can reduce topology uncertainty both by reducing alignment uncertainty and by adding additionally information regarding divergence times for estimating topology and branch lengths. We observe that in some cases a large decrease in topology uncertainty can be obtained even with a non-phylogenetic structural model (the ϵ-only model), which affects the tree only via the alignment. This suggests that alignment inaccuracy and/or uncertainty can be a major cause of topology uncertainty, and further highlights the benefits of approaching alignment and topology inference in a joint framework, as we have done here.

2.6.1 Future work

As discussed, several modelling assumptions are made to ensure tractability of likelihood computations. These are likely to be reasonable for modelling local fluctuations around a particular fold, but may be less appropriate for modelling larger deviations. In particular, the assumption of independence between sites under the structural model becomes questionable when considering large displacements of secondary structure or other structural motifs. We are currently exploring extensions to allow for dependency between sites, although this is computationally very demanding, just as it is for sequence-based models.

The current model requires experimental structural data for all sequences included in the analysis. This is somewhat restrictive, and we are also developing extensions to allow analyses when only a subset of the sequences have structural data available. A number of

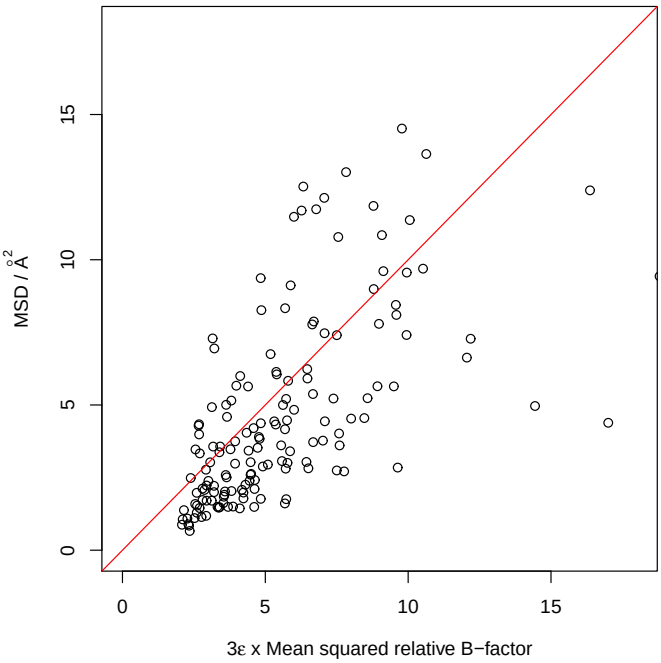


Figure 2.25: Average pairwise mean squared deviation (MSD) for each column plotted against $3\epsilon_i$. [cf. equation (2.15) in the main text] Shown for the maximum likelihood MCMC sample for the 12-globin set under the drift model, showing that for most columns the *B*-factor-derived information is a good predictor of the MSD (variance), which supports the use of *B*-factors as a measure of baseline variability. The multiplication by 3 is necessary because MSD contains a contribution from *x*,*y* and *z*. The surplus variability beyond the baseline is modelled by the diffusion component of the drift model.

other extensions to the model could be considered, including using mixture models in the diffusion process to increase flexibility of the model and potentially locate differing rates of evolution along the sequences, for example to identify structural features that are under strong selection.

Another modification that may improve model fit would be to allow the priors for each σ_k^2 to depend on the rate of the parent branch, as discussed by Thorne *et al.* (1998) and Aris-Brosou and Yang (2002), to account for the fact that evolutionary rates are likely to diverge as a function of time. From a biophysical perspective, this may reflect the fact that the σ^2 parameters are related to the ability of a structure to accommodate sequence mutations, and this property is likely to be inherited to some extent from the parent structure.

Currently the model uses the magnitude of the crystallographic *B*-factor to estimate

the expected standard deviation for each atom. In the cases we have examined, this relationship appears to hold very well (*see, for example, Figure 2.25*), but there may be cases where anisotropy and the presence of multiple conformers could lead to noticeable deviations from the expected behaviour (DePristo *et al.*, 2004). By instead using the *B*-factor information to specify a prior distribution for each ϵ_{ki}, it would be possible to allow the data to override the *B*-factors where appropriate, although a larger number of structures may be needed to carry out parameter estimation in such a model.

Finally, as mentioned earlier, the structural model presented here is independent of the particular choice of indel model. By combining structural drift with other stochastic models of insertion and deletion, for example the recently developed Poisson indel model (Bouchard-Côté and Jordan, 2013), which allows for analytical marginalisation of indel histories as a result of some simplifying model assumptions, it may be possible to increase the size of datasets that can be analysed using this type of joint approach.

Chapter 3

Improved MCMC techniques for joint sampling of alignments and trees

The Java software package StatAlign was originally developed as a tool for carrying out joint Bayesian estimation of phylogenies and alignments (Novák *et al.*, 2008). The software uses a Markov chain Monte Carlo (MCMC) scheme for sampling from the joint posterior distribution under the probabilistic model of substitution, insertion and deletion introduced by Thorne *et al.* (1992), using techniques such as those discussed by Lunter *et al.* (2005b). In this chapter, we discuss a number of improvements and new features that have been added to the software in order to extend its range of functionality.

3.1 New proposals for continuous parameters

Previously StatAlign made use solely of truncated uniform random walk moves on edge lengths and TKF92 indel model parameters. With the inclusion of additional layers into the model, the need for more efficient proposals becomes more pressing, and this moti-

vated the inclusion of several new types of proposal distribution for univariate continuous parameters.

3.1.1 Multiplicative proposals

As discussed by Lartillot (2006), improved mixing may be obtained by using multiplicative proposals, whereby the parameter, θ, is multiplied by a log-normal random variable. The log-normal multiplicative move can be considered either as an asymmetric move on the parameter itself, requiring an adjustment to the proposal probabilities in the Hastings ratio (equal to θ'/θ), or alternatively as a symmetric move on the logarithm. In the latter view, the Hastings ratio is unity, but a Jacobian is required to account for the transformation of variables from θ to $\log\theta$, resulting in an equivalent effect on the Metropolis-Hastings ratio.

One advantage of the change-of-variables construction is that it allows for more complicated proposals that may better match the expected shape of the posterior distribution, for which the proposal distribution may not have a simple analytical form. For example, for a parameter that lies within a restricted domain such as $[0, 1]$, one possible option is to consider a random walk (Gaussian or otherwise) on $\text{logit}\,\theta = \log\theta/\log(1-\theta)$. In this case, the Jacobian for the reverse transformation is given by $d(\text{logit}\,\theta)/d\theta = \theta(1-\theta)$. Due to the nature of the logistic function, with this move a particular jump size would have a larger effect for parameter values close to 0.5, but may be better able to explore the tails of a distribution whose domain is restricted to $[0, 1]$.

In addition to utilising multiplicative proposals for individual edge lengths, another move has been introduced by which all the edges are simultaneously multiplied by a log-normally distributed variable. The motivation behind this is to address the high correlation between neighbouring edge lengths, which can otherwise lead to poor mixing.

As part of the new StatAlign MCMC code framework, a modular structure has been introduced that allows proposal distributions to be easily constructed in a hierarchical fashion, greatly simplifying the exploration of new move types.

3.1.2 Automatic tuning of proposal variances

For all univariate continuous parameters, the proposals used involve a random walk, either on the parameter itself, or a transformation thereof (*see Section 3.2.1*), and the width of the proposal distribution will affect the acceptance rate of the move. Previously all proposal widths were set to hard-coded, fixed values in StatAlign, but this becomes increasingly inefficient as the number of parameters increases, and is unlikely to be optimal in a variety of different datasets.

In order to address this, an automatic proposal tuning scheme was introduced, whereby during the burn-in, at intervals of a certain specified length, the acceptance rate of each MCMC move is queried. If it does not lie within a certain specified range (which is set to $[0.2, 0.4]$ by default for univariate continuous parameters, as per the considerations outlined by Roberts *et al.* (1997) and Roberts and Rosenthal (1998), but can be modified on a per-move basis), then the proposal width control variable for that particular move is multiplied or divided by a fixed factor (set to 0.7 by default), depending on whether the acceptance rate is too high or too low. When the proposal width is modified, the acceptance counts for the move are reset to zero, and the process begins again. Provided that the parameters are close to convergence by the time the proposal tuning process terminates, this process is typically very successful, and almost always achieves the desired acceptance rates for all continuous parameter moves.

3.2 Joint moves on indel parameters

Since λ and μ are highly correlated, it is very difficult to traverse the joint posterior using only independent proposals to each parameter, even when combined using the scheme above (*see Figure 3.1*).

3.2.1 Reparameterisation

In order to improve mixing on (λ, μ), we first note that the high correlation between these two parameters is due to the fact that their ratio is related to the expected sequence length, which itself has a particular distribution, and may be highly constrained. As such, we consider a transformation of variables from (λ, μ) to (ρ, θ), where $\rho = \lambda + \mu \in \mathbb{R}^+$ and $\theta = \lambda/\rho \in (0, 0.5)$. Given this new parameterisation, we introduce moves that alter ρ keeping θ fixed, and vice versa, which have the following effects on λ and μ

$$(\rho := \rho^* \mid \theta) \Rightarrow (\lambda^* := \rho^* \theta\,,\ \ \mu^* := \rho^* - \lambda^*) \tag{3.1}$$

$$(\theta := \theta^* \mid \rho) \Rightarrow (\lambda^* := \rho\, \theta^*\,,\ \ \mu^* := \rho\ - \lambda^*) \tag{3.2}$$

3.2.2 Transformed priors

Having made the transformations above, we can consider the resulting priors on the transformed variables. With $\lambda \sim \mathsf{Gamma}(a_\lambda, b_\lambda)$ and $\mu \sim \mathsf{Gamma}(a_\mu, b_\mu)$, and assuming $a_\lambda = a_\mu$ and $b_\lambda = b_\mu$, this leads to the following implied priors on the reparameterised combinations

$$\rho \sim \mathsf{Gamma}(2a_\lambda, b_\lambda) \tag{3.3}$$

$$\theta \sim \mathsf{Beta}(a_\lambda, a_\lambda) \tag{3.4}$$

using standard results for convolutions of Gamma variates.

3.2.3 Illustration

As shown in Figure 3.1, switching from independent proposals for λ and μ to orthogonal moves on (ρ, θ) results in a significant improvement in mixing on these parameters, allowing for the posteriors to be more reliably inferred.

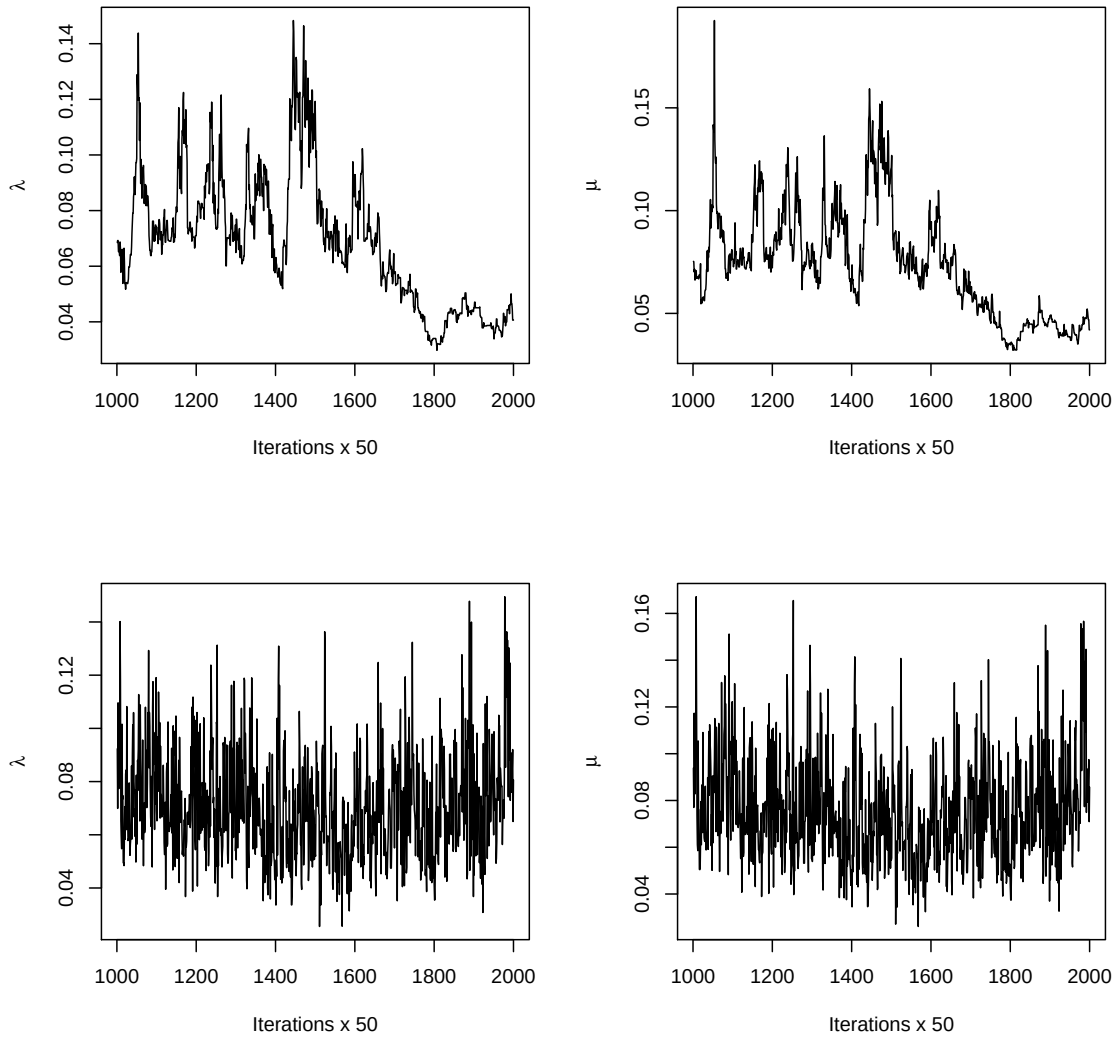


Figure 3.1: Improved mixing for the λ and μ parameters of the TKF92 indel model after switching to the alternatively parameterised moves described in Section 3.2.1. (Example generated using a globin dataset.)

After experimenting with different combinations, we opted to combine Gaussian random walk moves on ρ and θ with a pure-λ move in the ratio $3 : 3 : 2$, which yields significant improvements in mixing and effective sample size over the original scheme.

3.3 Interdependence of topology and alignment sampling

Although for certain indel models it is possible to analytically marginalise over indel histories on internal branches of the tree (Lunter *et al.*, 2003a,b), for the more realistic TKF92 model no such algorithms have been found (Lunter *et al.*, 2005a). Hence, in order to compute likelihoods under the joint model, StatAlign adopts a data augmentation approach, whereby the multiple alignment between the sequences at the leaves of the tree is represented in terms of a set of pairwise alignments along each branch. The full data likelihood for the augmented model can be easily computed as a product over branches, as described in Section 2.1.

Although alignments can be changed independently of the other parameters, the data augmentation setup means that MCMC moves which change topologies typically also require changes to the alignment, since the internal node structure may change as a result of such moves, which may invalidate certain columns of the alignment. The reasoning behind this is most easily illustrated by example.

3.3.1 Nearest-neighbour interchange moves may invalidate alignments

The main type of topology move we consider involves taking a particular vertex in the tree, and swapping it with its *uncle*, as outlined in Figure 3.2. This procedure is termed a *nearest-neighbour interchange* (NNI). Although more complex topology schemes can be useful in some circumstances (Lakner *et al.*, 2008), we have focused on the use of NNI moves in StatAlign, since they can generally be implemented much more efficiently in conjunction with changes to the alignment, and still result in ergodic sampling (Drummond *et al.*, 2002).

As shown in Figure 3.3, an NNI move may lead to invalidation of columns, since the definition of homology requires that a character cannot be inserted more than once in a particular column. As such, alignment and topology sampling are two closely interlinked problems, and must be considered in tandem. We will first consider pure alignment moves,

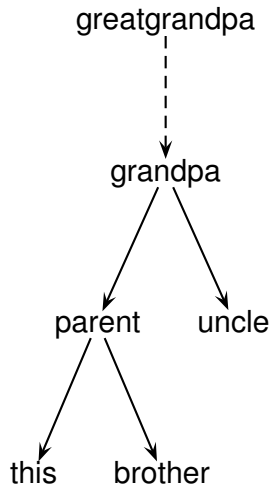


Figure 3.2: Tree showing the relationships between the nodes involved in the nearest-neighbour interchange (NNI) move. The NNI procedure detaches this from parent, and reattaches it as the right child of grandpa, and then sets uncle to be the left child of parent. The dotted line represents a branch that may not be present, if grandpa is the root of the tree.

and then discuss modified topology sampling schemes that include concomitant changes to the alignment.

3.4 Alignment sampling

Although dynamic programming to resample the full multiple alignment is infeasible, pairwise and three-way alignments can be sampled directly using standard HMM algorithms (Lunter *et al.*, 2005b). The proposal probability of such a move can be computed exactly as a by-product of the sampling procedure, and can then be used as a correction factor in the Hastings ratio. In order to improve the probability of acceptance of such moves, StatAlign makes use of alignment moves that cut out smaller windows of the overall alignment, along with a subset of sequences corresponding to a subtree of the overall tree, progressively realigning along the branches in the subtree.

For each internal node in the tree, partial likelihood vectors are stored for each position in the corresponding sequence. This allows for rapid evaluation of the likelihood after making local modifications to the alignment. One of the key moves used to update the alignment

BEFORE:	1234567	AFTER:	1234567
A (this)	o-o----	B (this)	oo-ooo-
brother	o-o----	brother	o-o----
B (uncle)	oo-ooo-	A (uncle)	o-o----
parent	XXXX--X	parent	X*X****
grandpa	XX-XXXX	grandpa	XXX**X*
greatgrandpa	oo---o-	greatgrandpa	oo---o-

Column	*Initial state*	*Status*	*Allowed states after move*
1	XX	valid	{XX}
2	XX	valid	{XX}
3	X-	invalid	{XX}
4	XX	valid	{XX,X-,--}
5	-X	invalid	{XX,X-,--}
6	-X	invalid	{XX}
7	XX	valid	can be eliminated (cf. §3.5.6)

Figure 3.3: Illustration of some possible ways in which columns can become invalidated after an NNI move. This invalidation is due to the requirement that insertions only occur once at any particular site. In the above tables, node *A* is swapped with its uncle, node *B*. Positions marked by a $*$ denote characters that are allowed to be present or absent after the nephew-uncle swap, and X denotes a character in the parent or grandpa. Characters at the other nodes are kept fixed, and denoted by o. Internal indel states are shown in the table as pairs, where X- indicates a character at the parent, and a gap at the grandpa, for example.

is a parent-child pairwise resampling move, which resamples the alignment along a particular branch. Algorithm 2 illustrates how the cached partial likelihoods (corresponding to the **fels** and **upp** vectors) are used within the context of this algorithm.

Algorithm 2 The dynamic programming algorithm used to sample a pairwise alignment from the TKF92 pair-HMM, including emission (substitution) probabilities.

```
1:  function HMM2ALIGN(Vertex child, Vertex parent)
2:  states = {(XX), (X-), (-X), (--)}
3:  Q = rate matrix describing substitution model
4:  T = TKF92 transition matrix
5:
6:  for i = 1,..., length(child) do
7:    for j = 1,..., length(parent) do
8:      for currState in states do
9:        for prevState in states do
10:         if currState places character at child then
11:           previ = i-1
12:           fels = FELSEN(child,i)
13:         else
14:           previ = i
15:           fels = 1^T
16:         if currState places character at parent then
17:           prevj = j-1
18:           t = time from parent to child
19:           upp = UPPER(parent,j) * (exp tQ)
20:         else
21:           prevj = j
22:           upp = equilibrium distribution
23:         trans = DP[previ][prevj][prevState]
24:                                 * T[prevState][currState]
25:         emissionProb = upp * fels
26:         DP[i][j][currState] += trans * emissionProb
27: Stochastic backtrack through DP
28: end function
```

In order to yield an algorithm capable of sampling from the full posterior of the TKF92 model for a particular pair of vertices, we further modified Algorithm 2 such that line 19 includes the partial likelihood of all upper vertices of the tree (as shown), rather than an equilibrium assumption at the parent (as was previously the case). To compute the required quantities, we can adopt a belief-propagation type approach in order to store partial

likelihoods for different regions of the tree.

Whereas the Felsenstein recursion (Algorithm 3) computes the partial likelihood for all vertices lying below a particular vertex, we also require a similar algorithm to compute partial likelihoods for all other vertices of the tree, propagated up to the root, and then back down to the current node. This involves the definition of an UPPER function, as shown in Algorithm 4, defined in such a way that the relevant quantities can easily be stored at the internal nodes of the tree for efficiency. This allows the emission probability portion to be efficiently computed on the fly during the dynamic programming recursions for pairwise alignment, as shown at line 25 of Algorithm 2.

Algorithm 3 Felsenstein recursion for computing partial likelihoods for all vertices lying below a specified vertex on a tree.

```
function FELSENVertex v, site i
t_l = length of branch to left child
t_r = length of branch to right child
Q = rate matrix describing substitution model

if left child of v is a leaf of the tree then
   k = observed character
   left = column of (exp t_l Q) corresponding to k
else
   left = (exp t_l Q) * FELSEN(left child, site aligned to i)
if right child of v is a leaf of the tree then
   k = observed character
   right = column of (exp t_r Q) corresponding to k
else
   right = (exp t_r Q) * FELSEN(right child, site aligned to i)

return elementWiseProduct(left, right)
end function
```

3.4.1 Three-way alignment sampling

Although it results in an exact sample from the posterior for each pairwise alignment, the HMM2ALIGN algorithm keeps the lengths of the parent and child sequences fixed, and so does not allow for a full exploration of the alignment space. In order to allow for changes to the

Algorithm 4 Algorithm for computing the upper emission probabilities on a tree.

function UPPER(Vertex `v`, site`i`)
`brother` = other child of `parent(v)`
t_p = length of branch from `parent` to `v`
t_b = length of branch from `parent` to `brother`
Q = rate matrix describing substitution model

if `parent` contains a character at site j, aligned with i **then**
  upp = UPPER(`parent`,j) * (exp $t_p Q$)
else
  upp = equilibrium distribution
if `brother` contains a character at site k, aligned with i **then**
  fels = (exp $t_b Q$) * FELSEN(`brother`,k)
else
  fels = $\mathbf{1}^T$

return elementWiseProduct(**fels**^{T}, **upp**)
end function

lengths of the sequences at the internal nodes, StatAlign also makes use of alignment proposals that operate on triplets of vertices (a parent and its two children), during which the length of the parent sequence is resampled. To realign three nodes, it is necessary to consider a product-HMM, containing seven non-silent states (Miklós *et al.*, 2008). However, for the purposes of proposing new alignments, in the original StatAlign implementation the full 3-way HMM was replaced by a simplified version of the product with just two parameters, termed *hmm3* (*see Figure 3.4*). This yields a much more efficient algorithm for 3-way alignment (HMM3ALIGN – details omitted here), at the expense of an increased mismatch between proposal and target distributions. The values of the *hmm3* parameters were originally optimised for efficient mixing on DNA alignments. In StatAlign 3 these parameters are instead sampled from a discrete set of possible values, such that the proposals are now effectively taken from a mixture distribution, allowing for a wider range of alignments to be explored.

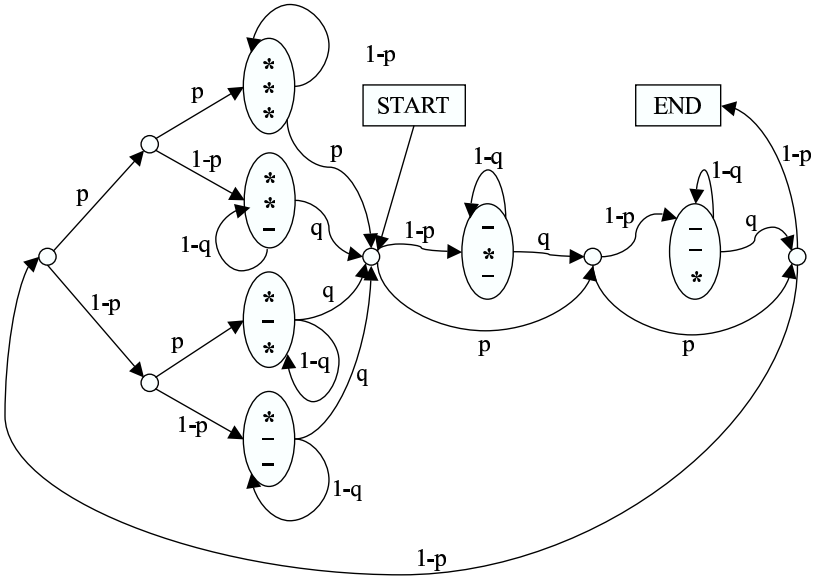


Figure 3.4: The multiple-HMM *hmm3* used to propose realignments pairs of sequences to an unknown parent sequence. Previous versions of StatAlign used a single fixed value for p and q; in StatAlign 3 these parameters are sampled from a discrete set of possible values, such that the proposals are now effectively taken from a mixture distribution, allowing for a wider range of alignments to be explored. (Reproduced from Miklós *et al.* (2008).)

3.5 Improvements to topology sampling

We faced two major challenges in sampling topologies. Firstly, the previous version of StatAlign suffered from slow mixing over topologies when the tree contained long branch lengths; secondly, the addition of model extensions (such as the StructAlign plugin) results in more concentrated modes in the likelihood surface, such that it may be more difficult to accept changes to the topology. Hence, significant time was spent developing improvements to the MCMC moves responsible for switching between topologies.

3.5.1 Original StatAlign topology+alignment move

The original NNI move in StatAlign operates by first proposing a new topology, and then realigning all the nodes involved in the neighbourhood, according to the scheme presented in Algorithm 5.

Algorithm 5 Nearest-neighbour interchange and subsequent realignments.

```
function swapWithUncle(Vertex nephew)
Set nephew as child of grandpa
Set uncle as child of parent
hmm3Align(uncle, parent, brother)
hmm3Align(parent, grandpa, nephew)
if greatgrandpa exists then
  hmm2Align(grandpa, greatgrandpa)

return log proposal ratio
end function
```

It should be noted that hmm3Align results in the parent and grandpa sequences being resampled according to the chosen path through the HMM. The existing implementation of this move also allows for completely homologous columns to be left as *anchors*, restricting the realignment steps to the regions between the anchors (Lunter *et al.*, 2005b).

However, while moves of this type usually result in an improvement in the indel portion of the log likelihood, and occasionally in the substitution log likelihood[1], the proposal ratio is often highly negative, such that the acceptance rate for these moves can end up very low, especially when model extensions are included. Part of the reason for this is that there is a mismatch between *hmm3* and the true likelihood, such that the proposed alignments are much more likely under the proposal density than the true model, resulting in a very small back proposal probability relative to the improvement in the likelihood.

3.5.2 Simultaneous changes to topology and branch lengths

The most trivial extension to the NNI move is to allow branch lengths to change at the same time as the topology, since the pre-existing branch lengths may be highly inappropriate under the new tree (Lakner *et al.*, 2008). This was implemented by including a simultaneous multiplicative or uniform branch length proposal for the nephew, parent and uncle branches.

[1]The substitution log likelihood is termed the `orphanLogLike`, since this term is computed recursively for each *orphan* character (with no parents), whether an insertion, or a character present at the root.

3.5.3 LOCAL topology move

As a further attempt to improve mixing, the LOCAL topology move of Larget and Simon (Holder *et al.*, 2005; Larget and Simon, 1999) was implemented. This move operates by shifting the location of the parent node along the path connecting the nephew and the uncle, and if the shift results in it moving beyond the grandpa, an NNI move is carried out. The advantage of this procedure is that topology changes are typically only proposed when the edge lengths are in a configuration that is more likely to favour such a move (for example with a very short branch connecting two subtrees).

3.5.4 Fixed-column topology moves

Although these modifications result in improvements in acceptance rates for NNI interchange moves, in many cases the alignment resampling step still causes overall acceptance rates to be low. This motivated the search for an NNI move that would change the alignment as little as possible, minimising the effect of the proposal density on the Metropolis-Hastings ratio, while still achieving proposals that are reasonable under the indel model.

In order to do this, we adopt an approach whereby the indel state (`-` / `X`, corresponding to gap/character) associated with `this`, `brother`, `uncle` and `greatgrandpa` is kept fixed in each column before and after the NNI move. Conditional on this *neighbour configuration*, the indel state of the `parent` and `grandpa` rows is then resampled for a subset of the columns, in order to yield a valid alignment.

There are numerous ways of carrying out this procedure. Redelings and Suchard (2005b) describe an algorithm in which the indel state of `parent` and `grandpa` (which we will term the *internal indel state*) is sampled according to its full conditional, after carrying out dynamic programming to sum over all possible combinations of internal indel states. However, this requires working with a product HMM containing several hundred states, and presents numerous technical and theoretical challenges. The main complication arises due to the fact if we select an internal indel state that results in there being no char-

acters on one of the five branches involved in the NNI move, then it is necessary to keep track of all possible predecessor combinations that could have led to this situation in order to compute the TKF92 probability corresponding to a particular imputation of `parent` and `grandpa` characters.

In order to circumvent these issues, we instead consider independence proposals for the internal indel states. For any column in which there is only one possible choice of internal indel state after the NNI move, there is no need to consider alternative internal indel states. For columns where there are multiple possible choices (*see Figure 3.3*), we sample (parent, grandpa) from the following distribution

$$p(\texttt{--}) = 0.70$$

$$p(\texttt{-X}) = 0.12$$

$$p(\texttt{X-}) = 0.12$$

$$p(\texttt{XX}) = 0.06$$

after excluding the invalid options and renormalising. To further improve the chances of proposing a favourable imputation, we reweight this distribution so as to favour keeping the current internal indel state where possible. This is done by multiplying the corresponding probability by a constant factor (set to 5 by default), and then dividing all the probabilities by the new total.

As shown in Figure 3.5, this move results in acceptable changes to the orphan and proposal probabilities, but the indel probability still often decreases quite significantly, such that the acceptance rate is very small.

3.5.5 Block imputation

Part of the reason for the low acceptance rate with the independent-sites imputation procedure described above is that it can result in isolated insertions or deletions with a non-

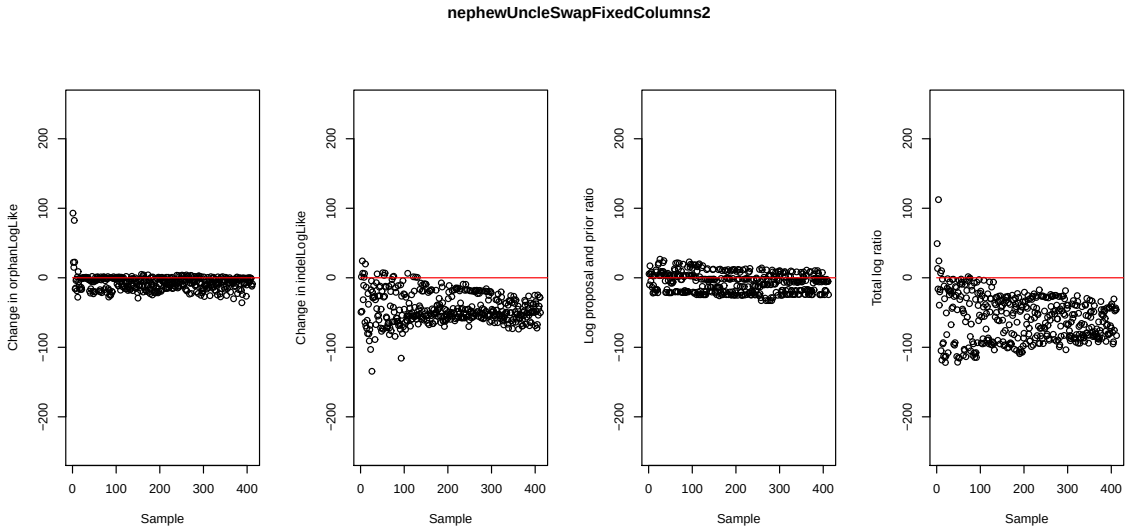


Figure 3.5: Breakdown of the different contributions to the log Metropolis-Hastings ratios for a set of topology proposals, using the fixed-column NNI move (without block imputation). Results shown for the 5-globin dataset of Chapter 2. The contributions from left to right are: `orphanLogLike` (log substitution likelihood), `indelLogLike` (log TKF92 probability), `Log proposal and prior ratio` (logarithm of ratio of new versus old prior density, multiplied by ratio of back proposal probability divided by forward probability) and the total (`Total log ratio`).

negligible probability. In order to address this issue, the algorithm was modified to propose new internal indel states in blocks. For each contiguous run of columns with the same neighbour configuration, we force the imputed internal indel state to be the same, favouring longer indels, which is more favourable under the TKF92 model.

As shown in Figure 3.6, this scheme results in much more favourable indel probabilities, and the proposal ratios are also of much smaller magnitude, since the proposal probability of such a move only requires the inclusion of one term for each block, rather than a term for each site.

Through the use of the block-imputation fixed-column NNI move, typical acceptance rates for topology moves are now in the region of 3-5% on sequence datasets, and 0.5-1.0% with structural models when using the StructAlign plugin, which is a large improvement over the previous case, and results in a scheme that allows for more reliable phylogenetic inference.

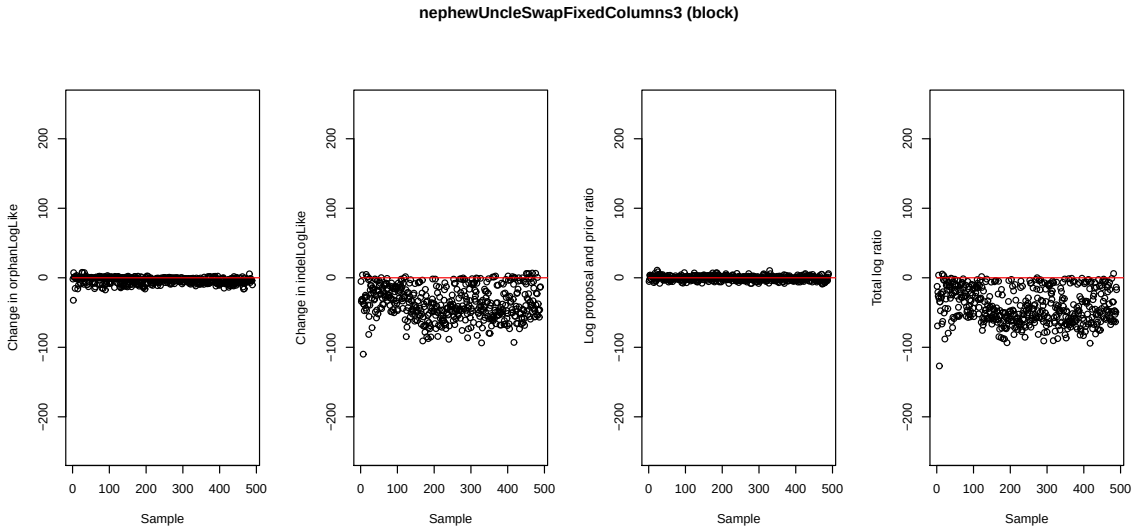


Figure 3.6: Breakdown of the different contributions to the log Metropolis-Hastings ratios for a set of topology proposals, using the block-imputation version of the fixed-column NNI move. Results shown for the 5-globin dataset of Chapter 2. The contributions from left to right are: `orphanLogLike` (log substitution likelihood), `indelLogLike` (log TKF92 probability), `Log proposal and prior ratio` (logarithm of ratio of new versus old prior density, multiplied by ratio of back proposal probability divided by forward probability) and the total (`Total log ratio`).

3.5.6 Persistent silent indels

Another issue that affects topology mixing in certain datasets is the accumulation of *silent indels*, which are stretches of characters at internal nodes that have no descendants. Such regions can arise when a sequence is realigned to its parent by HMM2ALIGN, after which some of the parent characters no longer have any descendants (*see Figure 3.7 for an example*).

```
BEFORE:                          AFTER:
this    XX----XX---              this    XX----XX---
brother XX----XXXX-              brother XX----XXXX-
parent  XX----XXXX-              parent  XX----XXXX-
uncle   X--XXXXX-XX              uncle   XXX---XX-XX
grandpa XXXXXXXXXXX              grandpa XXXXXXXXXXX
```

Figure 3.7: In the above example, the uncle is realigned to the grandpa, leaving behind three silent characters in the grandpa sequence. Removing these characters can be very difficult using the set of alignment MCMC moves previously implemented in StatAlign.

Due to the nature of the indel model, such regions will have a contribution to the overall posterior probability of an alignment, and their presence will alter the posterior for the indel parameters, λ, μ and r. Moreover, the contribution will depend on the length of the branch to which these regions are attached, meaning that they may affect the posterior on topologies as well.

We would expect such regions to be sampled with a very low probability, since they are highly unfavourable, and represent an extremely unparsimonious explanation of the data at the tips of the tree. However, as we have seen above, such regions can be easily created as the result of an HMM2ALIGN move, such that they then need to be removed somehow. The only move available in the previous version of StatAlign that is capable of removing such regions is HMM3ALIGN, in the case where the parent is the vertex containing the silent indel. However, under the *hmm3* proposal distribution, such regions are exceedingly unlikely, since they occur with probability proportional to $(1 - p)^2$. As such, any time we propose to remove such a region with HMM3ALIGN, the back proposal probability will be very small, and this will outweigh any gains in the likelihood, causing the move to be rejected. Hence, once a large silent indel appears, it will often end up persisting for a very long time in the alignment, potentially skewing the posterior distribution of the other parameters of interest, and occasionally causing proposed topology changes to be rejected.

To remedy this, we remove such regions deterministically when they arise during the burn-in, and stochastically extend or shrink silent indels thereafter.

3.6 Model extension framework

In order to facilitate the implementation of new coestimation approaches such as detailed in Chapter 2, it is highly desirable to have a software framework in which additional layers can easily be added on to the basic evolutionary model. In addition, it is also of great interest to explore ways in which other types of data besides sequences can be used to

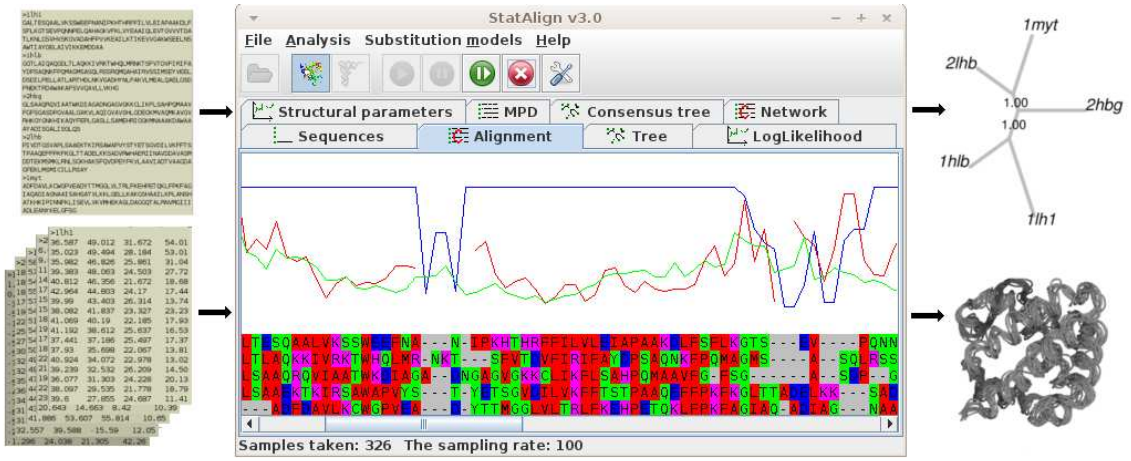


Figure 3.8: StatAlign 3 running in GUI mode, with the structural model extension enabled. Shown in the centre panel is the current alignment sample, annotated with marginal column probabilities (blue), average pairwise RMSD (red), and crystallographic *B*-factors (green). The input (left) consists of sequences and 3D coordinates; various outputs are generated, including consensus trees and structural superpositions.

assist with inference of phylogenies and alignments, since this may increase the robustness of the conclusions to model assumptions (Kumar *et al.*, 2012).

To accomplish both of the aforementioned aims, a generic model extension framework has been developed for StatAlign, whereby plugins can provide an additional contribution to the joint likelihood, based upon specified distributions for parameters of interest. Rather than simply allowing post-processing of alignment samples, this framework allows for model extensions to compute likelihoods at each step in the MCMC, thereby enabling information in these additional layers to be used to inform the estimation of other parameters.

For plugins that compute a column-wise contribution to the likelihood (as is the case for the structural alignment plugin used to implement the model in Chapter 2), this contribution can be used to improve the alignment proposals, by multiplying the emission probability on line 25 of Algorithm 2 by the likelihood arising from the model extension. In order to do this in an efficient fashion, each site at every node has assigned to it a vector containing the indices of the leaf nodes aligned above and below, such that for any particular

parent-child combination the implied alignment column can be obtained without additional computations.

3.7 Combination moves

Another new feature that has been added to StatAlign is the possibility to combine multiple MCMC move objects together into a joint move, in which each component is proposed separately but accepted as a single unit. The motivation behind this is to allow for improved mixing in the presence of high correlation between the parameters within the group. Although a more efficient solution may be to utilise a correlated proposal for the group, it is often difficult to design such joint moves, and the composition of independent proposals works well in many cases.

3.7.1 Combination moves for model extensions

One use for these types of combination moves is to allow for model extension parameters to be more efficiently sampled in conjunction with the core model parameters. For example, within the StructAlign plugin there are moves that resample the rotational transformations used to superpose the structures upon one another. One of these moves proposes a rotation to an entire subtree and simultaneously realigns the root of the subtree to its parent using *hmm2*, as described in Section 2.3.1. Without combining the rotation and alignment proposal, the subtree rotation is likely to be rejected due to movement of amino acids that are currently aligned to proteins in the rest of the tree. The combination move successfully alleviates this problem, and results in much faster convergence of the alignment and rotations/translations.

3.8 Future improvements

The developments detailed in this Chapter have made it possible to reliably carry out joint inference of alignments and trees under the joint sequence-structure model described in Chapter 2. Currently the software is capable of reliably handling up to 15 to 20 taxons, with larger datasets feasible for cases where uncertainty is lower. However, the amount of runtime needed to obtain precise split probabilities increases significantly with the size of the dataset, and further improvements to the MCMC scheme may be necessary to reduce these computational overheads.

There are a number of features that would help to further improve convergence and mixing of the MCMC samplers. Chief among these is the introduction of a parallel tempering scheme, allowing for more efficient traversal between well-separated modes. We have developed such an implementation, although this currently requires somewhat more testing before it can be incorporated into the main software release.

Chapter 4

Representation of alignment uncertainty using directed acyclic graphs

As discussed in the previous chapters, a number of approaches have been developed in recent years to generate collections of alignments according to their probability, yielding information about the distribution of alignments rather than simply reporting a single optimum. However, currently this type of probabilistic information is not widely used in the context of downstream inference, and this is in part due to a lack of obvious methodology for making use of a distribution over alignments.

In a Bayesian context this entails representing the approximation to the posterior distribution over alignments, given a collection of samples. We shall present here a graph-based formulation that allows for a compact representation of this distribution, permitting algorithms to be designed for efficient inference on exponentially large sets of alignments derived from a collection of samples. As well as allowing for more reliable estimation of posterior probabilities, this approach allows for summary alignments to be generated that maximise expected accuracy or minimise expected loss.

4.1 Representing the distribution of sampled alignments

4.1.1 Mapping columns to dynamic programming tables

A multiple sequence alignment can be represented as a path through a multidimensional matrix; an edge from one cell of the matrix to an adjacent cell represents a particular set of homology statements, synonymous with column in the alignment. It is a straightforward extension to consider a *set* of alignments as a set of paths in such a matrix (Bucka-Lassen *et al.*, 1999).

To formalise this intuition, we introduce a bijection between the set of alignment columns and the set of edges connecting cells in the multidimensional dynamic programming matrix, based on the coding scheme described in the supplementary section of Satija *et al.* (2009). More specifically, a column X containing N rows can be mapped to an N-tuple $C(X) = (c(X_1), \ldots, c(X_N))$, where $c(X_i)$ is defined as

$$c(X_i) = \begin{cases} 2j-1 & \text{if } X_i = s_j^{(i)} \\ 2j & \text{if } X_i = \text{gap, between } s_j^{(i)} \text{ and } s_{j+1}^{(i)} \end{cases} \tag{4.1}$$

such that $C(X)$ corresponds to the coordinates of the midpoint of an edge connecting two cells in the matrix. It is then possible to map any global alignment to a path from the start to the end of the dynamic programming matrix (*see Figure 4.1*).

The initial and terminal columns, $X^{(0)}$ and $X^{(T)}$ (dashed columns in Figure 4.1) can be thought of as all-gap columns preceding the first characters and following the last characters of the sequences, respectively. These will therefore be encoded as $C(X^{(0)}) = (0, \ldots, 0)$ and $C(X^{(T)}) = (2L_1, \ldots, 2L_m)$ where L_i is the length of the i^{th} sequence.

The definition of this mapping can be extended to an entire alignment, A, such that $C(A)$ denotes the (ordered) set $(c(A^{(1)}), \ldots, c(A^{(L)}))$.

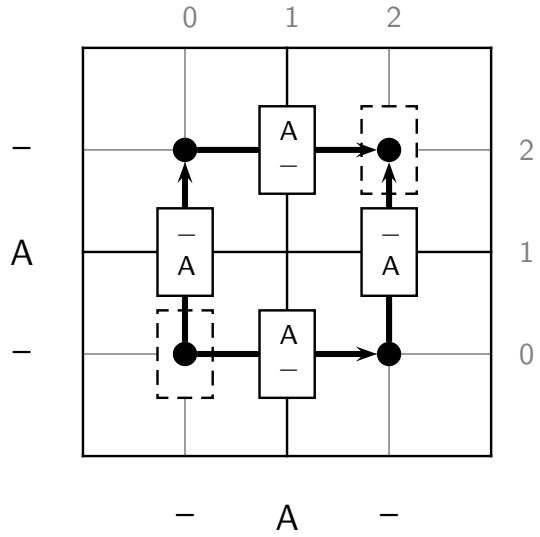


Figure 4.1: Correspondence between alignment columns and edges connecting cells in a dynamic programming matrix, illustrated for pairwise alignment. In order to permit a directed acyclic graph representation of the space of possible alignments, each column is given a code that distinguishes between gaps based upon where they occur in the alignment. The coding for each column represents a bijection to the edges connecting cells in a dynamic programming table (circles, in the above figure). Each path from the first cell to the last cell (here $(0, 0)$ and $(2, 2)$, respectively) represents a valid alignment.

4.1.2 Intersections between alignments

The paths corresponding to a particular set of alignments may intersect at one or more points in the matrix; as first discussed by Bucka-Lassen *et al.* (1999), subpaths can be 'spliced' at these points in order to generate new alignments. This approach was originally used to create an augmented search space for locating an optimal alignment (Bucka-Lassen *et al.*, 1999; Schwikowski and Vingron, 2003), and more recently has been used as part of a progressive alignment algorithm that keeps track of suboptimal alignments (Szabó *et al.*, 2010).

The types of intersections fall into two categories, as illustrated in Figures 4.2 and 4.3. The first of these, which we term an *interchange*, results when two or more sampled alignments contain the same column, but with a different predecessor and successor, as shown in Figure 4.2. The second type of intersection is termed a *crossover*, whereby two or more sampled alignments contain pairs of *equivalent* columns, as shown in Figure 4.3. Each interchange or crossover can result in a multiplication of the number of possible ways of recombining the sampled alignments, such that the total number of alignments is greatly increased.

As a result of this, an initial set of alignments sampled according to a particular model can be used to generate a much larger set of alignments sampled according to the same distribution, as we shall examine in further detail in the subsequent section.

4.1.3 Equivalence classes of columns

In order to delineate the ways in which a set of columns can be recombined to form new alignments, we introduce the *predecessor* and *successor* functions, f_P and f_S respectively. The functions f_P and f_S take the coordinates of a column X as input, and return the coordinates of an equivalence class of columns, corresponding to the midpoint of the predecessor

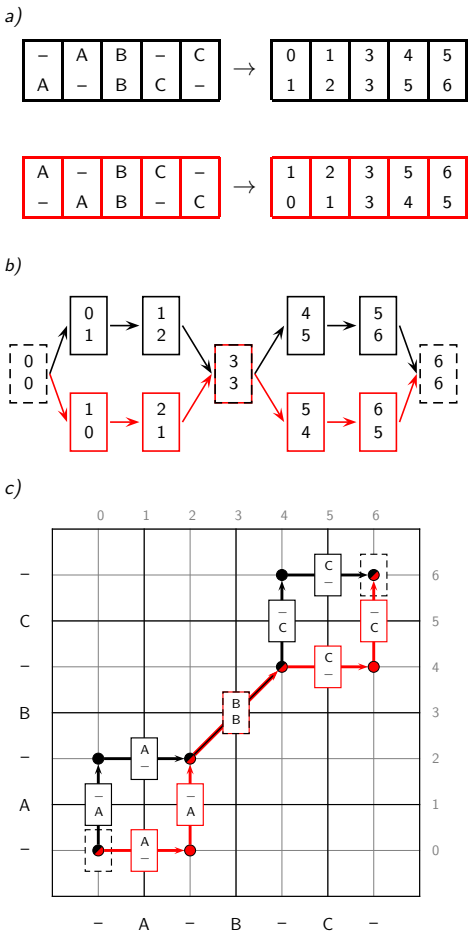


Figure 4.2: Interchanges between alignments can result in a multiplication of the number of possible paths through the DAG. *a)* Two alignments coded under the map C, as described in equation (4.1). *b)* The resulting alignment DAG contains an interchange column, such that there are four paths through the DAG, arising from only two alignments. *c)* Correspondence between alignment columns and edges connecting cells in a dynamic programming matrix.

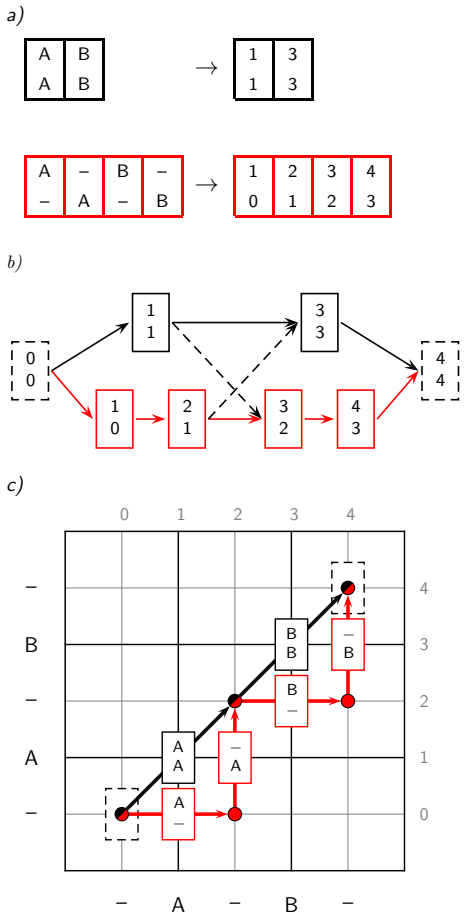


Figure 4.3: Crossovers between two alignments containing no interchange columns. *a)* Two alignments coded under the map C, as described in equation (4.1). *b)* The resulting alignment DAG allows for crossovers between these alignments, such that there are four possible paths through the DAG, two of which include pairs of columns that are not observed in the input alignments (dashed lines). *c)* Correspondence between alignment columns and edges connecting cells in a dynamic programming matrix.

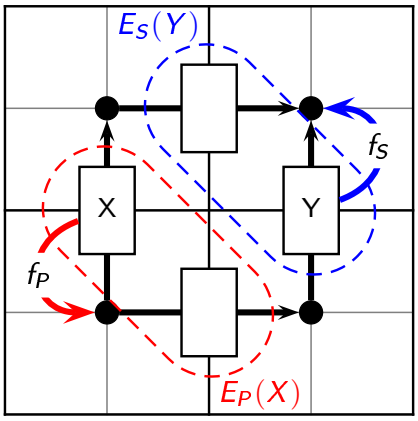


Figure 4.4: Predecessor and successor functions, and equivalence classes of columns. The predecessor and successor functions (f_P and f_S respectively) map from columns (edges) to nodes (circles) in the dynamic programming matrix. All columns mapping to a particular node under f_P share the same set of possible predecessor columns, and are grouped together in an equivalence class, denoted by E_P (shown in red). An analogous definition holds for E_S (blue).

(respectively successor) cell in the multidimensional matrix. Each column mapping to a particular f_P- or f_S-equivalence class can follow the same set of predecessor or successor columns, respectively (*see Figure 4.4*).

Denoting the ith coordinate of the output by $f_P(X)_i$ and $f_S(X)_i$, the functions are defined such that

$$f_P(X)_i = c(X_i) - c(X_i) \bmod 2 \tag{4.2}$$

$$f_S(X)_i = c(X_i) + c(X_i) \bmod 2 \tag{4.3}$$

The original column coding is then uniquely recovered by the backwards mapping

$$C(X) = (f_P(X) + f_S(X))/2 \tag{4.4}$$

The equivalence class $E_P(X)$ is then defined as the set of columns, $\{X' \mid f_P(X') = f_P(X)\}$, with $E_S(X)$ similarly defined.

Using the definitions above, a column X' is a predecessor of X if and only if $f_S(X') = f_P(X)$, since any path connecting them must pass through the separating equivalence class $E_S(X') \equiv E_P(X)$. We will use the notation $\mathcal{P}(X) \equiv \{X' \mid f_S(X') = f_P(X)\}$ to denote the set of predecessors of X.

4.1.4 The alignment column graph

We can then define the *alignment column graph*, $\mathcal{D}(\Xi)$, of a set of columns, Ξ, as a graph whose nodes are the columns in Ξ, with a directed edge from column X to column X' if and only if $f_S(X) = f_P(X')$, which we write as $X \ltimes X'$. From the definitions in equations (4.2) and (4.3), we have $f_P(X) < f_S(X)$ for all X, in the sense that $f_P(X)_i \leq f_S(X)_i$ for all i, with no column having $f_S(X) = f_P(X)$ unless it consists of all gaps. This ensures that the alignment column graph is acyclic, since it is never possible to return to the same equivalence class by following a set of directed edges in the graph.

Each directed path through the column graph generates a valid alignment; a *global alignment* is a valid alignment that begins at $X^{(0)}$ and ends at $X^{(T)}$, such that the number of possible global alignments is equal to the number of distinct paths in $\mathcal{D}(\Xi)$ that lead from $X^{(0)}$ to $X^{(T)}$. This is typically very large, growing rapidly with the number of intersection points between the alignments used to generate the graph (*see Figure 4.12*).

Implicit in the definition of the mapping in equation (4.1) is a distinction between gaps based on their position in the alignment, such that the two situations shown in Figure 4.1 represent distinct alignments, each yielding two different pairs of columns. This assumption is necessary in order to generate a sparse graph; treating all gaps as equivalent is tantamount to replicating each gap-containing column onto all parallels, such that the graph in general becomes maximally dense, making efficient algorithms difficult to implement (*see Figure 4.5*).

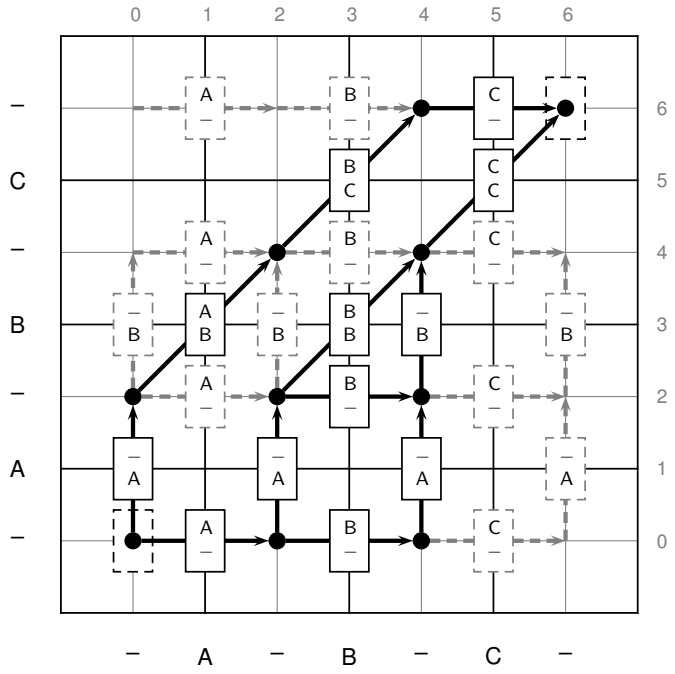


Figure 4.5: If gaps are not distinguished based upon their position in the alignment, it is effectively the same as replicating all gap-containing columns onto all parallels in the graph. In the pairwise case shown above, this is equivalent to replicating each gapped column onto all horizontal and vertical parallels (shown by dotted grey columns and edges in the figure above). This means that the graph in general becomes maximally dense, such that the complexity of any algorithms scales in the same way as the full dynamic programming problem. In contrast, by differentiating between columns based upon where the gaps occur, a sparse graph is retained.

4.2 Probability distributions on alignment DAGs

Due to the high-dimensional nature of the alignment space, in any particular set each alignment will typically occur with a very low frequency; even the most likely alignment may only be sampled once, if at all (Hamada and Asai, 2012; Lunter *et al.*, 2005b). As such, the relative probabilities of entire alignments are difficult—if not impossible—to estimate directly by their observed frequencies. However, a particular column may occur in many different alignments, allowing the *marginal* probability of each column, averaged over all alignments, to be estimated much more efficiently (Lunter *et al.*, 2005b; Redelings and Suchard, 2011). As we shall discuss, they also represent useful summary statistics of the full distribution.

4.2.1 Alignment probabilities in terms of pair marginals

For general evolutionary models, the DAG can be used to construct a factored approximation to the full distribution over alignments; this factored distribution corresponds to a graphical model with dependencies between neighbouring columns defined by the edges in the DAG. The probability of an alignment corresponding to a path through the DAG, can then be written in the form

$$p(A) = p(A^{(1)}) \prod_{i=2}^{L} p(A^{(i)} \mid A^{(i-1)}) \tag{4.5}$$

where

$$p(A^{(i)} \mid A^{(i-1)}) = p(A^{(i)}, A^{(i-1)}) / p(A^{(i-1)}). \tag{4.6}$$

4.2.2 Motivations for using factored approximations

There are three main reasons for making use of factored approximations of this type:

i) The number of possible column pairs is many orders of magnitude lower than the number of alignments, such that pair marginals can be estimated much more reliably from observed frequencies. These can then be used to construct more accurate estimates of the overall joint probability.

ii) Expression of the joint in terms of pair-marginals allows for interchanges in the alignment DAG (*cf.* *Figure* 4.2), allowing many alternative alignments to be generated from an initial collection of samples.

iii) Factorisation of the probability into a product of local terms allows for efficient algorithms to be implemented on the DAG structure.

We discuss these factors in further detail below.

4.2.3 Kullback-Liebler divergence

For evolutionary models based on first-order hidden Markov models (HMMs) (such as the one shown in Figure 4.6), the pair-marginal representation is exact, since the dependencies in the model are equivalent to those in the DAG. For models with non-local dependencies between columns, the deviation between the true distribution over alignments, $p(A)$, and an approximation, $q(A)$, can be measured using the Kullback-Liebler (KL) divergence

$$d(p \parallel q) = \sum_A p(A) \frac{\log p(A)}{\log q(A)} \tag{4.7}$$

$$= \text{const.} - \sum_A p(A) \log q(A) \tag{4.8}$$

Minimising the KL divergence for a fixed $p(A)$ is equivalent to maximising the *relative entropy*,

$\sum_A p(A) \log q(A)$, subject to restrictions on the form for q.

For DAG-based representations, $q(A)$ can be factored along the edges of the DAG;

writing the log of the product of conditionals as a sum of logs, the relative entropy can be written in the form

$$\sum_A p(A) \log q(A) = \sum_A p(A) \sum_X \sum_{X' \ltimes X} \mathbb{1}(X \in A)\mathbb{1}(X' \in A) \log q(X \mid X') \tag{4.9}$$

$$= \sum_X \sum_{X' \ltimes X} p(X \mid X') \log q(X \mid X') \tag{4.10}$$

Using a Lagrange multiplier to enforce the normalisation of $q(X \mid X')$, the distribution maximising equation (4.10) satisfies the following equation, for all X, X'

$$0 = \frac{\partial}{\partial q(X \mid X')} \left[p(X \mid X') \log q(X \mid X') + \lambda \left(1 - \sum_{X'' \in E_P(X)} q(X'' \mid X') \right) \right] \tag{4.11}$$

$$= \frac{p(X \mid X')}{q(X \mid X')} - \lambda \tag{4.12}$$

such that the divergence is minimised with $q(X \mid X') = p(X \mid X')$, which corresponds to setting the pairwise distributions in equation (4.6) to be equal to the true pair marginals. This is equivalent to the result stated in Theorem 11.1 of Cowell *et al.* (2007).

4.2.4 Mean-field approximation

As well as distributions involving pair terms, we will also consider a *mean-field* type approximation, whereby the conditional distribution of each column is averaged over all predecessors. Replacing $q(X \mid X')$ by $q(X \mid \mathcal{P}(X))$ in equation (4.12), and summing over X', it is clear that the KL divergence is also minimised by writing these conditionals in terms of the corresponding marginal distributions:

$$q(X \mid \mathcal{P}(X)) = p(X \mid \mathcal{P}(X)) \tag{4.13}$$

$$= p(X, \mathcal{P}(X)) / p(\mathcal{P}(X)) \tag{4.14}$$

$$= p(X) / \sum_{X'' \ltimes X} p(X'') \tag{4.15}$$

where $p(X \mid \mathcal{P}(X))$ is the probability of column X given that one of its possible predecessors is in the alignment. The third line uses the identities $p(X, \mathcal{P}(X)) \equiv p(X)$ (since a column can only be present if one of its predecessors is present), and $p(\mathcal{P}(X)) \equiv \sum_{X'' \ltimes X} p(X)$ (since only one member of an equivalence class can be present in any particular alignment, due to the acyclic nature of the graph).

An important corollary of the expression in equation (4.15) is that single-column marginals are sufficient to reconstruct the mean-field approximation to the joint probability; this has several important consequences, as we shall discuss below.

4.2.5 Motivations for using the mean-field approximation

The mean-field approximation described above is exact for fully independent sites models, for example pair HMMs with non-affine models for indels. For more general HMMs, there are three major advantages associated with using this approximation rather than the pair-marginal formulation:

i) Since the number of possible columns is substantially less than the number of possible column pairs, it is easier to obtain reliable estimates of single-column marginals from a collection of alignment samples. Hence, the mean-field approximation is likely to be more accurate for lower sample sizes.

ii) The use of single-column marginals allows for crossovers in the alignment DAG (*cf.* *Figure 4.3*), whereas the pair-marginal expression will assign a weight of zero to any pairs that are not observed, hence only permitting interchanges of the form shown in Figure 4.2. This allows for a higher effective sample size for the alignments under the mean-field approximation, with more alternative alignments generated from the same collection of samples.

iii) Restricting to single-column marginals more efficient algorithms to be constructed, involving one-step rather than two-step recursions.

In the rest of this section, we examine these points in further detail.

4.2.6 Estimating marginal probabilities

For a pairwise alignment, column marginals can be easily represented using a matrix in which the (i, j) entry contains the marginal probability $p(s_i^{(1)} \diamond s_j^{(2)})$, where $s_i^{(1)}$ and $s_j^{(2)}$ are the ith and jth characters in two sequences $s^{(1)}$ and $s^{(2)}$, and the symbol $\diamond$ denotes homology. When only two sequences are under comparison, dynamic programming recursions allow for the exact computation of these marginal probabilities under certain types of evolutionary models (Metzler *et al.*, 2001; Thorne and Churchill, 1995; Yu and Smith, 1999).

In the multiple sequence case, such exact computations are typically infeasible. However, if we are provided with a set, $\mathcal{A}$, of sampled alignments, an estimate of the marginal probability of each column can be computed as the proportion of the alignments in $\mathcal{A}$ that contain the column, weighted according to the alignment probability. This can be written using the following indicator notation

$$\hat{p}(X) = \sum_{A \in \mathcal{A}} p(A)\ \mathbb{1}(C(X) \in C(A)) \tag{4.16}$$

If we consider a *multiset*, $\mathcal{M}$, containing global alignments sampled one or more times, according to their posterior probability, then the factor $p(A)$ can be replaced by the relative frequencies of the sampled alignments, and the estimator for the marginal written as

$$\hat{p}(X) = n_{\mathcal{M}}(X)/|\mathcal{M}| \tag{4.17}$$

with $n_{\mathcal{M}}(X)$ denoting the number of occurrences of column X across all the alignments contained in the multiset $\mathcal{M}$. If enough alignments are sampled from the correct distribution, the above estimator will converge to the true value $p(X)$.

The conditional marginals can also be computed from local alignments. In this case a common normalising factor of $|\mathcal{M}|$ cannot be used for all columns, such that the quantity

$p(X)$ cannot be estimated. However, due to the mutual exclusivity of the columns following a particular equivalence class as defined by equation (4.2), the normalised conditional marginals can still be recovered from the expression

$$\hat{p}(X \mid \mathcal{P}(X)) = \hat{p}(X \mid E_P(X)) = \frac{n_{\mathcal{M}}(X)}{n_{\mathcal{M}}(E_P(X))} \tag{4.18}$$

where $n_{\mathcal{M}}(E_P(X)) = \sum_{X' \in E_P(X)} n_{\mathcal{M}}(X')$. In this work we will consider only global alignments, in the interests of simplicity.

Marginal probabilities can also be estimated for pairs of columns using observed pair frequencies. However, the space of possible pairs of columns can be much larger than the space of columns, in the worst case by a factor of $O(2^N)$, where N is the number of sequences. Hence, a larger number of alignment samples will be needed to obtain accurate estimates for pair marginals. As we shall see, this means that pair-based reconstructions of joint probabilities are typically less accurate unless a very large number of samples is used.

4.2.7 Reconstructing alignment probabilities from marginals

Generally, with sampling-based procedures such as MCMC, posterior probabilities are estimated via sampled frequencies. However, in the case of a very high dimensional parameter such as a multiple sequence alignment, each point in the space may only be visited once, such that it is not possible to estimate posterior probabilities based on these frequencies.

As discussed above, the set of marginal probabilities for each column (or pair of neighbouring columns) can be used to reconstruct the posterior probability for any particular alignment, via equation (4.5). Although the likelihood for each sampled alignment will often be known as a by-product of the sampling procedure, the *marginal* posterior probability of each alignment after integrating over other unknown parameters (for example indel rates), will typically not be known. Hence, the DAG-based approach presented here represents a useful way to calculate posterior probabilities in such cases. A similar approach

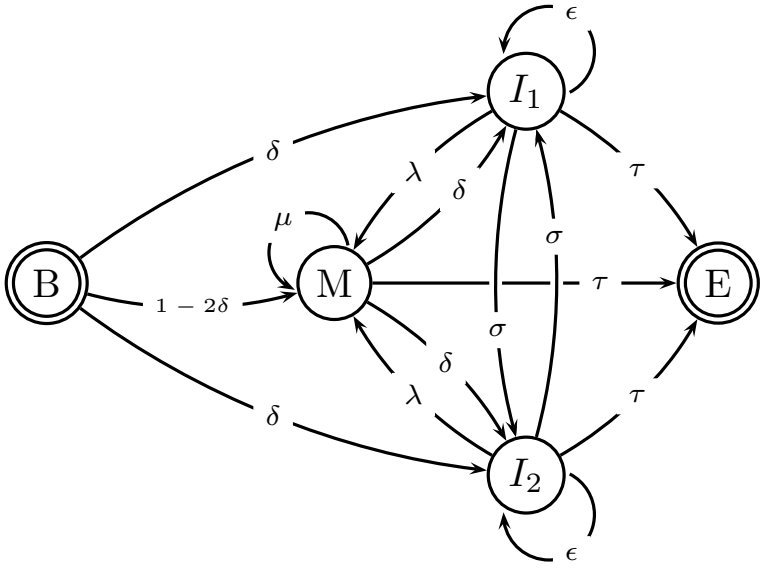


Figure 4.6: The pair-HMM used to sample pairwise alignments between the two globin sequences. Samples generated from this HMM, in conjunction with the Dayhoff substitution model, are shown in Figure 4.7. The states correspond to: B = begin, E = end, M = match, I_1 = indel, and I_2 = indel. We have used the shorthand $\mu = 1 - 2\delta$ and $\lambda = 1 - \epsilon - \sigma - \tau$. For the analyses described in the text, we set $\delta = 0.03$ and $\epsilon = 0.3$, corresponding to an affine gap model; τ was set to the expected sequence length, i.e. $2/(L_1 + L_2)$. The parameter σ, representing the probability of independent adjacent insertions, was set to 0.1, reflecting the fact that insertions may be more common in certain regions of a protein, such as flexible loops. Very similar results were observed with small variations on these parameter values. Sampling was carried out using the algorithms described by Durbin *et al.* (1999).

has been used recently to compute the posterior probabilities of phylogenetic trees based on the probabilities of each of the constituent clades, under the assumption of conditional independence between clades (Larget, 2013).

As an illustration of this procedure, a set of pairwise alignments were sampled from the pair-HMM in Figure 4.6, combined with the Dayhoff amino acid rate matrix (Dayhoff *et al.*, 1978), for two globin sequences (sampled alignments illustrated in Figure 4.7). As shown in Figures 4.8 and 4.10, the DAG-based estimates of the posterior probability converge towards the true probability as the number of samples is increased, reaching a good agreement after just 200 samples.

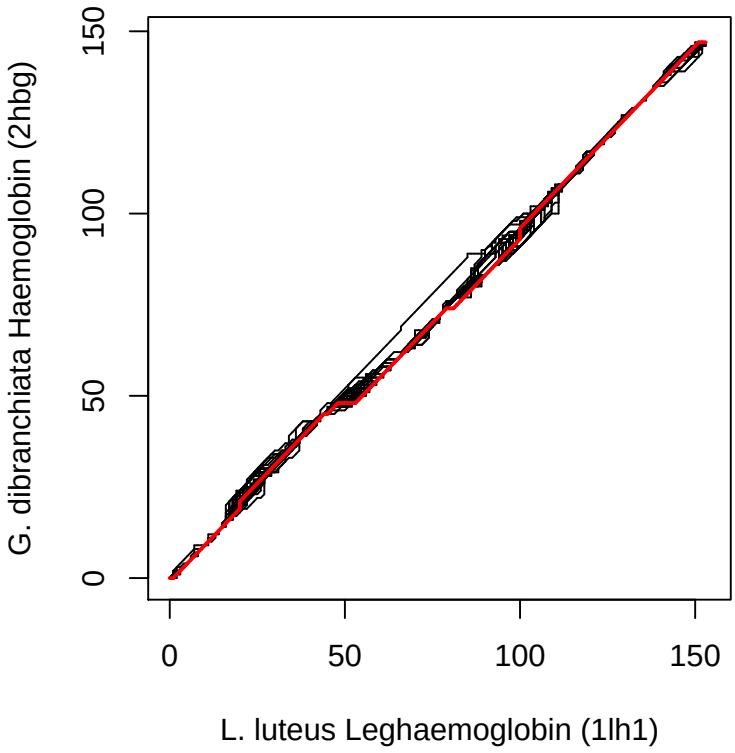


Figure 4.7: A set of 100 **pairwise alignments sampled directly from the pair-HMM shown in Figure 4.6, for two globin sequences.** Overlaid in red is the structural alignment taken from the HOMSTRAD database (Mizuguchi *et al.*, 1998). Despite strong similarity between the alignments, each sample is unique, such that it is not possible to estimate posterior alignment probability estimation on the basis of whole alignment frequency.

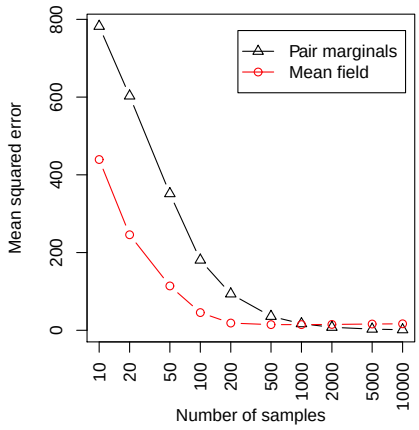


Figure 4.8: Mean squared error in the approximation to the true posterior, as a function of the number of alignment samples. Shown for the pairwise globin example. Although the pair-HMM involves neighbour-dependent terms (leading to an affine gap penalty), the mean-field approximation leads to a better estimate of the true posterior until around 1000-2000 samples are taken. This is due to the presence of intersections between paths in the alignment DAG, which allows for a higher effective sample size to be obtained from the same number of alignments.

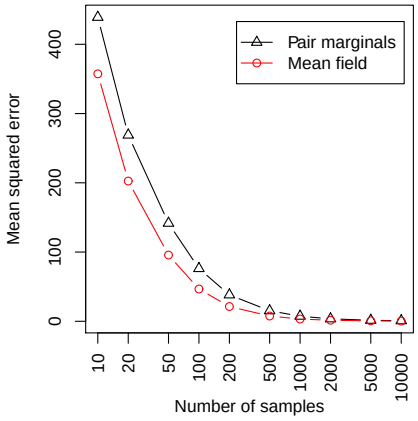


Figure 4.9: Mean squared error in the approximation to the true posterior, for a neighbour-independent HMM. Shown for the pairwise globin example, with $\delta = \epsilon = \sigma$, such that the likelihood is completely site-independent. In this case, the mean-field, single-column marginal estimate always dominates the pair marginal estimate, due to the increased effective sample size.

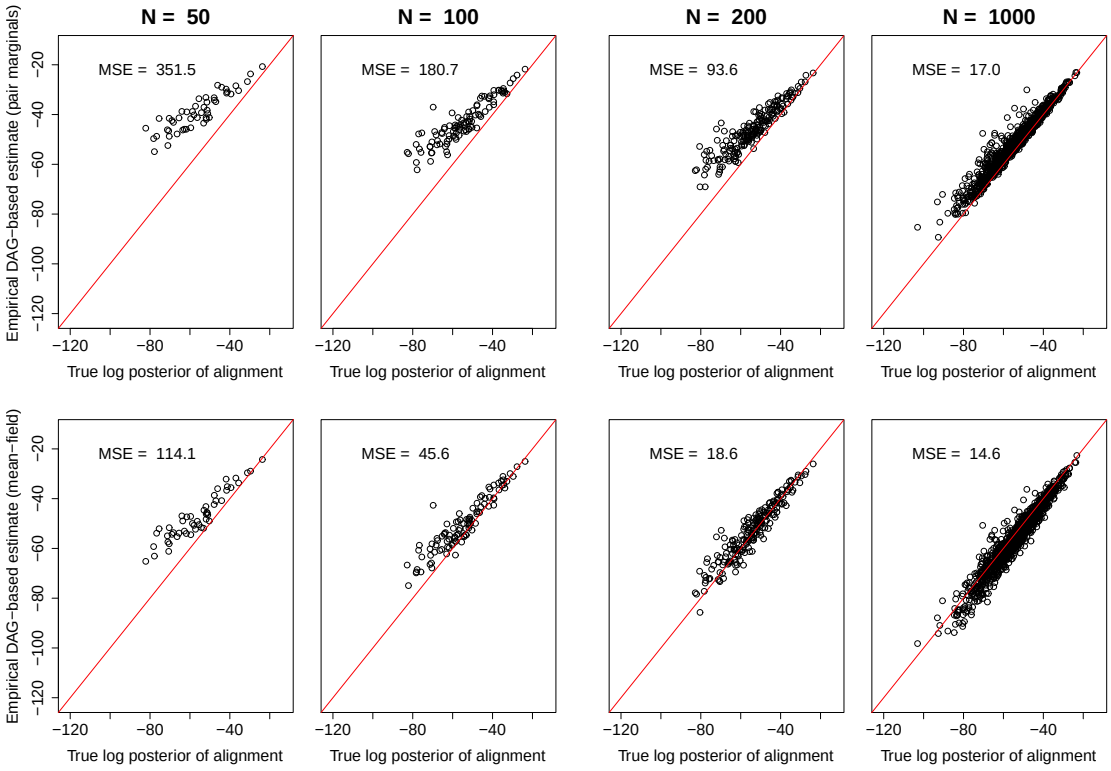


Figure 4.10: As more alignment samples are taken, the DAG-based estimate of the posterior probability for each alignment converges towards the true probability. The DAG-based probabilities already yield a good estimate when the number of alignments, N, is just 100. Shown on the top row are the reconstructed probabilities derived using pair marginals, and on the bottom using the mean field approximation, with the line $y = x$ overlaid in red. Since each sampled alignment is generally observed only once, the posterior probability estimated directly from alignment frequency would be $1/N$ in each case above. The DAG methodology therefore offers a clear advantage for the purposes of computing posterior alignment probabilities. The mean-field approximation results in a lower mean-squared error (MSE), due to the higher effective sample size (*see Figure 4.8*).

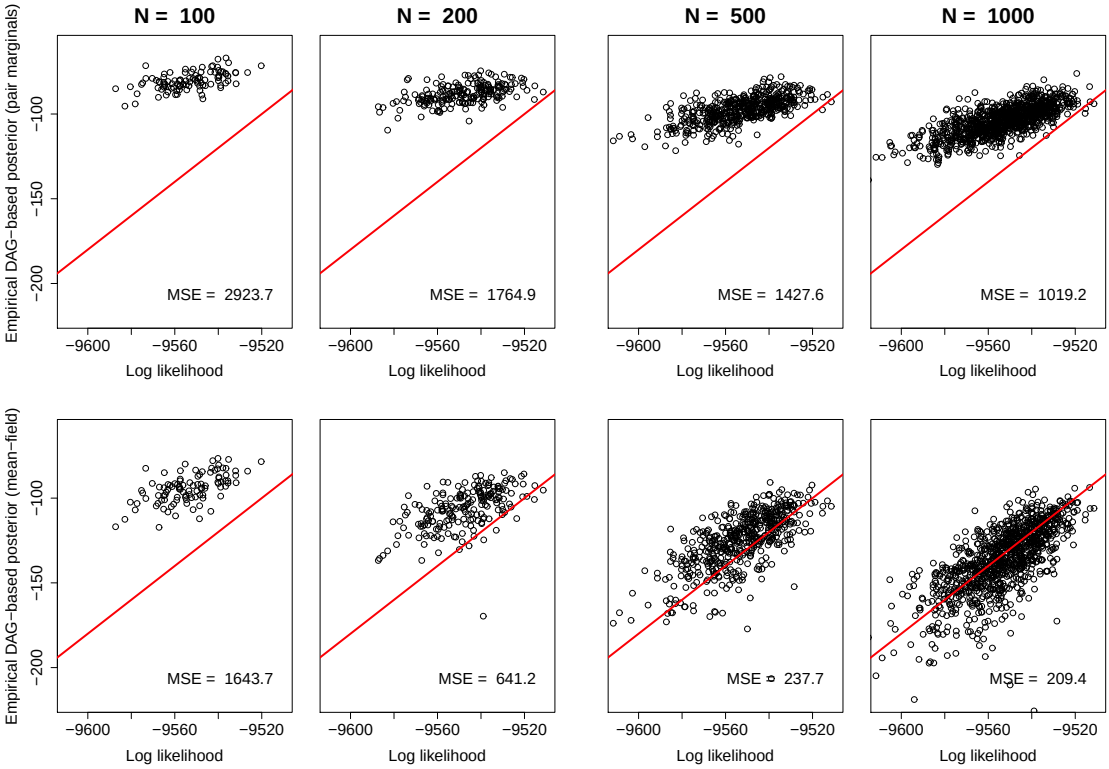


Figure 4.11: For a larger multiple sequence alignment, the mean-field approximation to the posterior (bottom row) converges much more quickly than the pair marginal estimate. Results shown for the simulated dataset described later in the main text; despite the fact that the indel model used (TKF92 (Thorne *et al.*, 1992)) includes neighbour-dependent terms. This is due to the fact that column marginals can be estimated more reliably than pair marginals, combined with the fact that allowing crossovers in the DAG results in a higher effective sample size (*see Figure 4.12*). In this case the true posterior probability is not known, but the log likelihood (conditional on specific values of the other unknown parameters) is known. Since the log likelihood is likely to be linearly related to the log posterior, convergence can be gauged approximately by assessing the fit to a relationship of $y = x + k$ (overlaid in red, with k, the approximate normalising constant, chosen to match the distribution to which the mean-field approximation converges, here $k = -9420$).

For lower numbers of samples, the estimates are more accurate for the more probable alignments, since the more extreme regions of the space are sampled with lower probability, and hence converge more slowly.

Although both pair-marginal and mean-field estimates converge in this case at a similar rate, closer analysis shows that the mean squared error in the approximation to the true posterior is considerably less for the mean-field approximation. This suggests that the improvement obtained by summing over a larger number of paths (*see Figure 4.12*) outweighs the approximation introduced by averaging over predecessor states, although eventually at around 2000 samples the pair-marginal estimates begin to dominate the mean-field approximation (*see Figure 4.8*), since the true pair-HMM involves neighbour-dependent terms. The precise location of this crossover point will depend on the degree of neighbour dependency; for a completely site-independent model (e.g. the pair-HMM in Figure 4.6 with $\delta = \epsilon = \sigma$), the single-column marginal estimate always dominates (*see Figure 4.9*).

This same pattern is observed in a more striking fashion for a larger, 10-sequence alignment, as shown in Figure 4.11. Moreover, since the space of possible alignments increases very rapidly with the number of sequences, the benefit of using the mean-field approach to boost the effective sample size is greater in the multiple-sequence case, resulting in much faster convergence of the posterior estimates (*see Figure 4.11*).

4.2.8 Approximate summation over all alignments

Examining the number of paths in the DAG as a function of the number of alignment samples shows a super-exponential relationship when crossovers are allowed, whereas restricting to observed column pairings increases close to exponentially (*see Figure 4.12*). For the pairwise example discussed above, the total number of paths in the DAG tends towards a maximum of 10^{113}, equal to the number of routes through the dynamic programming matrix.

In the pairwise case, where it is possible to analytically compute the sum over all align-

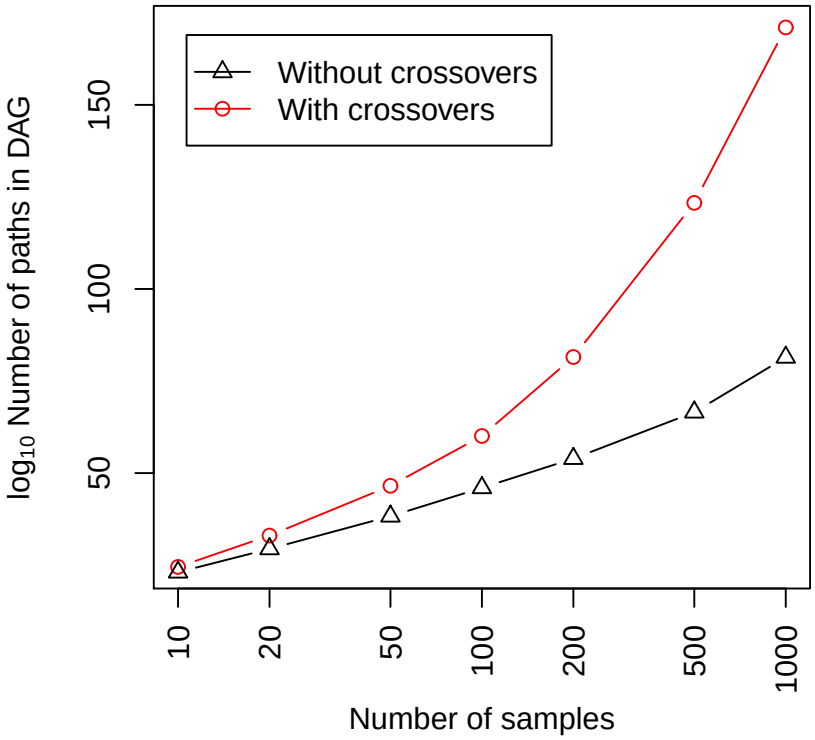


Figure 4.12: The number of paths through the alignment column graph as a function of the number of alignments used to generate the graph. Shown for a set of 10 sequences simulated using DAWG (simulation procedure described in the main text). When crossovers are allowed (corresponding to a mean-field approximation for the conditional marginal for each column), the number of paths increases super-exponentially, resulting in a much higher coverage of the space of possible alignments, and hence more accurate approximations to the posterior probability for each path (*see Figure 4.11*).

ments, it is possible to examine how much of the posterior mass is contained within the DAG resulting from a particular set of samples. The summation over all alignments contained within the DAG can be carried out using a standard dynamic programming algorithm.

Defining the partial sum for column X as

$$z(X) = \begin{cases} \sum_{X' \ltimes X} z(X') p(X \mid X') & (\textit{pair marginals}) \\ \\ z(E_P(X'))\ p(X)/p(E_P(X)) & (\textit{mean field}) \end{cases} \tag{4.19}$$

where $z(X^{(0)}) = 1$ and

$$z(E_P(X)) = \sum_{X' \in E_P(X)} z(X') \tag{4.20}$$

the total sum can then be written as

$$Z(\mathcal{A}) = z(X^{(T)}_{\mathcal{A}}) \tag{4.21}$$

with $X^{(T)}_{\mathcal{A}}$ denoting the terminal column in the DAG $\mathcal{D}(\mathcal{A})$, as defined in the main text. This quantity can be computed in time and space linearly proportional to the number of columns in the DAG, in contrast to the $O(L^N)$ time and space taken for filling the full N-dimensional dynamic programming table. Replacing $p(X)$ with $\mathbb{1}(p(X) > 0)$ results in an algorithm for computing the number of paths through the DAG.

Figure 4.14 shows that the probability mass contained within the individual samples increases relatively slowly, and encapsulates only a very small fraction of the total. In contrast, as shown in Figure 4.13, the proportion of the posterior mass encapsulated in the set of paths through the alignment DAG increases much more rapidly, reaching in the order of 10-15% of the total posterior mass over the entire set of possible alignments with just 100 samples, increasing to around 80% after including 2000 samples.

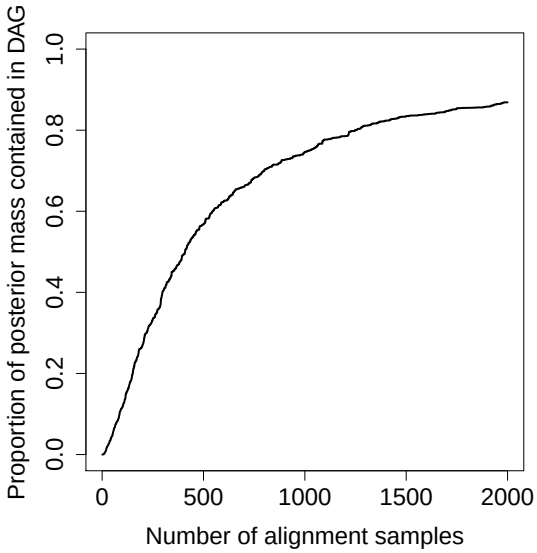


Figure 4.13: The proportion of the posterior mass contained in paths through the DAG increases rapidly with the number of samples. For the pairwise example discussed in the text, the proportion reaches in the order of 10-15% of the total posterior mass with just 100 samples, increasing to over 80% after including 2000 samples.

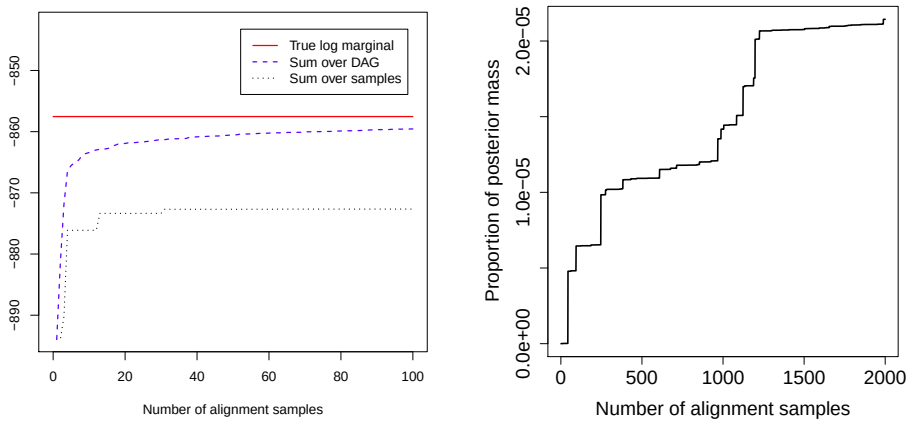


Figure 4.14: The probability mass contained within the individual samples increases relatively slowly, and encapsulates only a very small fraction of the total. In contrast, the proportion of the posterior mass encapsulated in the set of paths through the alignment DAG increases much more rapidly (cf. Figure 4.13).

4.3 Summarising the alignment distribution

Although the set of alignments encoded by the DAG contains a great deal of additional information beyond that contained in any one alignment, there may be situations where a single alignment is desired as a summary of the distribution. Due to the high-dimensional and constrained nature of the state space, standard summary statistics such as the mean are not applicable in this case (Carvalho and Lawrence, 2008).

One of the simplest summaries of the distribution is the *maximum a posteriori* (MAP) alignment. As mentioned earlier, estimation of this quantity directly from sample frequencies is typically very unreliable, since each alignment is typically only sampled once, such that each sample has the same empirical posterior probability. However, as discussed above, the DAG-based approach to estimating posterior probabilities can be used to obtain good estimates of the probability for each possible alignment contained in the DAG. We can then use the fact that the DAG-based log posterior is additive over the columns in the alignment

$$\log p(A) = \log p(A^{(1)}) + \sum_{i=2}^{L} \log p(A^{(i)} \mid A^{(i-1)}) \tag{4.22}$$

such that the path with the maximum posterior can be found using standard dynamic programming algorithms for DAGs (*see Algorithm 6*).

Nevertheless, due to large size of the space of possible alignments, there may be a large number of very similar alignments with very similar posterior probability. Hence, quantities such as the MAP can be poor summary statistics of the distribution (Green and Mardia, 2006; Lunter *et al.*, 2005b; Redelings and Suchard, 2005a). Instead, we will consider alternative types of summary alignments that account for the uncertainty contained within the DAG.

4.3.1 Loss function formulation

The problem of choosing a single summary alignment can be approached within a decision theoretical framework, whereby the choice of summary is designed to minimise the expected value of a particular loss function, also known as the *posterior risk* (Carvalho and Lawrence, 2008). For a loss function defined in terms of alignment *accuracy*, minimising the posterior risk is equivalent to selecting the *maximum expected accuracy* alignment (Hamada and Asai, 2012; Hamada *et al.*, 2011; Roshan and Livesay, 2006).

The *loss* of an alignment, A, with respect to a reference alignment, A', will be denoted by $L(A \parallel A')$, and represents a penalty associated with choosing alignment A, given that the true alignment is A'. The posterior risk associated with A can then be defined as

$$\mathcal{R}(A) = \mathbb{E}\left[\mathcal{L}(A \parallel A')\right] \tag{4.23}$$

$$= \sum_{A'} p(A')\mathcal{L}(A \parallel A') \tag{4.24}$$

where the sum over A' includes all alignments. The minimum-risk alignment is then $\hat{A} = \arg\min_A \mathcal{R}(A)$.

For loss functions defined as a sum over columns (equivalent to the *pointwise gain* functions discussed by Hamada *et al.* (2011)), we have

$$\mathcal{L}(A \parallel A') = k \sum_{X \in A} \mathcal{L}(X \parallel A') \tag{4.25}$$

where k is independent of A. In order to define the loss for a particular column, we will consider the following four categories of columns in the predicted alignment, A:

True positives (TP)	*Columns correctly present*
False positives (FP)	*Columns incorrectly present*
True negatives (TN)	*Columns correctly absent*
False negatives (FN)	*Columns incorrectly absent*

such that $TP \cup FP \cup TN \cup FN = \Xi$.

Generally we will not be interested in the number of negatives (i.e. columns not included in the alignment), since this will depend on how many alignment samples are used to generate the DAG. We will therefore focus on loss functions of the form

$$\begin{aligned}\mathcal{L}_f(X \parallel A) =& \lambda_{FP}(1 - \mathbb{1}(f(X) \in f(A))) \\ & - \rho_{TP}\mathbb{1}(f(X) \in f(A))\end{aligned} \tag{4.26}$$

$$= \lambda_{FP} - (\rho_{TP} + \lambda_{FP})\mathbb{1}(f(X) \in f(A)) \tag{4.27}$$

where f is a bijective function operating on columns, with $f(A) = (f(A^{(1)}, \ldots, f(A^{(L)}))$, and λ_{FP} and ρ_{TP} are loss/reward functions associated with false positives and true positives respectively.

The posterior risk can then be written as

$$\begin{aligned}\mathcal{R}_f(A) &= \sum_{A'} p(A') \sum_{X \in A'} \lambda_{FP} - (\rho_{TP} + \lambda_{FP})\mathbb{1}(f(X) \in f(A)) \\ &= \sum_{A'} p(A') \sum_{X \in A} \lambda_{FP} - (\rho_{TP} + \lambda_{FP})\mathbb{1}(f(X) \in f(A')) \\ &= \sum_{j=1}^{L_A} \sum_{A'} p(A')[\lambda_{FP} - (\rho_{TP} + \lambda_{FP})\mathbb{1}(f(X) \in f(A'))]\end{aligned}$$

where the second line interchanges A and A', which relies on the bijective nature of f. Defining a weighted marginal probability under a function, f, as

$$p_f(X) = \sum_A p(A)\ \mathbb{1}(f(X) \in f(A)) \tag{4.28}$$

this can be rewritten as

$$\mathcal{R}_f(A) = \sum_{j=1}^{L_A} \lambda_{FP} - p_f(A^{(j)})(\rho_{TP} + \lambda_{FP}) \tag{4.29}$$

$$\propto \sum_{j=1}^{L_A} \frac{\lambda_{FP}}{(\rho_{TP} + \lambda_{FP})} - p_f(A^{(j)}) \tag{4.30}$$

where $p_f(X) = \sum_A p(A)\ \mathbb{1}(f(X) \in f(A))$ is the marginal probability of column X being present according to the mapping specified by f.

The expression in equation (4.30) includes a penalty $g = \lambda_{FP}/(\rho_{TP} + \lambda_{FP})$ for each column, thereby penalising longer alignments by a factor proportional to the penalty on false positives. In contrast to an arbitrarily chosen gap penalty, the penalty, g, has a direct interpretation in this case. It is also a straightforward extension to allow λ_{FP} and ρ_{TP}, and hence g, to depend on the specific column, X, for example penalising a false positive proportionally to the number of non-gap characters contained in the column.

4.3.2 Loss functions corresponding to common accuracy measures

The simplest choice in equation (4.27) is to set $f(X) = C(X)$ as defined in equation (4.1), such that $p_f(X)$ is equal to the marginal probability as defined in equation (4.16). The loss function formulation can also be used to represent commonly used measures of *alignment accuracy*. Perhaps the simplest of these is the so-called *column score*; this measures the proportion of correct columns, but without differentiating between the positions of the gaps. This can be defined more formally by first introducing an alternative column mapping, $C^+(X) = (c^+(X_1), \ldots, c^+(X_N))$, which groups together all columns that contain the same non-gap characters:

$$c^+(X_i) = \begin{cases} 2j - 1 & \text{if } X_i = s_j^{(i)} \\ 0 & \text{if } X_i = \text{gap} \end{cases} \tag{4.31}$$

The column score (C^+-score) for an alignment, A, with respect to a reference, A', can then be defined as $-\mathcal{L}_{C^+}(A \parallel A')$, with λ_{FP} most commonly set to zero. Since we have

$$\mathbb{1}(C(X) \in C(A)) \Rightarrow \mathbb{1}(C^+(X) \in C^+(A)) \tag{4.32}$$

and hence $p_{C^+}(X) \geq p_C(X)$ and $\hat{p}_{C^+}(X) \geq \hat{p}_C(X)$, the C^+-risk, i.e. $\mathcal{R}_{C^+}$, represents an upper bound to the C-risk, $\mathcal{R}_C$. As shown in Figure 4.15, the alignment minimising the C^+-risk will not in general be the same as that minimising the C-risk, although there may be considerable overlap.

4.3.3 Pairwise loss functions

The above approach can easily be extended to make use of a function, f, which splits a column up into a set of pairwise homology statements. It is possible to describe several different types of pairwise accuracy scores using a loss function of the form

$$\mathcal{L}_{pw}(X \parallel A) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \rho_{TP}(X_i, X_j) \mathbb{1}((X_i, X_j) \in A) \tag{4.33}$$

where N is the number of sequences. With $\rho_{TP}(X_i, X_j) = -\mathbb{1}(X_i \neq \text{gap})\mathbb{1}(X_j \neq \text{gap})$, this is equivalent to the commonly used *sum-of-pairs* score (Thompson *et al.*, 1994), and the AMAP alignment metric of Schwartz (Schwartz and Pachter, 2007; Schwartz *et al.*, 2005) can be obtained by setting

$$\begin{aligned} \rho_{TP}(X_i, X_j) = &- \mathbb{1}(X_i \neq \text{gap})\mathbb{1}(X_j \neq \text{gap}) \\ &- G_f \mathbb{1}(X_i = \text{gap})\mathbb{1}(X_j \neq \text{gap}) \\ &- G_f \mathbb{1}(X_i \neq \text{gap})\mathbb{1}(X_j = \text{gap}) \end{aligned} \tag{4.34}$$

where $G_f > 0$.

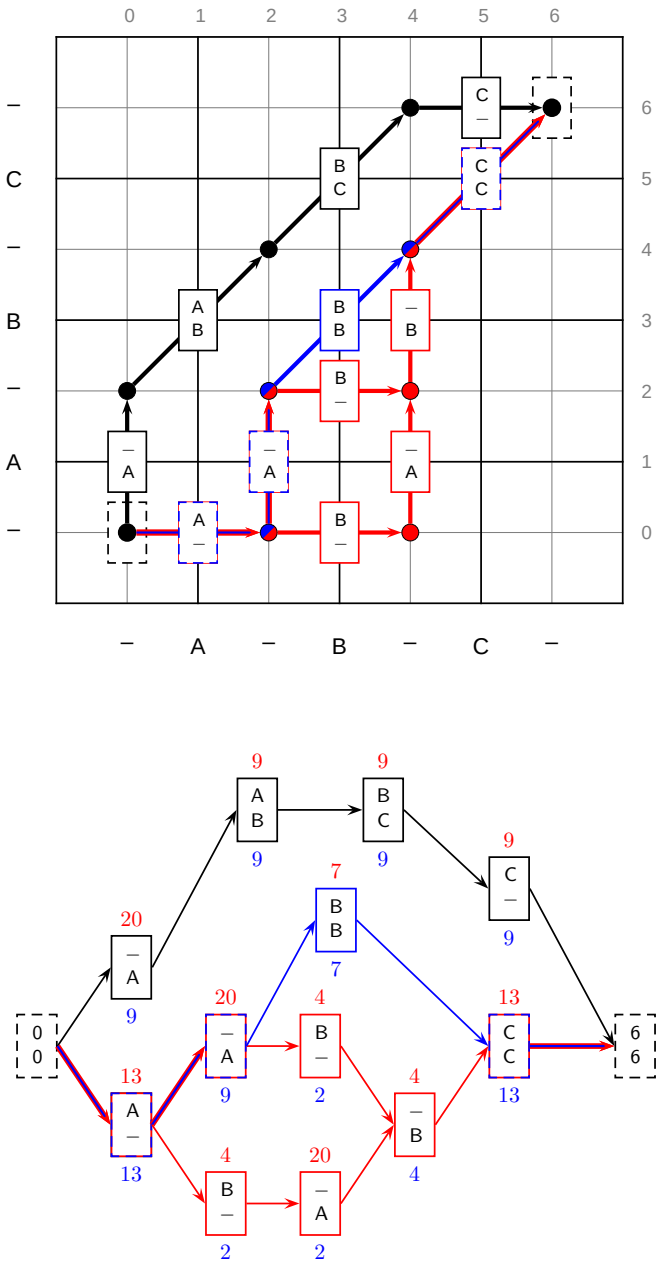


Figure 4.15: The minimum-risk path under the C-based loss function (blue) may not be the same as that under the C^+-based loss function (red). Column frequencies are shown in blue below each column, and the p_{C+} marginals shown in red above (as frequencies from a total of 20 samples). In this case, there are two equivalent paths with the same C^+-score.

4.3.4 Modeller scores

One other class of loss function worth mentioning here is the so-called *modeller* version of each of the aforementioned scores, $\mathcal{L}^m_f(A \parallel A')$, which involve normalising $\mathcal{L}_f(A \parallel A')$ by the length of the predicted alignment, A. For example, the modeller C-score, corresponding to $\mathcal{L}^m_C(A \parallel A')$, was considered by Collingridge and Kelly (2012); as we shall see, the dependence on the length of the predicted alignment precludes the use of exact optimisation algorithms for loss functions such as this.

4.3.5 Efficient algorithms

In general, minimising the expectation of any of the aforementioned loss functions over the space of all possible multiple alignments is a problem whose complexity grows exponentially with the number of sequences (Wang and Jiang, 1994). For the pairwise case, the minimum-risk/maximum expected accuracy problem can be implemented efficiently using standard dynamic programming algorithms (Durbin *et al.*, 1999; Green and Mardia, 2006; Green *et al.*, 2010b; Hamada and Asai, 2012; Holmes and Durbin, 1998; Lunter *et al.*, 2008; Miyazawa, 1995; Ruffieux and Green, 2009; Wolfsheimer *et al.*, 2012), but approximate techniques have generally been used to tackle the multiple sequence version of this problem, including simulated annealing (Bradley *et al.*, 2009; Schwartz, 2007; Schwartz and Pachter, 2007), and greedy (Sahraeian and Yoon, 2010) or progressive alignment algorithms (Do *et al.*, 2005; Liu *et al.*, 2010; Notredame *et al.*, 2000; Roshan and Livesay, 2006).

However, if the solution set is restricted to the (still very large) space of alignments encoded in the DAG, any risk function that is additive over columns [in the sense of equation (4.25)] can be minimised in time linear in the number of columns in the DAG, by making use of efficient maximum-weight path algorithms (*see Algorithm 7; Figure 4.17*). This type of approach was first mentioned by Lunter *et al.* (2005b), and an implementation described by Satija *et al.* (2009) (although that algorithm was designed to maximise the product of

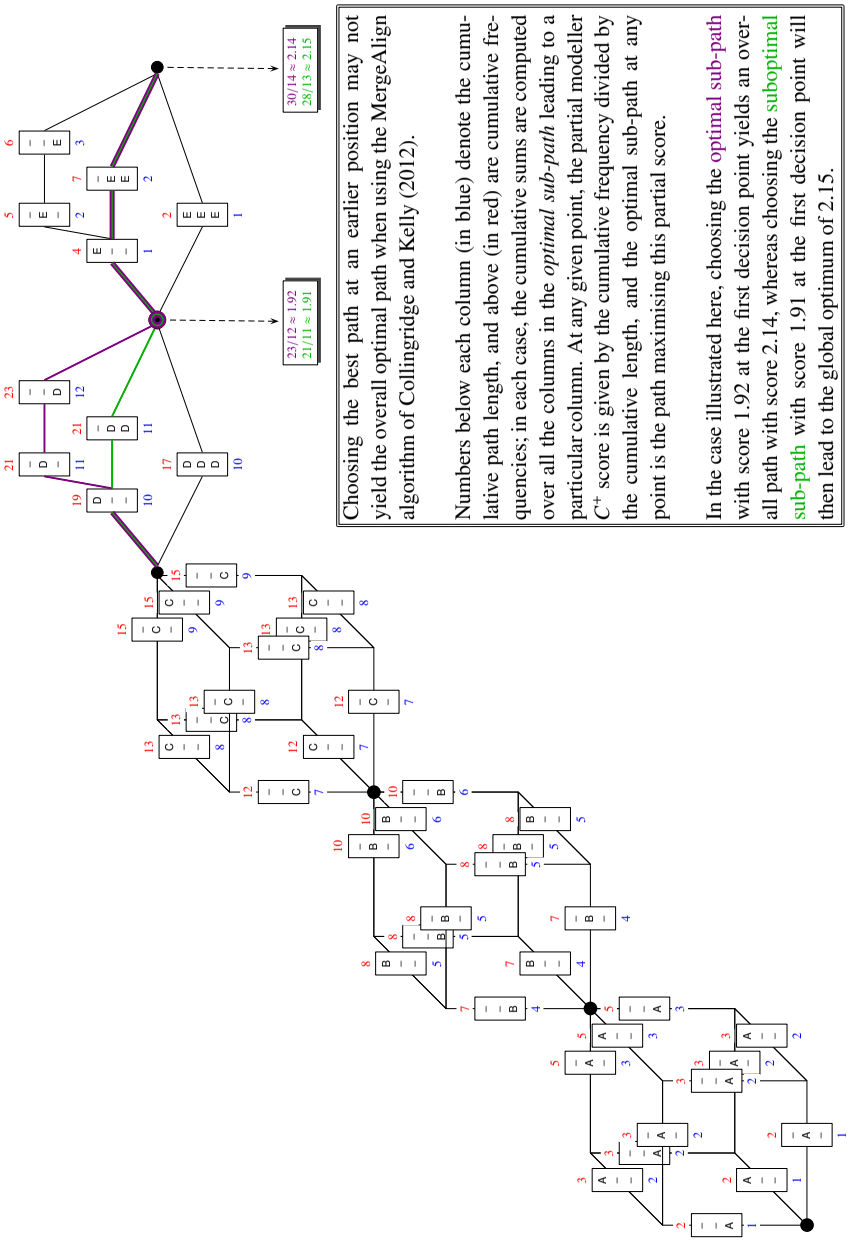


Figure 4.16: Example illustrating a case where the MergeAlign algorithm does not yield the global optimum under the modeller (length-normalised) version of the C-score.

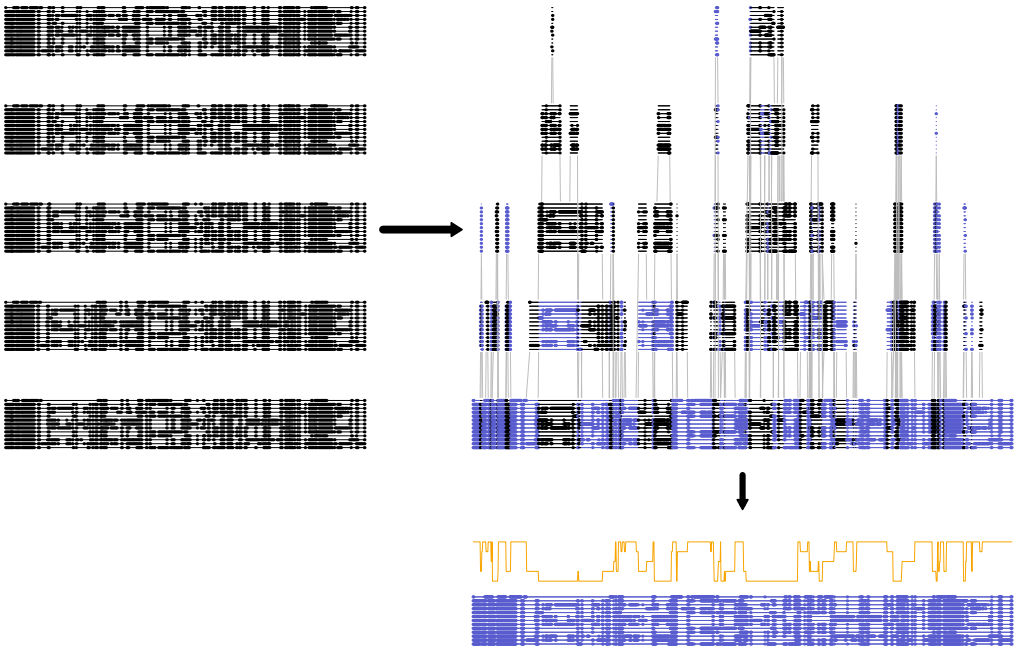


Figure 4.17: A collection of alignment samples can be combined into a DAG structure, and a summary algorithm generated using efficient algorithms. The graph can be visualised by vertically ordering columns based on the longest path length to the end of the DAG (as shown above). Each path represents a valid combination of the columns in the input alignments, with valid recombinations shown as grey lines in the above figure. The *maximum a posteriori* or minimal-risk path can then be found efficiently using linear-time algorithms, yielding a single summary alignment (shown in blue) that accounts for the uncertainty in the alignment set, and can be annotated with posterior probabilities for each column (shown in orange).

column marginals, which does not have an obvious probabilistic interpretation).

The same approach cannot be applied to minimise the risk under modeller variants, however, since the contribution of each column to the partial sum at each step in the dynamic programming algorithm depends on the unknown final alignment length. Collingridge and Kelly recently presented an algorithm, entitled MergeAlign, that proposed to optimise a score of this type, but as shown in Figure 4.16, it is possible to construct counter-examples for which the algorithm does not compute the optimal solution. As we shall illustrate, this lack of optimality can result in significant losses when summarising a set of alignments. Moreover, the same objective, i.e. penalising longer alignments, can be achieved through

the use of a non-zero g parameter as described above, such that the use of modeller variant loss functions is unnecessary.

Algorithm 6 MAP alignment (mean-field)

```
M = { }  //Max. cumulative log posterior for each equivalence class
T = { }  //Traceback hash
π = ( )  //List that will contain the MAP alignment
e  //Represents a vector indexing an equivalence class

function mapPathTo(e)
if M{e} undefined then
    if e = 0 then
        M{e} ← 0
    else
        M{e} ← −∞
        for all X | f_S(X) = e do
            //Increment using mean-field approx. to conditional
            m ← mapPathTo( f_P(X) ) + log(p(X)/p(E_P(X)))
            if m > M{e} then
                M{e} ← m
                T{e} ← X
return M{e}
end function

function traceback( )
e ← f_P(X^(T))
while e ≠ 0 do
    prepend(T{e}, π)
    e ← f_P(T{e})
return π
end function

function mapAlignment( )
mapPathTo( f_P(X^(T)) )
return traceback( )
end function
```

Algorithm 7 Minimal risk alignment, $\arg\min_A \mathcal{R}_f(A)$

```
M = { }  //Max. negative cumulative risk for each equivalence class
T = { }  //Traceback hash
π = ( )  //List that will contain the minimum risk alignment
e  //Represents a vector indexing an equivalence class
p_f(X)  //∑_{A∈D(Ξ)} p(A)p(f(X) ∈ f(A))
g(X)  //Penalty function, defined such that g(X^(0)) = 1

function minRiskPathTo(e)
if M{e} undefined then
    if e = 0 then
        M{e} ← 0
    else
        M{e} ← −∞
        for all X | f_S(X) = e do
            m ← minRiskPathTo( f_P(X) ) + p_f(X) − g(X)
            if m > M{e} then
                M{e} ← m
                T{e} ← X
return M{e}
end function

function minRiskAlignment( )
minRiskPathTo( f_P(X^(T)) )
return traceback( )
end function
```

4.4 Efficient data structures

In representing the alignment DAG, it is essential to do so in such a way that the space complexity of the data structure is less than the total number of paths through the graph, which increases very rapidly with the number of columns. The obvious way to represent a graph is via a list of neighbours for each node, which requires $O(\bar{d}|\Xi|)$ storage, where $|\Xi|$ is the number of observed columns and $\bar{d}$ is the average node in-degree.

However, within the mean-field setting, we can use the predecessor and successor equivalence classes to significantly increase the space efficiency, since each column need only record its predecessor and successor equivalence class. Given the definitions of the predecessor and successor equivalence classes, we can see that each equivalence class is of size at most $2^N - 1$, where N is the number of sequences, since each row can take one of two possible values (gap/character) in each equivalence class, with the restriction that the column cannot be all gaps. In general, the number of equivalence classes is therefore somewhat less than the number of columns, with $|\Xi| = \bar{d}|\mathcal{E}|$, where $1 \leq \bar{d} \leq 2^N - 1$. Using an equivalence-class representation of the DAG structure therefore results in $O(\bar{d}|\mathcal{E}|) = O(|\Xi|)$ space requirements, saving a factor of $\bar{d}$.

Similar gains can be made in time complexity. Since any column in a particular f_P-equivalence class will have the same set of possible predecessors, and similarly for successors, the partial sums required in dynamic programming algorithms can be stored per equivalence class rather than per node, which results in algorithms of $O(|\Xi|)$ time complexity rather than $O(\bar{d}|\Xi|)$ (*see Algorithms 6 and 7 for examples*). In the limit of a large number of short sequences with high uncertainty, this results in going from approximately quadratic time, to time linear in the number of columns.

4.5 Example application: summary alignments for simulated data and BAliBASE

In order to illustrate the utility of the aforementioned procedure, we first simulated sequence data using the program DAWG (Cartwright, 2005), yielding sets of sequences for which the true alignment is known. Data were generated using the sequence evolution simulation tool DAWG (Cartwright, 2005). A random phylogeny of 10 sequences was chosen and fixed (*see Figure 4.18*), and sequences were simulated under the GTR substitution model (rates of substitution for AC, AG, AT, CG, CT and GT were set at 1.5, 3.0, 0.9, 1.2, 2.5 and 1.0; equilibrium frequencies for A, C, G and T were set at 0.20, 0.30, 0.30, 0.20), with the G+I rate heterogeneity model ($\gamma = 0.9$, $\iota = 0.05$), and an indel process with lengths distributed according to a negative binomial $NB(3, 0.7)$ distribution. The indel rate was set to three different values [0.01 (low), 0.02 (medium) and 0.03 (high)], to generate datasets of varying alignment uncertainty. For each indel rate, 50 alignments were generated, yielding 150 datasets overall, each containing 10 sequences, with average sequence length equal to 905 nucleotides.

As a biologically relevant example, we also considered a set of 78 alignments taken from the BAliBASE database, comprising the full-length alignments from the Reference 1 set (Thompson *et al.*, 2005). This set further comprises two subsets, consisting of low sequence identity (*Ref 1a*, ID $<$ 25%) (short: 14, medium: 12, long: 12; average 6.8 sequences per alignment; average sequence length 309), and medium sequence identity (*Ref 1b*, ID $= 20 - 40\%$) (short: 14, medium: 16, long: 10; average 9.0 sequences per alignment; average sequence length 351).

For each dataset, we ran the statistical alignment software StatAlign (Novák *et al.*, 2008), which jointly samples alignments and trees under a stochastic model of substitution, insertion and deletion (Lunter *et al.*, 2005b). StatAlign v1.1 was run using the default settings for nucleotides (for the simulated data), and amino acids (for the BAliBASE data),

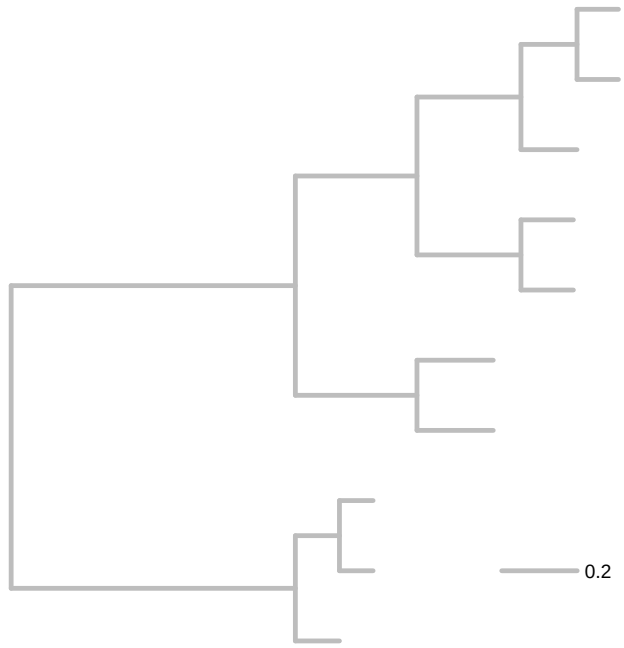


Figure 4.18: The tree used to generate simulated data as described in Section 4.5.

with a burnin of 500,000, and 2 million sampling steps, taking alignment samples every 2000 steps, thus producing 1000 alignment samples for each test case. A Java-based implementation of Algorithm 7 was used to compute a summary alignment minimising the risk under the C- and C^+-based loss functions.

4.5.1 Comparison to other methods

For comparison, we also generated summary alignments for each dataset using the MergeAlign method of Collingridge and Kelly (2012), and a consistency-based approach whereby the alignment samples are used as a library for input to the program T-Coffee (Notredame *et al.*, 2000), using the `-aln` option (Wallace *et al.*, 2006). We call the latter approach S-Coffee, with the 'S' signifying that the T-Coffee method is being used on a library derived from StatAlign alignments.

As shown in Table 4.1, our DAG-based implementation is substantially faster than the other methods. Increasing the indel rate results in higher alignment uncertainty and longer

	Indel rate		
	low	*medium*	*high*
MinRisk (C)	1.5	1.8	2.2
MinRisk (C^+)	1.9	2.4	2.8
MergeAlign	12.0	17.6	22.9
S-Coffee	43.0	48.4	50.9

Table 4.1: Average time (in seconds) taken to generate a summary alignment from 1000 samples, for the three simulated datasets. All tests performed on a single AMD Opteron 2.3GHz core.

alignments, resulting in an increase in runtime for all methods, although the increase is small for the minimum risk algorithm (henceforth referred to as MinRisk). Minimising the risk under the C^+-based loss function incurs an additional overhead due to the time needed to compute the weighted marginal probabilities, $p_{C^+}(X)$, but this takes a matter of milliseconds.

Name	Notation	Definition
C-score	$\alpha_C(\hat{A})$	$\sum_{X\in\hat{A}} \mathbb{1}(C(X) \in C(A))/\lvert A\rvert$
Modeller C	$\alpha^m_C(\hat{A})$	$\sum_{X\in\hat{A}} \mathbb{1}(C(X) \in C(A))/\lvert\hat{A}\rvert$
C^+-score	$\alpha_{C^+}(\hat{A})$	$\sum_{X\in\hat{A}} \mathbb{1}(C^+(X) \in C^+(A))/\lvert A\rvert$
Modeller C^+	$\alpha^m_{C^+}(\hat{A})$	$\sum_{X\in\hat{A}} \mathbb{1}(C^+(X) \in C^+(A))/\lvert\hat{A}\rvert$

Table 4.2: Accuracy measures used to assess the relative performance of the different summary methods. A denotes the true alignment and $\hat{A}$ an estimated alignment.

4.5.2 Accuracy metrics

To assess the performance of each approach, we make use of several measures of alignment accuracy, including the AMA metric of Schwartz (Bradley *et al.*, 2009; Schwartz, 2007), and the column score (C^+-score). In addition, we use the measures shown in Table 4.2.

For the simulated data, accuracy is computed relative to the known true alignments, and for the BAliBASE datasets, relative to the benchmark alignment provided.

Since the minimal $\mathcal{R}_C$ and $\mathcal{R}^+_C$ alignments maximise the expectation of the C- and C^+-

score respectively, it would be expected that these methods perform best under the corresponding scores. The MergeAlign method seeks to maximise the Modeller C score, although as mentioned earlier, the algorithm cannot guarantee an optimal solution. As a pairwise progressive algorithm, the S-Coffee method might be expected to perform best under a sum-of-pairs score, such as the AMA metric.

Given that the absolute value of the accuracy varies substantially over the different datasets, we measure the performance of each method by computing a rank score, which indicates the rank of the accuracy of an alignment, $\hat{A}$, relative to the 1000 samples used as an input (denoted by the multiset $\mathcal{M}$)

$$\text{rank}_\alpha(\hat{A} \parallel \mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{A \in \mathcal{M}} \mathbb{1}(\alpha(\hat{A}) > \alpha(A)) \tag{4.35}$$

A rank of 1 therefore indicates an alignment that is more accurate under measure α than each of the StatAlign samples, whereas a rank of 0 indicates an accuracy lower than any of the individual samples.

4.5.3 Results: simulated data

As shown in Table 4.3, the MinRisk method generally yields summary alignments that are more accurate than the majority of the samples, resulting in a rank score close to 1. As expected, minimising the risk under the C-based loss function results in the highest accuracy under metric α_C, and similarly minimising the risk under $\mathcal{R}_{C^+}$ results in the highest scores under measure α_{C^+}. Interestingly, the MinRisk C^+ method also results in the highest accuracy under the AMA sum-of-pairs metric. In all cases setting $g = 0$ results in the best performance, since these accuracy metrics do not penalise false positives, although setting $g = 0.5$ does not result in a large loss of performance.

In contrast, on these datasets MergeAlign typically yields a summary alignment whose accuracy is close to the median, with a rank score close to 0.5, although performance is

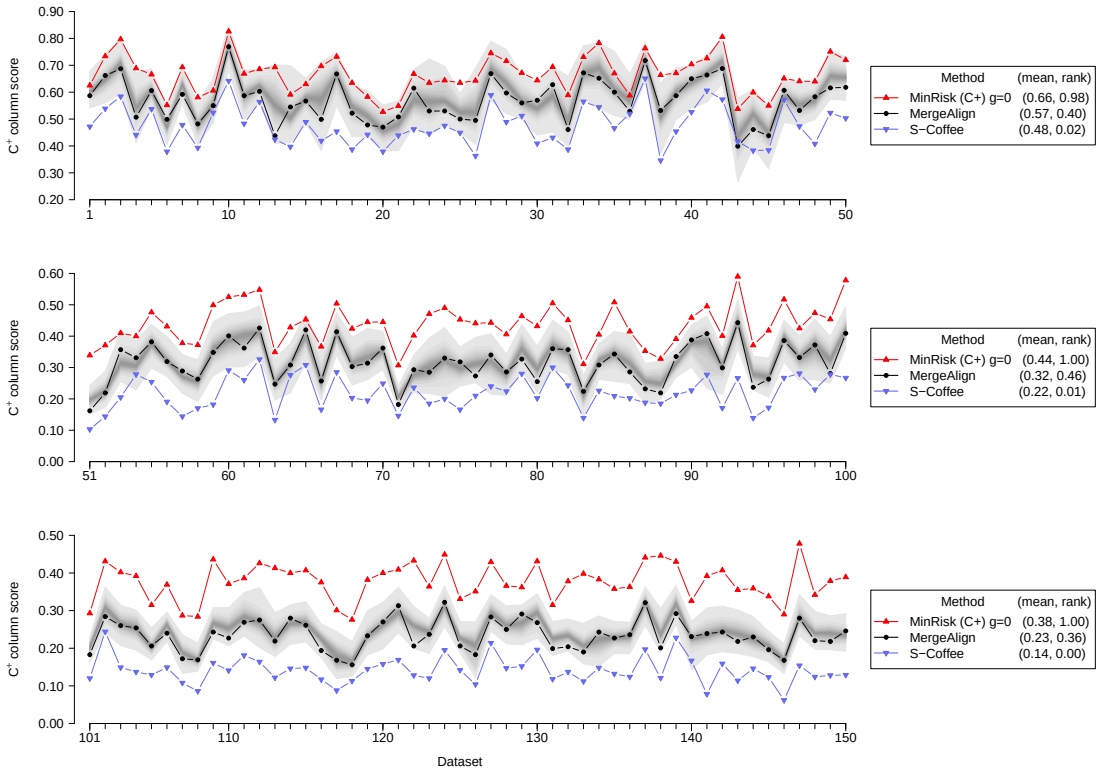


Figure 4.19: Accuracy of summary alignments for simulated data. Results for the MinRisk, MergeAlign and S-Coffee methods shown in red, black and blue respectively, for low (top panel), medium (middle panel) and high (bottom panel) indel rates, with accuracy measured by α_{C^+}. The range of values covered by the 1000 samples is shown in grey, with lighter shading indicating greater distance from the median.

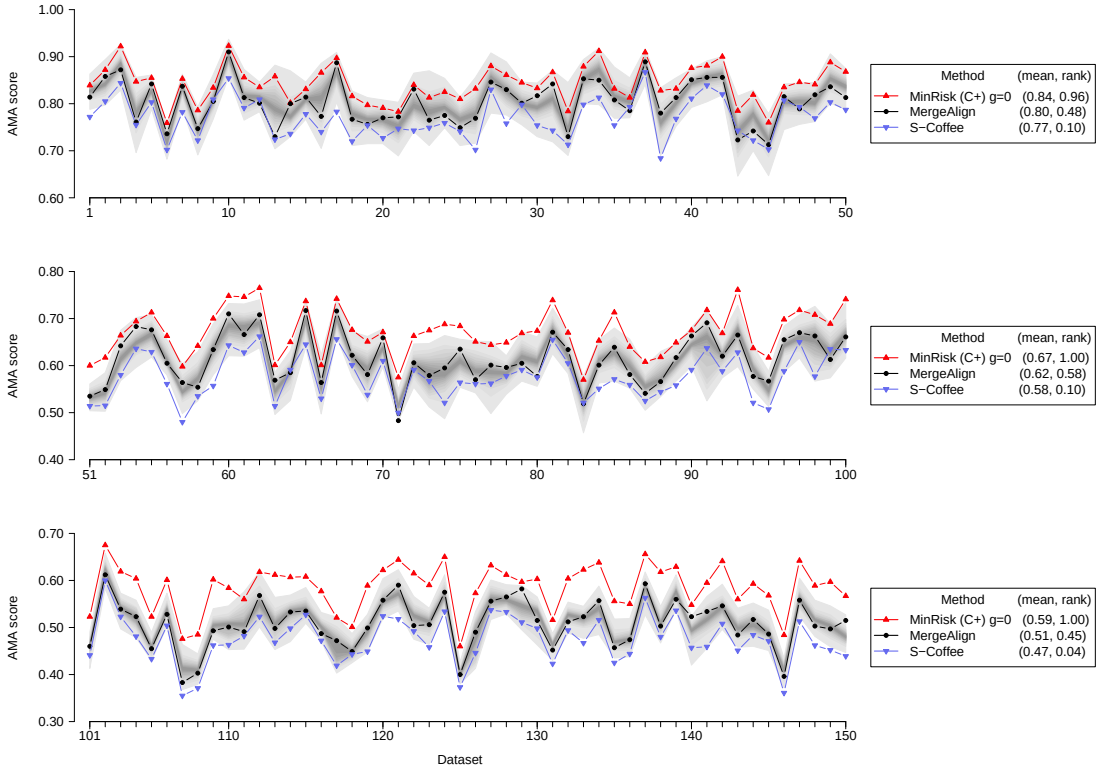


Figure 4.20: Accuracy of summary alignments for simulated data under three different methods as measured by α_{AMA}, for low (top panel), medium (middle panel) and high (bottom panel) indel rates. The range of accuracy values covered by the StatAlign samples is shown in grey, with lighter shading indicating greater distance from the median.

	low indel rate			*medium indel rate*			*high indel rate*		
	α_C	α_{C^+}	AMA	α_C	α_{C^+}	AMA	α_C	α_{C^+}	AMA
MinRisk (C), $g = 0$	**0.91**	0.89	0.90	**0.96**	0.92	0.93	**0.89**	0.88	0.88
MinRisk (C), $g = 0.5$	0.89	0.73	0.84	0.93	0.50	0.78	0.84	0.09	0.40
MinRisk (C), $g = 1$	0.88	0.63	0.80	0.90	0.30	0.65	0.79	0.03	0.28
MinRisk (C^+), $g = 0$	0.86	**0.98**	**0.96**	0.87	**1.00**	**1.00**	0.76	**1.00**	**1.00**
MinRisk (C^+), $g = 0.5$	0.89	0.92	0.92	0.93	0.94	0.94	0.86	0.94	0.94
MinRisk (C^+), $g = 1$	0.89	0.84	0.88	0.91	0.74	0.85	0.83	0.34	0.55
MergeAlign	0.65	0.40	0.48	0.80	0.46	0.58	0.73	0.36	0.45
S-Coffee	0.08	0.02	0.10	0.15	0.01	0.10	0.29	0.00	0.04

Table 4.3: Average rank scores for the different methods on simulated datasets, using the accuracy metrics described in the main text and in Table 4.2. Highest values for each column shown in bold.

	low indel rate		*medium indel rate*		*high indel rate*	
	α^m_C	$\alpha^m_{C^+}$	α^m_C	$\alpha^m_{C^+}$	α^m_C	$\alpha^m_{C^+}$
MinRisk (C), $g = 0$	0.92	0.91	0.96	0.95	0.89	0.92
MinRisk (C), $g = 0.5$	0.93	0.88	**0.97**	0.80	**0.90**	0.35
MinRisk (C), $g = 1$	**0.95**	0.85	0.96	0.65	0.87	0.23
MinRisk (C^+), $g = 0$	0.69	0.88	0.62	0.96	0.56	**1.00**
MinRisk (C^+), $g = 0.5$	0.90	**0.94**	0.95	**0.97**	0.88	0.96
MinRisk (C^+), $g = 1$	0.93	0.92	0.95	0.91	0.88	0.74
MergeAlign	0.74	0.57	0.85	0.67	0.78	0.63
S-Coffee	0.15	0.05	0.22	0.03	0.37	0.00

Table 4.4: Average rank scores for the different methods on simulated datasets, measured using the modeller scores. Highest values for each column shown in bold.

more reasonable under the α_C measure. The progressive heuristic S-Coffee algorithm performs consistently badly in all cases, yielding summary alignments that are typically worse than the majority of the samples used to build the library, suggesting a conflict between the information contained in the samples, and the heuristics used to construct the alignment.

When the modeller variants of the scores are considered (Table 4.4), the general patterns stay much the same, although there is now a benefit observed in increasing the g parameter, since the modeller scores penalise longer alignments. For alignments with more gaps (higher indel rate), the value of g yielding the highest accuracy under the modeller scores tends to decrease (*see Figure 4.21*). This reflects the fact that for cases where the true align-

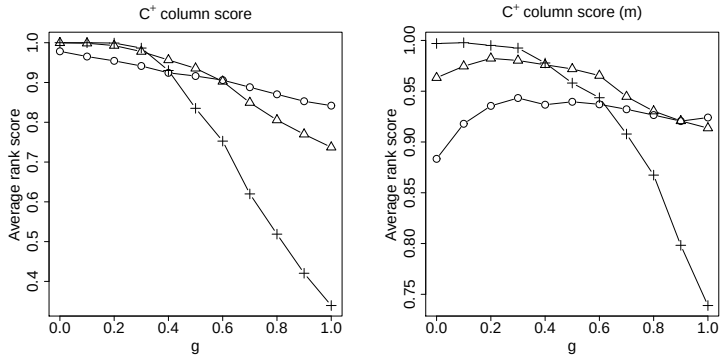


Figure 4.21: Accuracy on the simulated datasets under the α_{C^+} (left) and $\alpha^m_{C^+}$ (right) measures as a function of the g parameter for low ($\circ$), medium ($\triangle$) and high ($+$) indel rates.

ment contains many gaps we may wish to be more lenient with the inclusion of additional columns, allowing the alignment to increase in length. Overall, setting $g = 0.5$ yields the best average performance, which corresponds to a loss function that equally penalises false positives and false negatives.

As might be expected, the performance of MergeAlign improves when the accuracy is measured using the modeller scores. However, better performance can still be obtained under the modeller variants by using the MinRisk method and a non-zero g parameter (*see Table 4.4*). As discussed earlier, the g parameter accomplishes the key aims of the modeller score (i.e. to penalise longer alignments) while maintaining computational tractability, and a meaningful statistical interpretation.

Given the heterogeneity of the different datasets, it is also useful to visualise the results for the individual datasets. As shown in Figure 4.19 and Figure 4.20, the results are consistent across all datasets, with the MinRisk method yielding alignments that are significantly better than the majority of samples, especially as the indel rate is increased. Conversely, the MergeAlign method consistently yields summary alignments that are close to the median accuracy of the sampled alignments, and the S-Coffee method performs consistently worse than the majority of samples.

	Ref 1a (< 25%)			*Ref 1b* (20 – 40%)		
	α_C	α_{C^+}	AMA	α_C	α_{C^+}	AMA
MinRisk (C), $g = 0$	**0.94**	0.77	**0.88**	0.88	0.85	0.82
MinRisk (C), $g = 0.5$	0.90	0.41	0.66	0.92	0.81	0.90
MinRisk (C), $g = 1$	0.88	0.41	0.63	**0.94**	0.83	**0.93**
MinRisk (C^+), $g = 0$	0.67	**0.92**	0.77	0.71	0.87	0.66
MinRisk (C^+), $g = 0.5$	0.86	0.86	**0.88**	0.85	**0.91**	0.89
MinRisk (C^+), $g = 1$	0.88	0.64	0.78	0.90	0.88	**0.93**
MergeAlign	0.91	0.59	0.74	0.80	0.75	0.84
S-Coffee	0.45	0.14	0.26	0.52	0.32	0.52

Table 4.5: Average rank scores for the different methods on BAliBASE datasets, using the accuracy metrics described in the main text and in Table 4.2. Highest values for each column shown in bold.

4.5.4 Results: BAliBASE

For the BAliBASE datasets, the MinRisk method also consistently yields summaries that are better than the majority of samples, and outperforms the other methods examined here in all cases (*see Tables 4.5 and 4.6*). Nevertheless, although still ranking behind most of the MinRisk combinations, MergeAlign performs somewhat better on the BAliBASE datasets than on the simulated data, with ranks scores consistently much higher than the median. This suggests that these particular BAliBASE alignments contain fewer of the types of features (for example large numbers of indels) that are likely to lead to suboptimal solutions under the MergeAlign algorithm. Similarly, the S-Coffee method, although still often worse than the median accuracy of the samples, performs better than on the simulated data, suggesting that the heuristics employed by T-Coffee are tailored more towards aligning these types of datasets. These heuristics may to some extent be overriding the information input via the library, which may explain the poor performance on the simulated datasets.

We can see also that in general the optimal value of g for the MinRisk method is higher for the Ref 1b dataset, reflecting the fact that these sequences are less diverged, and hence likely to contain fewer indels. However, as with the simulated data, a value of $g = 0.5$ gives results that are close to optimal in all scenarios.

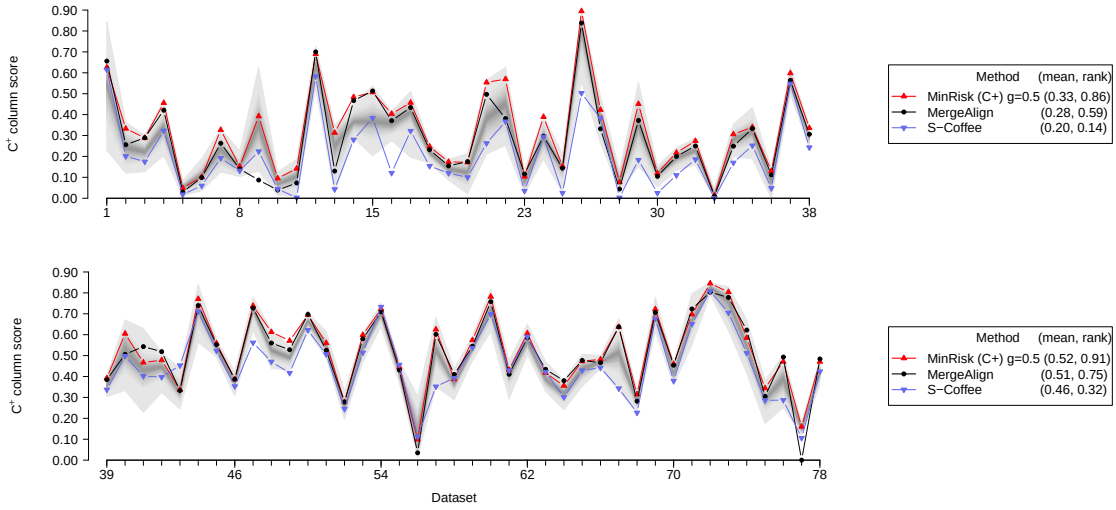


Figure 4.22: Accuracy of summary alignments for BAliBASE alignments under three different methods as measured by α_{C^+}, for low (top panel) and medium (bottom panel) sequence identity. The range of accuracy values covered by the StatAlign samples is shown in grey, with lighter shading indicating greater distance from the median.

	Ref 1a ($< 25\%$)		*Ref 1b* ($20 - 40\%$)	
	α_C^m	$\alpha_{C^+}^m$	α_C^m	$\alpha_{C^+}^m$
MinRisk (C), $g = 0$	0.93	0.74	0.82	0.78
MinRisk (C), $g = 0.5$	**0.95**	0.70	0.96	0.96
MinRisk (C), $g = 1$	0.92	0.68	**0.97**	**0.97**
MinRisk (C^+), $g = 0$	0.40	0.50	0.34	0.33
MinRisk (C^+), $g = 0.5$	0.86	**0.88**	0.83	0.85
MinRisk (C^+), $g = 1$	0.90	0.86	0.93	0.96
MergeAlign	0.93	0.74	0.85	0.86
S-Coffee	0.59	0.46	0.76	0.75

Table 4.6: Average rank scores for the different methods on BAliBASE datasets, measured using the modeller scores. Highest values for each column shown in bold.

	Simulated data			BAliBASE	
	low	*medium*	*high*	*Ref 1a*	*Ref 1b*
$p_C \rightarrow \mathbb{1}(C(X) \in C(A))$	0.93	0.92	0.90	0.99	0.99
$p_{C^+} \rightarrow \mathbb{1}(C^+(X) \in C^+(A))$	0.84	0.78	0.75	0.79	0.89

Table 4.7: Accuracy of marginal probabilities in predicting column presence/absence, as measured by the area under a ROC curve (AUC).

Predictive power of marginals

As well as providing a way to approximate full alignment probabilities, posterior column marginal probabilities can also be good predictors of the presence or absence of a column in the true alignment (Lunter *et al.*, 2008). In all cases examined here, the column marginals are excellent predictors of the presence or absence of the column in the true alignment, with an AUC close to 1, especially for the BAliBASE datasets (*see Table 4.7*). The C^+-weighted marginals (the marginal probability of a column after grouping with all other columns containing the same characters, regardless of position in the alignment) are less accurate in predicting the presence/absence of a column under the C^+ definition, which may be due to the fact that the estimates of p_{C^+} make stronger assumptions about the exchangeability of columns, averaging over a larger set of possible predecessors. In all cases, predictive power is higher for alignments containing fewer indels, although the predictive power of the marginals will depend largely on the suitability of the evolutionary model for analysing the dataset.

4.6 Other applications of the alignment DAG

4.6.1 Combining the output of other alignment programs

The approaches detailed in this Chapter are in theory applicable to a set of alignments generated by any type of method, although the quality of the probability estimates generated

by the DAG will depend on the quality of the underlying model used to generate the alignments. Although this type of method can be used to combine the output of other alignment programs, in a similar fashion to the M-Coffee procedure (Wallace *et al.*, 2006), such an approach does not have a probabilistic interpretation, and will depend heavily on the choice of programs used to generate the input.

We have observed that this type of procedure usually yields summary alignments that are similar in accuracy to the program that typically generates the most accurate alignments (data not shown); however, since this the identity of this program is usually known from the outset, based on benchmarking results, there is not much to be gained by employing such a procedure. Moreover, the reliability of such an approach as a heuristic will depend strongly on the degree of similarity between the different alignment programs, hence we would recommend against such using alignment DAGs as a way of combining the output of non-probabilistic alignment programs.

4.6.2 Alignment DAGs as generators of alignment samples

One other obvious application of the alignment DAG is as a way of generating additional alignment samples, which can be sampled by using a DAG-based version of the stochastic traceback algorithm described in Durbin *et al.* (1999). This simply involves building up an alignment according to the following recursive formula

$$p(A^{(i)} = X \mid A^{(i-1)} = X') = p(X \mid X')z(X')/z(X) \tag{4.36}$$

where $z(X)$ is defined as in equation (4.19), and $p(X \mid X')$ can be derived using either the pair-marginal or mean-field expression, as desired.

One potential use for these alignment samples could be as a source of proposals within an MCMC alignment sampler, allowing for a new state to be efficiently generated, along with a known proposal probability for use in a Metropolis-Hastings accept/reject step. Al-

though this type of approach does not allow for the exploration of previously unobserved columns, it could be useful as an additional MCMC move to improve mixing, particularly once much of the space has already been explored. The data structures needed to efficiently implement such an approach may present some significant challenges, but this is likely to be a fruitful direction for future research.

Chapter 5

Downstream analysis in the presence of alignment uncertainty

Although joint sampling approaches may be analytically tractable for comparison of a small number of sequences (Hamada *et al.*, 2009; Satija *et al.*, 2008; Sinha and He, 2007), the computational complexity involved in analysing these hierarchical joint models typically does not scale well with the number of sequences; even the use of procedures such as MCMC is only able increase the range of tractability to a limited extent (Novák *et al.*, 2008; Satija *et al.*, 2009; Suchard and Redelings, 2006). Moreover, although the inclusion of additional levels of annotation or information in the joint model may help reduce the uncertainty in the inference, approaches such as detailed in Chapters 2 and 3 generally require a new evolutionary model to be formulated, along with carefully designed procedures for conducting inference under the model (Meyer and Miklós, 2007; Satija *et al.*, 2008, 2009). For these reasons, for certain cases of interest the Bayesian coestimation approach may still be considered impractical (Liu *et al.*, 2012; Lunter *et al.*, 2005a).

5.1 Propagating alignment uncertainty into downstream inference

In the previous chapter, we have examined how the DAG-based representation of the set of sampled alignments facilitates the efficient computation of alignment probabilities, as well as generation of accurate summary alignments. These summary alignments can then be used as the input to downstream analyses that are set up to make use of a single alignment. However, given the sensitivity of downstream analysis to the specific choice of alignment, it is also desirable to explore ways in which the entire set of alignments can be used for downstream inference, rather than using only a single summary alignment.

5.1.1 Sequential approach

One way of accomplishing this is to carry out the downstream analyses separately on each of the sampled alignments, averaging or summarising the results as appropriate. This type of *sequential* approach has been used to assess the sensitivity of phylogenetic inference to the starting alignment (Liu *et al.*, 2009, 2012; Wong *et al.*, 2008), as well as examining the effect of alignment uncertainty on estimates of positive selection (Blackburne and Whelan, 2013) and RNA secondary structure prediction (Arunapuram *et al.*, 2013).

However, as discussed earlier, a set of alignment samples will typically contain only a small portion of the total probability mass, even for pairwise alignments with relatively low uncertainty. Hence, the uncertainty quantified in the individual samples will be a significant underestimate of the true alignment uncertainty.

Moreover, since the relative frequencies of whole alignments are a very poor estimator of posterior probabilities, simply carrying out an independent analysis on each sampled alignment and then averaging is likely to yield unreliable results. Reweighting procedures such as those discussed by Blackburne and Whelan (2013) are only feasible when the posterior probability of each alignment can be computed exactly, which is not the case for

many models of interest.

5.1.2 DAG-based approach

In order to address these issues, we can make use of the alignment DAG, making use of intersections between alignments to increasing the effective sample size.

Due to the acyclic structure of the alignment column graph, it is possible to adapt many standard algorithms, such as forward-backward algorithms for HMMs, to operate on the DAG structure rather than an individual alignment. This allows for downstream inference to be averaged over a very large number of alignments, weighted according to a more reliable estimate of the posterior probability for each alignment, rather than analysing only a small collection of individual samples.

5.1.3 Approximate marginalisation over alignments

As a simple example, we can consider the case of tree inference under an independent-sites model. On a single alignment the posterior probability of a tree, Υ, can be written as a product of contributions from each column:

$$p(\Upsilon \mid A) \propto p(\Upsilon) \prod_{i=1}^{L_A} p(A^{(i)} \mid \Upsilon) \tag{5.1}$$

where $p(X \mid \Upsilon)$ is the probability of column X under the independent-sites model given the tree Υ, and the proportionality above involves the quantity $\int p(A, \Upsilon) d\Upsilon$. It is a straightforward extension then to compute the posterior averaged over all alignments in the DAG, using a similar expression to equation (4.19)

$$z_\Upsilon(X) \propto p(X \mid \Upsilon) \sum_{X' \ltimes X} z_\Upsilon(X') p(X \mid X') \tag{5.2}$$

such that the marginal posterior for the tree, Υ, summing over all alignments in a DAG $\mathcal{D}(\mathcal{A})$, can be written as

$$p(\Upsilon \mid \mathcal{D}(\mathcal{A})) \propto p(\Upsilon) \sum_{A \in \mathcal{D}(\mathcal{A})} p(A) p(\Upsilon \mid A) \tag{5.3}$$

$$\propto p(\Upsilon)\, z_{\Upsilon}(X_{\mathcal{A}}^{(T)}) \tag{5.4}$$

It should be noted that the above quantity will in general not be an unbiased estimator of the full marginal likelihood, $\sum_A p(\Upsilon \mid A)$, where the sum runs over all possible alignments. Hence, although the approximate marginal likelihood may be useful when searching for optimal trees, it may be more complicated to incorporate this into MCMC procedures, which typically require unbiased estimators of the likelihood (Andrieu and Roberts, 2009), although in practice the effect of using a biased estimator may be smaller than the Monte Carlo error (Nicholls *et al.*, 2012).

5.2 Sequence annotation in the presence of alignment uncertainty

Site-independent phylogenetic evolutionary models such as mentioned above can be easily extended to include an additional layer accounting for heterogeneity across sites. One way in which this can be achieved is to use an HMM to annotate the columns of the alignment, with each annotation class having its own set of evolutionary parameters (Felsenstein and Churchill, 1996). The simplest such scheme may involve selecting a different substitution rate parameter for each class, which can be used to detect sites that are highly conserved (Siepel *et al.*, 2005). One way of incorporating alignment uncertainty into such schemes is to design joint estimation approaches that simultaneously infer alignments, annotations and model parameters (Satija *et al.*, 2008, 2009). However, such approaches may be highly computationally intensive, limiting the applicability to larger datasets.

Here we will consider instead annotating the set of alignments that are contained within an alignment DAG. As we have seen in Chapter 4, this set is typically very large, and may contain a large proportion of the total posterior mass even with only 1000 samples. The main approximation that is made with such an approach is that the alignment sampling must be carried out before the annotations are known, rather than being co-estimated (Satija *et al.*, 2009).

5.2.1 Annotation of a single alignment with an HMM

We begin by defining a single multiple sequence alignment (MSA) A of length L as a set of columns $\{A^{(1)}, \ldots, A^{(L)}\}$, along with an ordering defined by the $\ltimes$ operator, which denotes concatenation into an alignment. Each character $A_m^{(i)}$ in the column is either a character taken from one of a set of M sequences $s^{(1)}, \ldots, s^{(M)}$, or a gap character, –, and we say that an alignment is *valid* iff $A_m^{(i)} = s_k^{(m)}$ and $A_m^{(j)} = s_l^{(m)}$ implies $k < l$ for all $A^{(i)} \ltimes \ldots \ltimes A^{(j)}$.

We can define a *hidden Markov model* (HMM) on the columns of the alignment, whereby each column $i = 1, \ldots, L$ is annotated with a *hidden state* $h_i \in \{1, \ldots, K\}$, where K is the number of different states. An HMM is usually based on the assumption that the hidden state at a particular column depends only on that of its nearest neighbours, which defines a model of the form shown below.

The HMM is specified by the set of possible hidden states, along with a matrix R of *transition probabilities* between these states, where $R_{kl} = p(h_i = l \mid h_{i-1} = k)$, and a set of

emission probabilities. The emission probabilities define the probability of observing $A^{(i)}$ conditional on its annotation with state k, given a (possibly multidimensional) auxiliary parameter θ.

$$e_k(A^{(i)} \mid \theta) = p(A^{(i)} \mid h_i = k, \theta) \text{ for } i = 1, \ldots, L \tag{5.5}$$

$$e_k(A^{(L+1)} \mid \theta) = 1 \tag{5.6}$$

In our case, where $A^{(i)}$ represents a column of an alignment, the emission probabilities can be defined in terms of a phylogenetic evolutionary model, which gives rise to what has been termed a 'phylo HMM' (Siepel *et al.*, 2005).

5.2.2 Decoding the HMM

Given the above components, the *maximum a posteriori* (MAP) annotation

$$\hat{\boldsymbol{h}}_{MAP}(\theta) = \arg\max_{h} \left\{ e_{h_1}(A^{(1)} \mid \theta) \prod_{i=2}^{L} R_{h_{i-1}h_i} e_{h_i}(A^{(i)} \mid \theta) \right\} \tag{5.7}$$

can be computed using the Viterbi algorithm. However, it is often the case that the MAP annotation is just one of many annotations with similar posterior probability, such that the choice of a single MAP is difficult to justify (Durbin *et al.*, 1999; Yau and Holmes, 2013). In such cases, it is often useful to consider the marginal posterior for a particular annotation at each column of the alignment, which can be derived using the *forward-backward* algorithm. More specifically, using the symbol $\odot$ to denote element-wise vector or matrix multiplication, we recursively define a set of row vectors that we term *forward vectors*

$$\boldsymbol{f}^{(i)}(\theta) = [\boldsymbol{f}^{(i-1)}(\theta)\, R] \odot \boldsymbol{e}(A^{(i)} \mid \theta) \tag{5.8}$$

with $\boldsymbol{f}^{(0)}(\theta) = \pi^{(R)}(eq)$ given by the equilibrium distribution defined by the transition matrix R. We similarly define a set of *backward vectors*, which are column vectors defined as

$$\boldsymbol{b}^{(i)}(\theta) = [R\, \boldsymbol{b}^{(i+1)}(\theta)] \odot \boldsymbol{e}(A^{(i+1)} \mid \theta) \tag{5.9}$$

where $\boldsymbol{b}^{(L+1)} = \mathbf{1}$. The total likelihood of the observations, summed over all annotations, is then given by

$$p(A \mid \theta) = \sum_{\boldsymbol{h}} p(A, \boldsymbol{h} \mid \theta) \tag{5.10}$$

$$= \boldsymbol{f}^{(L)}(\theta)\mathbf{1} \tag{5.11}$$

The marginal posterior probability for h_i, summed over all other annotations of h_{-i}, is then given by

$$p(h_i = k \mid A, \theta) = f_k^{(i)}(\theta) b_k^{(i)}(\theta) / p(A \mid \theta) \tag{5.12}$$

Given the marginal posteriors $p(h_i \mid A, \theta)$, the *maximum posterior decoding* (MPD) of the HMM is given by

$$\hat{\boldsymbol{h}}(A, \theta) = \arg\max_{\boldsymbol{h}} \sum_i p(h_i \mid A, \theta) \tag{5.13}$$

The MPD can be derived as the minimum risk decoding under a loss function of the form

$$L(\hat{\boldsymbol{h}} \parallel \boldsymbol{h}) = \sum_i \lambda_{FP}(1 - \mathbb{1}(\hat{h}_i = h_i)) - \rho_{TP}\mathbb{1}(\hat{h}_i = h_i) \tag{5.14}$$

using similar arguments to those presented in Section 4.3.1. Since the maximum under such a function is invariant to the addition of a constant to each term in the sum, it is effectively equivalent to a loss function of the form

$$L(\hat{\boldsymbol{h}} \parallel \boldsymbol{h}) = \sum_i (\lambda_{FP} - \rho_{TP})\mathbb{1}(\hat{h}_i = h_i) \tag{5.15}$$

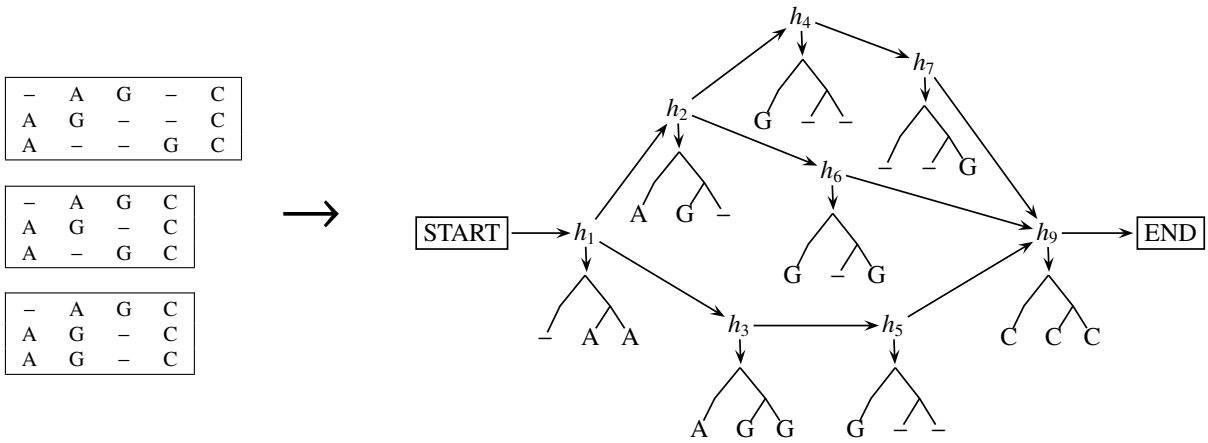


Figure 5.1: Three alignments are combined into a alignment DAG. Associating a hidden state with each column, we can define a generalisation of the phyloHMM on the alignment DAG.

5.2.3 Annotating alignment DAGs

The linear HMM formulation can easily be extended to the case where the dependencies between the elements of $\boldsymbol{h}$ are described by a DAG, as illustrated in Figure 5.1.

As with the single-alignment case, we now consider annotating each column with an unknown hidden state. One option would be to annotate each alignment separately, and then compute a weighted average of the annotations for each column over all N alignments, such that we would have

$$p(h_i = k \mid \mathcal{A}, \theta) = \frac{1}{\sum_{A \in \mathcal{A}} p(A \mid \theta)} \sum_{A \in \mathcal{A}} \mathbb{1}(X^{(i)} \in A) f_k^{(i)}(A, \theta)\, b_k^{(i)}(A, \theta) / p(A \mid \theta) \tag{5.16}$$

where $p(A \mid \theta)$ is as described in equation 5.11, and $\boldsymbol{f}$ and $\boldsymbol{b}$ have been rewritten as functions of each alignment.

However, the above approach does not provide any way to combine the predictions on the individual columns, except by projecting down the annotation probabilities onto a single sequence. Moreover, as discussed in Section 5.1.1, this procedure assumes that the relative frequencies of the whole alignments can be used as a proxy for posterior alignment probability, which is likely to be an unreliable assumption.

Rather than restricting to the set of sampled alignments, it is also possible to average

the annotation over all alignments within the DAG, by annotating each column dependent on its predecessors and successors in the DAG, setting the forward and backward scores for each column to be the sum of scores for all the predecessors and successors, respectively.

$$\boldsymbol{f}^{(X)}(\theta) = \boldsymbol{e}(X \mid \theta) \odot \sum_{X' \ltimes X} \boldsymbol{f}^{(X')}(\theta)\, R \times p(X \mid X') \tag{5.17}$$

$$\boldsymbol{b}^{(X)}(\theta) = \sum_{X \ltimes X'} [R\, \boldsymbol{b}^{(X')}(\theta)] \odot \boldsymbol{e}(X' \mid \theta) \times p(X' \mid X) \tag{5.18}$$

This formulation sums over all the paths in the graph, and hence sums not only the observed alignments, $\mathcal{A}$, but all ways in which the observed columns can be combined into valid alignments, as described in Chapter 4. The conditional probabilities $p(X \mid X')$ cause each path to be weighted according to its posterior probability, and can either be estimated using pair frequencies, or using the mean-field approximation, $p(X \mid E_S(X'))$, as discussed in relation to equation (4.15).

These generalised forward and backward vectors can be used to compute the joint posterior probability of a column being present in the alignment, and being annotated with a particular state, k

$$p(X, h_X = k \mid \mathcal{D}, \theta) = f_k^{(X)}(\mathcal{D}, \theta) b_k^{(X)}(\mathcal{D}, \theta) / p(\mathcal{D} \mid \theta) \tag{5.19}$$

where h_X denotes the annotation of column X, and $p(\mathcal{D} \mid \theta) = \boldsymbol{f}^{(X^{(T)})}(\theta)\mathbf{1}$ can be interpreted as the approximate marginal likelihood of the sequences, summed over all annotations, and summed over all alignments in the DAG $\mathcal{D}$.

For any particular path through the DAG, the MPD annotation, after incorporating information from the entire set of alignments in the DAG, can then be computed using equation (5.13), with $p(h_i \mid A, \theta)$ replaced by $p(X, h_X = k \mid \mathcal{D}, \theta)$. Rather than being based on just a single alignment, this annotation incorporates an average over all possible alternative alignments.

5.2.4 Simultaneous decoding of alignment and HMM

As well as projecting the joint marginals onto a single alignment, as described above, we can also use these to compute the minimum-risk decoding of alignments and annotations satisfying the equation

$$(\hat{A}, \hat{\boldsymbol{h}}) = \arg\max_{A, \boldsymbol{h}} \sum_{A^{(i)} \in A} p(A^{(i)}, h_i) - g \tag{5.20}$$

which corresponds to a loss function of the form

$$\mathcal{L}(\hat{A}, \hat{\boldsymbol{h}} \parallel A, \boldsymbol{h}) = \sum_{i=1}^{L_{\hat{A}}} [\lambda_{FP} - (\rho_{TP} - \lambda_{FP}) \mathbb{1}(\hat{A}^{(i)} \in A) \mathbb{1}(\hat{h}_i = h_i)] \tag{5.21}$$

similar to the case discussed in equation (4.27).

The minimum-risk alignment and annotation pair can be found using the same maximum-weight path algorithm presented in Algorithm 7, except with each column effectively replicated according to the number of states in the HMM.

5.2.5 Parameterising the model

The model parameters, θ, can be estimated from the data according to the marginal posterior probability $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta) p(\theta)$, in a fashion similar to that described in Section 5.1.3. However, in this investigation we consider these parameters to be fixed, in the interests of simplicity.

In the application we will consider below, we consider HMMs with two states, corresponding to sites with low mutation rate (conserved) and high mutation rate (non-conserved), which will be encoded by states 0 and 1 respectively. The general rate matrix for such a

model can be written as

$$R = \begin{pmatrix} 1-\mu & \mu \\ \nu & 1-\nu \end{pmatrix} \tag{5.22}$$

$$\pi^{(R)}(eq) = (\mu, \nu)/(\mu + \nu) \tag{5.23}$$

where μ represents the probability of leaving a conserved region, and ν presents the probability of entering a conserved region. The ratio $\nu/(\mu + \nu)$ yields the expected proportion of conserved sites (the *expected coverage*), and the expected length of a conserved fragment is given by $1/\mu$. This is of the same form as the phylo-HMMs discussed by Siepel *et al.* (2005).

The emission probabilities are derived using a generalised time-reversible substitution model (Tavaré, 1986), and a multiplier, ρ, describing the relationship between the rates in the conserved and non-conserved categories, i.e.

$$Q_0 = \rho Q_1 \tag{5.24}$$

The emission probability for a column given the annotation, $\boldsymbol{h}$, and a tree, Υ, is then given by

$$e_k(X \mid \theta) = p(X \mid Q_{h_X}, \Upsilon) \tag{5.25}$$

where the quantity on the right-hand side can be computed using the standard Felsenstein recursion.

In our test applications, the rate matrix Q_1 is estimated by first computing the minimum-risk alignment for the DAG, according to Algorithm 7, and then running the `phyloFit` tool from the `phastCons` package (Siepel *et al.*, 2005) to generate a maximum-likelihood rate matrix under the GTR model. This procedure implicitly computes an ML annotation for the alignment, and generates two trees, with tree lengths scaled according to the estimated

rates, r_0 and r_1, in the non-conserved and conserved categories respectively. The multiplier, ρ, described above is then estimated by taking the ratio of any one of the branch lengths in the conserved model, versus the unconserved model, which is equal to r_1/r_0. Gaps are treated as an extra nucleotide, with transition rates to/from gap states specified by the user.

5.2.6 Example: annotation of binding sites

The stripe 2 enhancer region for the Drosophila even-skipped (*eve*) gene contains a number of experimentally verified binding sites for at least four transcription factors—the activators bicoid (*bcd*) and hunchback (*hb*), and the repressors giant (*gt*) and Kruppel (*Kr*)—which are conserved across different species (Arnosti *et al.*, 1996; Satija *et al.*, 2008). Previous joint estimation approaches have shown that accounting for alignment uncertainty is able to improve that accuracy of binding site predictions in such regions (Satija *et al.*, 2008, 2009).

As a test case, we used the dataset of Satija *et al.* (2008), corresponding to a region of 486 nucleotides from the *eve* enhancer taken from assembly *dm2* for *D. melanogaster*, along with homologous regions from 9 other Drosophila genomes (from assemblies *droPer1*, *dp4*, *droEre2*, *droYak2*, *droSec1*, *droSim1*, *droGri2*, *droWil1* and *droAna3*), as taken from FlyBase (Tweedie *et al.*, 2009). Known binding sites for *dm2* within this region were taken from the results of Bergman *et al.* (2005). Using StatAlign v1.1, 1000 alignment and tree samples were generated for this region (taken from a total of 5m iterations after a 500,000 burn-in), and a consensus tree computed from the sampled trees. The consensus tree and 100 of the alignment samples (with a thinning interval of 10) were used as input to compute the minimum-risk alignment and annotation.

We explored a range of different parameter values, and Figure 5.2 shows results with $\mu = \nu = 0.033$, corresponding to an expected fragment length of 30, and an expected coverage of 0.5. These values are are similar to the values used by Siepel *et al.* (2005), which were $\mu = 0.0356$, $\nu = 0.0313$. We set the entry rate into the gap state to be close to the av-

	AUC	Rank
$g = 0$	0.773	0.62
$g = 0.5$	0.776	0.68
$g = 1$	0.783	0.87
Median	0.765	0.5

Table 5.1: AUC values and rank scores for predictions using the MinRisk annotation. Shown below is the median AUC for predictions made on each of the 100 samples individually.

erage exit rate into all other nucleotides, which was generally around 0.3. The gap exit rate is set to $0.3 \times \pi^{(Q)}(eq)$, where $\pi^{(Q)}(eq)$ represents the equilibrium distribution corresponding to the non-gap portion of the rate matrix; this corresponds to an overall exit rate from the gap state approximately equal to the entry rate, leading to an equilibrium probability of approximately 0.5 for a gap character, which is appropriate since the alignment contained approximately 50% gaps. The ratio ρ was estimated individually on each alignment sample, with median value 0.202 when computed on the individual samples and 0.183 when computed on the MinRisk alignments.

Predictive accuracy was measured by computing the area under the ROC curve, using the posterior probabilities of conservation for each column, $p(A^{(i)}, h_i = 0)$, as predictors of the presence of a transcription factor binding site. To measure the improvement in performance over taking the samples individually, we made predictions on the individual samples, and computed the proportion of samples whose AUC was below that of the DAG-based method, yielding a rank score. When this rank is equal to 1, the DAG-based prediction is better than all of the single-sample predictions.

As shown in Table 5.1, the DAG-based predictions are typically better than the majority of predictions on individual alignment samples. As the penalty parameter is increased, and hence the number of columns in the alignment decreases (cf. Chapter 4), the performance increases, up to a maximum AUC of 0.783, which is better than 87% of the samples.

In this example, the minimum-risk procedure outlined here also results in improved predictive accuracy when compared to the `phastCons` method, with the latter producing

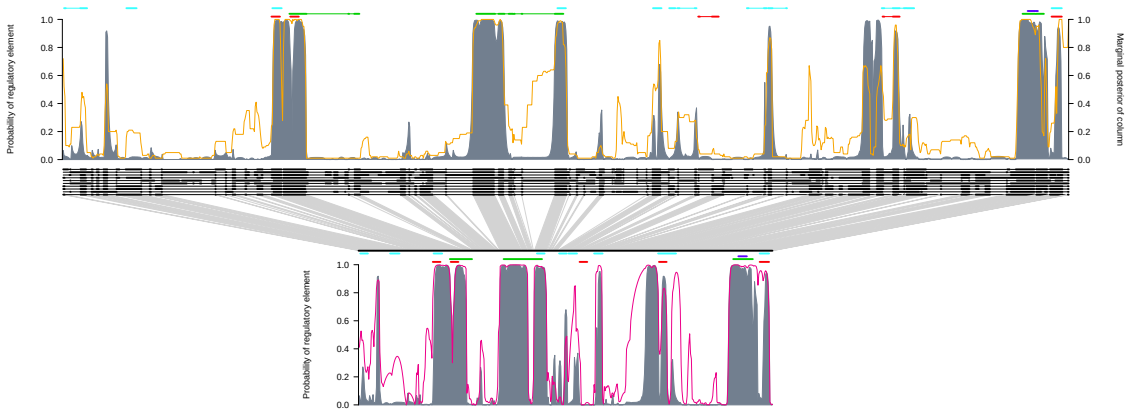


Figure 5.2: Minimum-risk annotation and alignment for a section of the Drosophila genome, projected onto a sequence of interest. The generalised forward-backward algorithm on the alignment DAG is used to compute marginal posteriors for the hidden state at each column. These can then be projected onto the alignment corresponding to the minimum-risk decoding (*joint marginals, $p(X, h_X)$, are annotated in grey, and column marginals, $p(X)$, overlaid in orange*), and onto the reference sequence (*dm2*). Predictions from `phastCons` (Siepel *et al.*, 2005), based on the same alignment, are shown in magenta on the lower panel. The coloured bars above the annotations indicate the locations of known regulatory elements (*bcd* = red, *gt* = green, *hb* = dark blue, *Kr* = cyan), and predictive accuracy for the locations of these elements, as measured by *AUC*, was 0.783 (MinRisk), 0.749 (`phastCons`).

more false positive predictions (*see Figure 5.2 for an example*), and an average AUC of 0.717 over the 100 samples. This may in part be due to the fact that `phastCons` makes use of the Viterbi algorithm to produce a maximum likelihood annotation, rather than selecting the minimum-risk annotation as we have done here. In addition, `phastCons` treats gaps as missing data rather than allowing for a separate gap state, which may lead to statistically inconsistent results (Warnow, 2012).

5.2.7 Conclusions and future work

The key conclusion of this section is that combining a set of alignment samples into a DAG structure, and using the minimum-risk framework outlined here, leads to predictions that are better than the majority of the individual samples. Since the computational cost associated with considering multiple alignments is very small (the typical runtime on a set of 100 alignments is less than one second), this represents a simple and effective way to

improve prediction accuracy.

However, it is also necessary to factor in the time taken to generate the alignment samples in the first place; in our case this step constituted the majority of the runtime, since we made use of StatAlign to jointly sample alignments and trees from the posterior under a statistical model of indels and substitutions. It would be interesting to investigate whether similar improvements could be gained by sampling alignments under a less sophisticated model, for example by keeping the tree fixed in StatAlign. In cases where the phylogeny is well-known from other sources of information, this is likely to significantly reduce runtime as well as reducing alignment uncertainty, which may result in improved predictions with a lower computational cost.

Although we obtain performance that exceeds that of `phastCons`, the scheme we have used here for modelling gap states is certainly a simplistic approach, and improvements to the model would likely increase the predictive power overall. A more comprehensive treatment might allow for each hidden state to have separate rates into and out of gap states, but we leave such approaches for future work.

As discussed above, rather than using the maximum-likelihood parameters for the GTR model derived from `phastCons`, along with user-specified HMM parameters, these parameters could be coestimated with the annotation, which would likely improve the results.

In the analysis of Satija *et al.* (2008), the AUC for `phastCons` on the same genomic region was reported at 0.803, which is higher than the maximum value of 0.752 observed here. This may be due to the fact that an additional binding site (for the transcription factor *sloppy-paired*) was annotated within the region we have considered here, although this is not present in the dataset of Bergman *et al.* (2005). Inclusion of this site in the analysis would likely increase the AUC values, since both our method and `phastCons` typically yield a high posterior probability of a binding site in this region.

5.3 RNA secondary structure prediction in the presence of alignment uncertainty

A similar principle to that discussed in the previous section can be applied to the problem of RNA secondary structure prediction. With the completion of whole-genome sequencing projects, the importance of non-coding RNA molecules (ncRNAs) has become increasingly evident (Washietl *et al.*, 2007), with ncRNAs implicated in a wide range of biological processes. These ncRNAs often adopt very specific structures, which are of key importance for functionality. Given the difficulty associated with experimental structure determination, there has been a large amount of interest in computational methods for predicting structure of ncRNAs from sequence information alone (Eddy, 2004).

Predicting and detecting RNA secondary structure is greatly improved by the use of comparative approaches (Gardner and Giegerich, 2004), which take as input a set of aligned sequences all assumed to have a conserved secondary structure, making use of patterns of conservation and mutation to help identify this common structure. The most effective comparative methodologies are generally thermodynamics-based (Bernhart *et al.*, 2008) or grammar-based approaches (Dowell and Eddy, 2004; Knudsen and Hein, 2003). Here we focus on the grammar based approach, since grammars can be used within a fully probabilistic framework, avoiding the need for heuristic approximations.

Although highly successful in many cases, there can be a high variance in the accuracy of the structure predictions generated by such approaches, part of which may be attributable to sensitivity to the choice of multiple sequence alignment (Anderson *et al.*, 2013). To address the interdependency between alignment and structure prediction, there have been initiatives to co-estimate RNA alignments and secondary structures from a joint statistical alignment model (Meyer and Miklós, 2007). However, sampling from combined models is computationally extremely expensive, limiting the range of applicability of such approaches.

Here we will make use of the alignment DAG structure to allow for secondary structure prediction to be carried out simultaneously on the entire set of alignments contained within the DAG. As well as being more computationally efficient, this approach allows for larger sets of alignments to be considered; as we shall see, this can lead to improved predictive accuracy.

5.3.1 Stochastic context-free grammars

A *grammar* consists of a start symbol, S, a set of terminal symbols, Σ, non-terminal symbols, $\mathbb{V}$, and a set of production rules, $\mathbb{P}$. The production rules describe ways in which non-terminal symbols in the set $\mathbb{V}$ can be replaced by combinations of other terminal or non-terminal symbols; these rules can then be used to generate finite combinations of terminal symbols.

Within the set of grammar-based approaches, the main class of interest is that of *stochastic context-free grammars* (SCFGs), since these allow for the types of long-range base-pair dependencies observed in RNA secondary structures. Following on from initial work by Sakakibara *et al.* (1994), Knudsen and Hein (1999, 2003) devised the grammar that forms the basis of the widely-used Pfold algorithm, which has proven to be one of the most successful SCFGs for RNA secondary structure prediction (Anderson *et al.*, 2012; Dowell and Eddy, 2004).

For context-free grammars (CFGs), the sequence being replaced by a particular rule is restricted to a single variable in $\mathbb{V}$, rather than combinations of variables. Hence, rules of t the form $U \rightarrow VW$ are allowed, but $UV \rightarrow W$ is disallowed. A *stochastic* context-free grammar (SCFG) is then a CFG with probabilities associated with each of the production rules.

5.3.2 SCFGs for RNA secondary structure

We focus on cases where all the production rules fall into one of the following categories:

S	→	$\cdot$ .117	(F) .014	LS .869
L	→	$\cdot$ .895	(F) .105	
F	→		(F) .788	LS .212

Table 5.2: Probabilities for production rules under the grammar of Knudsen and Hein (1999), after transforming to double-emission normal form (Anderson *et al.*, 2012). Here, the set of non-terminal symbols is $\mathbb{V} = \{S, L, F\}$.

1. $U \to \cdot$, where the $\cdot$ represents an unpaired (or uncorrelated) position
2. $U \to (V)$, where the (and) represent two paired positions
3. $U \to VW$ with $V, W \in \mathbb{V}$, corresponding to a bifurcation into two uncorrelated parts

corresponding to the so-called *double-emission normal form* (DENF), as discussed by Anderson *et al.* (2012). Since an RNA secondary structure is specified completely by the set of paired bases, any structure without pseudoknots can be represented in dot-bracket form; the partner for a particular base can be found by parsing the brackets from single pairs of brackets separated only by dots, and moving progressively outwards.

The SCFG which we will use in our analysis is that of Knudsen and Hein (1999). Probabilities for these rules will be taken as in Knudsen and Hein (1999); after converting to the DENF representation, these take the values shown in Table 5.2.

5.3.3 Emission probabilities

For each generated pair (. . .), or unpaired base, $\cdot$, we must also include an associated emission probability, which takes into account the likelihood of the observed sequence, given that it occurs at a paired or unpaired site. While it is possible to make use of evolutionary models for this purpose, as discussed by Knudsen and Hein (1999), in this analysis we

focus on simpler forms for the emission probabilities that treat each observed sequence as an independent sample from the same distribution.

We take the emission probabilities $p(s[i], s[j] \mid U \to (V))$ and $p(s[i] \mid U \to \cdot)$ to be those used in Pfold (Knudsen and Hein, 2003). For a set of N aligned sequences, we then compute the contribution for a column by multiplying the probabilities for each row

$$p(X^{(i)}, X^{(j)} \mid U \to (V)) = \prod_{n=1}^{N} p(X_n^{(i)}, X_n^{(j)} \mid U \to (V)) \tag{5.26}$$

Gaps are treated as a separate character, as described by Knudsen and Hein (2003).

5.3.4 Computing probabilities of structures under a SCFG

Analogously to the forward and backward algorithms used for computing probabilities under an HMM, dynamic programming algorithms exist for computing the probability of a particular sequence (here meaning a sequence of symbols in Σ) summing over all ways in which the sequence can be produced by the grammar (Baker, 1979). In the case of an SCFG, the computations can be carried out via two recursively defined functions, namely the *inside* function, `In`, and the *outside* function, `Out`. The `In` function recursively computes the probability of generating a particular subsequence, summed over all combinations of production rules, and the `Out` function computes the probability of generating all of a sequence excluding a particular subsequence.

As illustrated in Figure 5.3, a subsequence $s[i..j]$ can be generated in one of three ways from a single variable U:

- directly producing $s[i]$ according to a production $U \to \cdot$ of Type 1 (when $i = j$)
- either replacing U according to a production $U \to (V)$ of Type 2 and generating the subsequence $s[i+1..j-1]$ from variable V, or replacing U according to a production $U \to VW$ of Type 3 and generating the subsequence $s[i..k]$ from variable V and the subsequence $s[k+1..j]$ from variable W for some $i \leq k < j$, if $i < j$

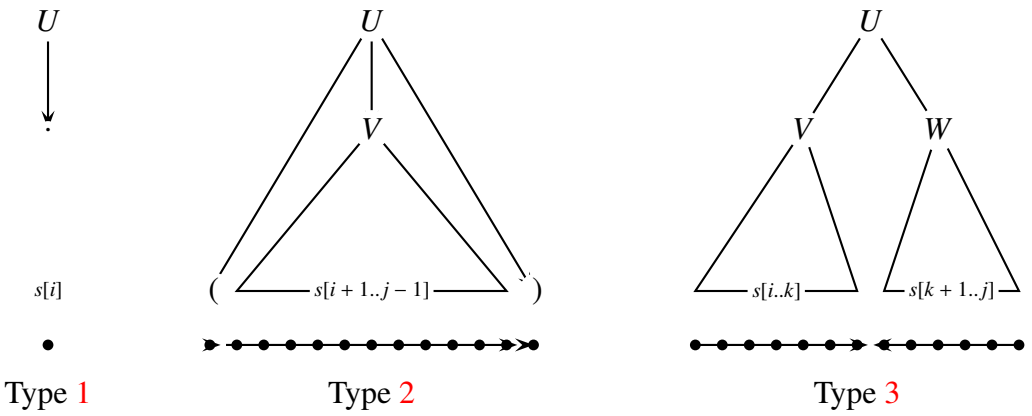


Figure 5.3: The derivations for variables replaced according to each of the three types of productions in DENF. *Figure generated by Rune Lyngsø.*

The inside function for this subsequence can therefore be computed via the following recursion:

$$\mathrm{In}\,(U,i,j) = \sum \begin{cases} \sum_{U\to\,\cdot\,\in\mathbb{P}} p\,(U \to\ \cdot\,)\, p\,(s[i] \mid U \to\ \cdot\,) & \text{if } i = j \\ \sum_{U\to(V)\in\mathbb{P}} p\,(U \to (V))\, p\,(s[i], s[j] \mid U \to (V))\, \mathrm{In}\,(V, i+1, j-1) & \text{if } i \le j-2 \\ \sum_{U\to VW\in\mathbb{P}} p\,(U \to VW) \sum_{i\le k<j} \mathrm{In}\,(V,i,k)\, \mathrm{In}\,(W,k+1,j) & \text{if } i < j \end{cases} \tag{5.27}$$

with the three summation terms corresponding to the Type 1, Type 2 and Type 3 rules shown in Figure 5.3.

Similarly, as shown in Figure 5.4, the portion of a sequence that *excludes* a particular subsequence (i.e. the total probability of generating $s[1..i-1]Us[j+1..|s|]$ under the grammar, starting from the empty sequence, S) can be computed according to the following recursion:

$$\mathrm{Out}\,(U,i,j) = \sum \begin{cases} \sum_{V\to(U)\in\mathbb{P}} p\,(V \to (U))\, p\,(s[i-1], s[j+1] \mid V \to (U))\, \mathrm{Out}\,(V, i-1, j+1) \\ \sum_{V\to UW\in\mathbb{P}} p(V \to UW) \sum_{k>j} \mathrm{Out}\,(V,i,k)\, \mathrm{In}\,(W, j+1, k) \\ \sum_{V\to WU\in\mathbb{P}} p(V \to WU) \sum_{k<i} \mathrm{Out}\,(V,k,j)\, \mathrm{In}\,(W, k, i-1) \end{cases} \tag{5.28}$$

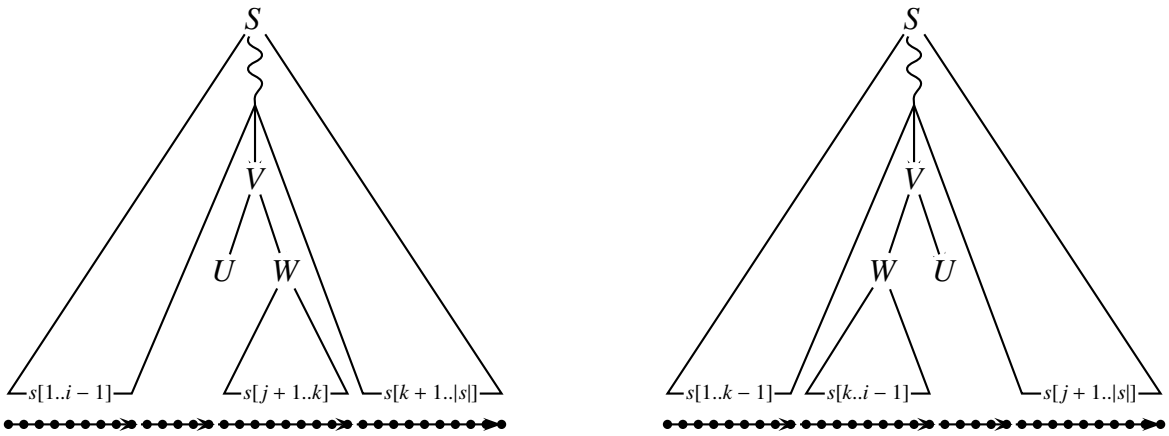


Figure 5.4: The intuition behind the outside algorithm is to progressively fill out the entire sequence, excluding a particular subsequence. The left-hand and right-hand diagrams correspond to the second and third lines of equation (5.28) respectively, i.e. the emissions of Type 3. In each case, the subsequence U is being excluded, and the emission probability computed for the remainder of the sequence. Summing over k and adding on the term for emissions of Type 2 (the first line in equation (5.28)) then yields $Out(U, i, j)$. *Figure generated by Rune Lyngsø.*

with $\text{Out}(S, 1, |s|) = 1$.

5.3.5 Marginal probabilities

Once we have computed inside and outside values, it is possible to compute the marginal probability of a particular unpaired base or pair of bases, summed over all possible productions, by multiplying the relevant inside and outside contributions

$$p(i \text{ unpaired}) \propto \sum_{U \to \cdot \in \mathbb{P}} p(U \to \cdot) \, p(s[i] \mid U \to \cdot) \, \text{Out}(U, i, i) \tag{5.29}$$

$$p(i \text{ paired with } j) \propto \sum_{U \to (V) \in \mathbb{P}} p(U \to (V)) \, p(s[i], s[j] \mid U \to (V)) \times \text{Out}(U, i, j) \, \text{In}(V, i+1, j-1) \tag{5.30}$$

where the proportionality involves dividing by the normalising constant, $p(s) = \text{In}(S, 1, |s|)$, which represents the marginal likelihood of the entire sequence, s, under the SCFG, starting from the empty sequence, S.

5.3.6 Generalising the Inside and Outside algorithms to alignment DAGs

In equation (5.27), the linearity of a sequence is not the crucial feature for the algorithm to be evaluated in polynomial time. The key requirement is to be able to identify the positions that can lie between i and j for recursion on Type 3 productions, and to identify succeeding and preceding positions for recursion on Type 2 and Type 3 productions.

This means that equation (5.27) has a straightforward generalisation to an alignment DAG. We first introduce a labelling of the columns in the DAG, such that

$$X^{(i)} \ltimes X^{(j)} \Rightarrow i < j \;\; \forall X^{(i)}, X^{(j)} \in \Xi \tag{5.31}$$

where Ξ is the set of columns, as introduced in Section 4.1.4, and $X \ltimes X'$ signifies that X is a valid predecessor of X'. Thence, $\mathrm{In}_\Xi(U, i, j)$ represents the sum over probabilities for all subalignments corresponding to paths starting with column $X^{(i)}$ and ending with column $X^{(j)}$, with each path weighted by its probability in the DAG.

Using the shorthand $i \ltimes j$ to denote the relationship $X^{(i)} \ltimes X^{(j)}$, and $p(i \mid j)$ to denote $p(X^{(i)} \mid X^{(j)})$, the generalised inside recursion becomes

$$\mathrm{In}_\Xi(U, i, j) = \sum \begin{cases} \displaystyle\sum_{U \to \cdot \in \mathbb{P}} p\left(X^{(i)} \mid U \to \cdot\right) \text{ if } i = j \\ \displaystyle\sum_{U \to (V) \in \mathbb{P}} \sum_{\substack{k<l \\ i \ltimes k \\ l \ltimes j}} p\left(X^{(i)}, X^{(j)} \mid U \to (V)\right) \mathrm{In}_\Xi(V, k, l)\, p(k \mid i) p(j \mid l) \\ \displaystyle\sum_{U \to VW \in \mathbb{P}} p(U \to VW) \sum_{\substack{i \le k < l < j \\ k \ltimes l}} \mathrm{In}_\Xi(V, i, k)\, \mathrm{In}_\Xi(W, l, j)\, p(l \mid k) \end{cases} \tag{5.32}$$

As illustrated in the example in Figure 5.5, for Type 2 productions it is necessary to consider paths starting at any successor to $X^{(1)}$ and ending at any predecessor of $X^{(8)}$. For Type 3 productions, for any column between $X^{(1)}$ and $X^{(8)}$ (denoted by the free index k in

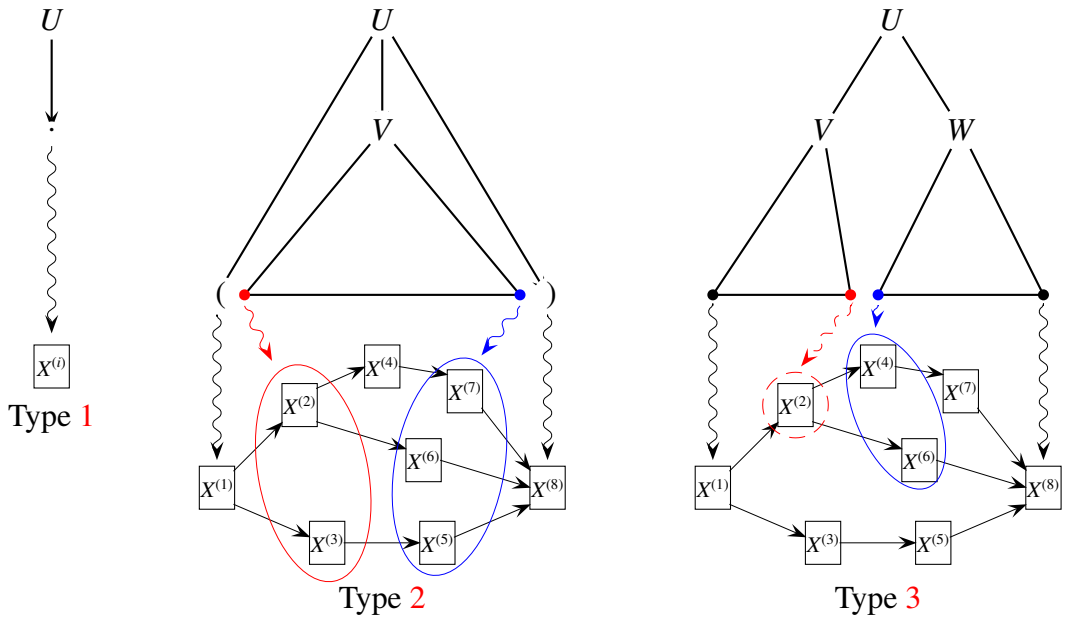


Figure 5.5: The generalisation of the recursion in Figure 5.3 to alignment DAGs. Correspondence between positions in the derivation and columns in the alignment DAG are depicted with $\rightsquigarrow$. *Figure generated by Rune Lyngsø.*

the third line of equation 5.32), we need to consider all successors of this column (denoted by the index l). The dashed line around $X^{(2)}$ indicates that this is just one of the possible columns that can follow $X^{(1)}$, the other being $X^{(3)}$.

The outside algorithm can similarly be generalised to the alignment DAG, yielding

$$\mathrm{Out}_\Xi(U,i,j) = \sum \begin{cases} \displaystyle\sum_{V\to(U)\in\mathbb{P}} \sum_{\substack{k\ltimes i\\ j\ltimes l}} p\left(X^{(k)}, X^{(l)} \mid V \to (U)\right) \mathrm{Out}_\Xi(V,k,l)\, p(i \mid k) p(l \mid j) \\ \displaystyle\sum_{V\to UW\in\mathbb{P}} \sum_{\substack{j\ltimes k\\ l\geq k}} p(V \to UW)\, \mathrm{Out}_\Xi(V,i,l)\, \mathrm{In}_\Xi(W,k,l)\, p(k \mid j) \\ \displaystyle\sum_{V\to WU\in\mathbb{P}} \sum_{\substack{k\ltimes i\\ l\leq k}} p(V \to WU)\, \mathrm{Out}_\Xi(V,l,j)\, \mathrm{In}_\Xi(W,l,k)\, p(i \mid k) \end{cases} \tag{5.33}$$

using the same notation as in equation (5.32).

The marginal probability for a column being present in the alignment and being un-

paired or paired can then be computed in a similar fashion to equation (5.29)

$$p(X^{(i)} \in A \wedge X^{(i)} \text{ unpaired}) \propto \sum_{U \to \cdot \, \cdot \in \mathbb{P}} p\left(U \to \cdot \, \cdot\right) p\left(X^{(i)} \mid U \to \cdot \, \cdot\right) \mathrm{Out}_{\Xi}\left(U, i, i\right) \tag{5.34}$$

$$p(X^{(i)}, X^{(j)} \in A \wedge X^{(i)} \text{ paired with } X^{(j)}) \propto \sum_{U \to (V) \in \mathbb{P}} p\left(U \to (V)\right) p\left(X^{(i)}, X^{(j)} \mid U \to (V)\right) \times \mathrm{Out}_{\Xi}\left(U, i, j\right) \mathrm{In}_{\Xi}\left(V, i+1, j-1\right) \tag{5.35}$$

The normalising constant is now $\mathrm{In}_{\Xi}(S, 0, T)$, which is equal to the marginal likelihood of the sequences summed over all structures, production rules, and all alignments contained within the DAG.

5.3.7 Computational complexity

Recursion (5.27) converts to the inside algorithm by computing the recursive entities in order of increasing distance between i and j, or similar pre-specified orders that ensures that when computing a recursive entity all entities it depends on have already been computed. This requires time $\mathrm{O}\left(|\mathbb{P}| \cdot |s|^3\right)$ and space $\mathrm{O}\left(|\mathbb{V}| \cdot |s|^2\right)$. The evaluation of the `Out` function has the same complexity.

For a DAG with $|\Xi|$ columns, and $\bar{d}|\Xi|$ edges (where $\bar{d}$ is the average equivalence class size, equal to the average in-degree), the generalised inside recursion can be solved for $\mathrm{In}_{\Xi}(S, 0, T)$ in time $O\left(\bar{d}|\Xi|^3 \, |\mathbb{P}|\right)$ and space $\mathrm{O}\left(|\Xi|^2 \, |\mathbb{V}|\right)$. Under a mean-field approximation to the conditional column probabilities (cf. Section 4.2.1), the complexity can be reduced by a factor of $\bar{d}$, the average equivalence class size, using techniques similar to those discussed in Chapter 4, which also relates to the scheme discussed by Jagota *et al.* (2001).

The cubic time complexity (and square space complexity) of the algorithm is usually

not too significant an issue when dealing with single sequences or individual alignments. However, the alignment DAG may contain many thousands of of columns, such that the complexity can grow very quickly. One approach to reduce computational resource requirements for the generalised inside algorithm is to observe that we do not need to compute $\mathrm{In}_{\Xi}(\ldots, i, j)$ for all $i < j$, but only for the cases where $X^{(i)}$ and $X^{(j)}$ can occur in the same alignment. For example, in the Type 2 production in Fig. 5.5 there is no need to consider the combination of $X^{(2)}$ and $X^{(5)}$, or the combination of $X^{(3)}$ and any of $X^{(6)}$ and $X^{(7)}$.

This observation can be formalised by introducing the notation $i \twoheadrightarrow j$ to denote the *transitive, reflexive closure* of the relation $\ltimes$, such that

$$i \twoheadrightarrow j \Rightarrow (i = j) \vee (\exists k_1, \ldots, k_n \cdot i \ltimes k_1 \ltimes \ldots \ltimes k_n \ltimes j) \tag{5.36}$$

We can then set $\mathrm{In}_{\Xi}(V, i, j) = 0$ for all i, j unless $i \twoheadrightarrow j$. Trivially, $i > j \Rightarrow \neg(i \twoheadrightarrow j)$, and $\neg(i \twoheadrightarrow j)$ if $X^{(i)}$ and $X^{(j)}$ are in the same f_P- or f_S-equivalence class, as defined in Section 4.1.3.

5.3.8 Minimum-risk decodings

As with the HMM example considered in Section 5.2, we can compute the decoding (in this case the secondary structure) that maximises expected accuracy, or equivalently minimises risk, by using efficient dynamic programming algorithms.

For a loss function defined in terms of the number of correctly predicted paired and unpaired columns, the minimum-risk decoding can be found by analogy to the scheme used by Knudsen and Hein (2003) for a single alignment, by filling out the following dynamic

programming matrix

$$E(i,j) = \begin{cases} 0 \text{ if } \neg(i \twoheadrightarrow j) \\ \max \begin{cases} \gamma_1 p(X^{(i)} \text{ unpaired}) + \max\limits_{i \ltimes k} E(k,j) \\ \gamma_2 p(X^{(i)} \text{ paired with } X^{(j)}) + \max\limits_{\substack{i \ltimes k \\ l \ltimes j}} E(k,l) \\ \max\limits_{i \twoheadrightarrow k \,\ltimes\, l \twoheadrightarrow j} E(i,k) + E(l,j) \end{cases} \end{cases} \tag{5.37}$$

The minimum-risk structure can then be found by tracing back from $E(0,T)$, following the choice that gave the maximum score at each step. As before, the above procedure can be made more efficient by caching the innermost maxima for each equivalence class. In the following analysis we use $\gamma_1 = 0.5$ and $\gamma_2 = 1$, which attaches equal weighting to correctly predicted pairs, and correctly predicted unpaired bases.

5.3.9 Evaluation data

In order to test the above framework, we created a RNA dataset based on the the Rfam database (Griffiths-Jones *et al.*, 2005). Alignments of homologous RNA sequences with known consensus secondary structure were extracted from Rfam seed alignments, and filtered to keep only sequences with fewer than 1000 nucleotides. From this set, 44 RNAs were randomly selected, and StatAlign v3.0 was used to generate $10,000$ alignment samples for each RNA, from a total of 5m iterations, after a burn-in of 1m. From these alignments, subsets of 5, 10 and 20 alignments were produced for each RNA by subsampling at intervals of 2000, 1000 and 500 respectively, and these were used as input to the DAG-based predictive framework. Predictions were also carried out on 100 individual alignment samples for each RNA, subsampled at intervals of 100 from the initial $10,000$.

For the curated dataset discussed above, we know the true secondary structure, such that it is possible to measure the predictive accuracy using various types of measures. We

explored the following three commonly-used measures of accuracy:

$$\text{sensitivity} = TP/(TP + FN) \tag{5.38}$$

$$\text{PPV} = TP/(TP + FP) \tag{5.39}$$

$$\text{F-score} = 2TP/(2TP + FN + FP) \tag{5.40}$$

where PPV stands for *positive predictive value*. Here, true positives are defined as base-pairs correctly predicted, false positives as predicting a base-pair between two bases that should not be paired, and false negatives as base pairs not predicted.

In order to compute the above statistics it is necessary to have a bijection between positions in the predicted alignment and the true alignment. In order to achieve this, the predicted structure was projected onto each of the sequences in the alignment individually, and the PPV, sensitivity, and F-score calculated for each such projection. Results were then averaged over all the projections.

5.3.10 Results

As can be seen in Figure 5.6, the DAG method is almost never worse than the median prediction over the 100 individual alignment samples, and is often as good as or better than the majority of the samples. We can measure this more rigorously by computing a rank score for each dataset, indicating the number of samples lying below the DAG-based predictions. As shown in Figure 5.7, the rank scores typically lie close to 1, indicating that the DAG-based predictions are better than more than 90% of the predictions on individual samples. We can also observe a small improvement as the number of samples is increased from 5 to 10 to 20.

Averaging the three performance measures over all 44 datasets, there is a significant improvement under all three measures when moving from a single alignment to a 5-alignment DAG. We also see an improvement in all cases as the number of samples is increased, which

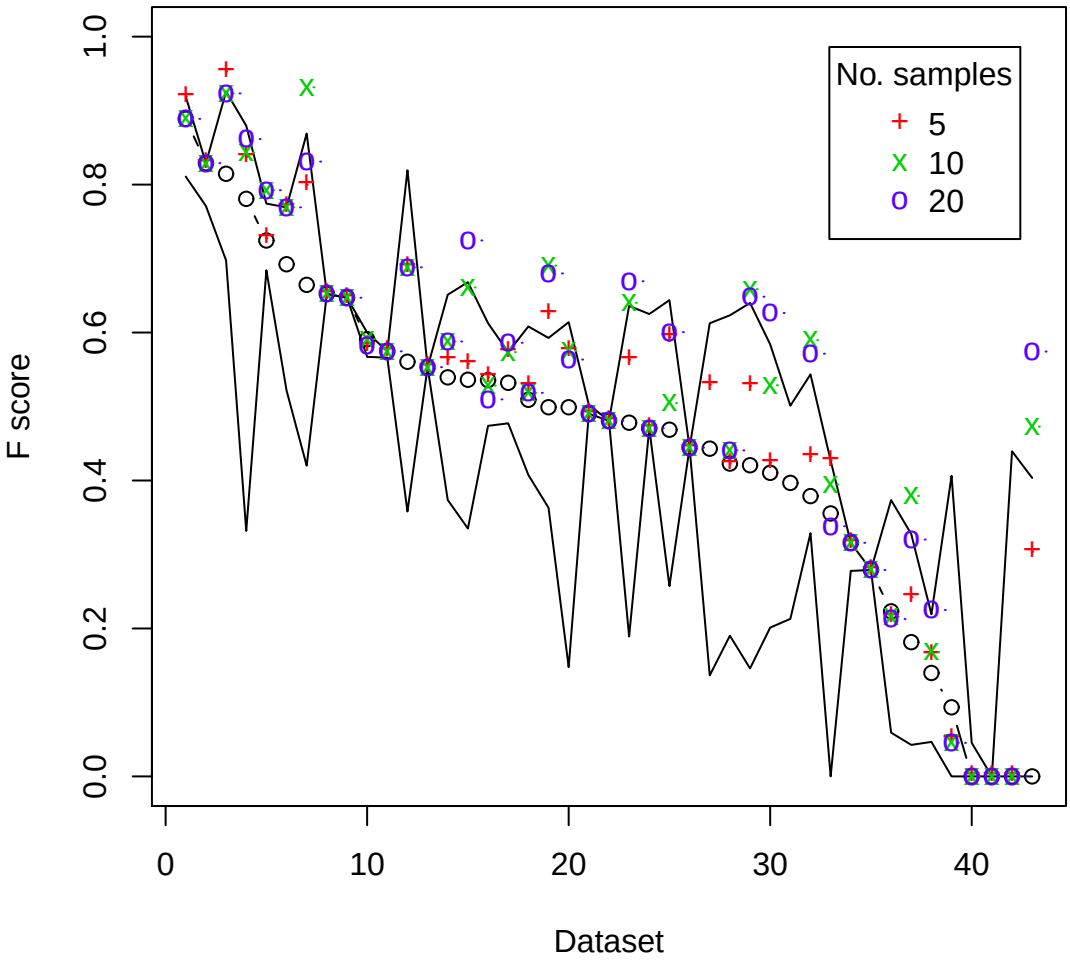


Figure 5.6: Prediction accuracy on test datasets. Distribution of F-scores across all datasets, sorted by median, with medians over the individual samples shown as black circles, and black lines above and below showing the 5% and 95% points in the F-score distribution for each dataset. Superimposed are coloured symbols denoting the accuracy as produced by the DAG method, labelled according to how many alignment samples were used to generate the DAG. Very similar patterns were observed with the sensitivity and PPV measures (data not shown).

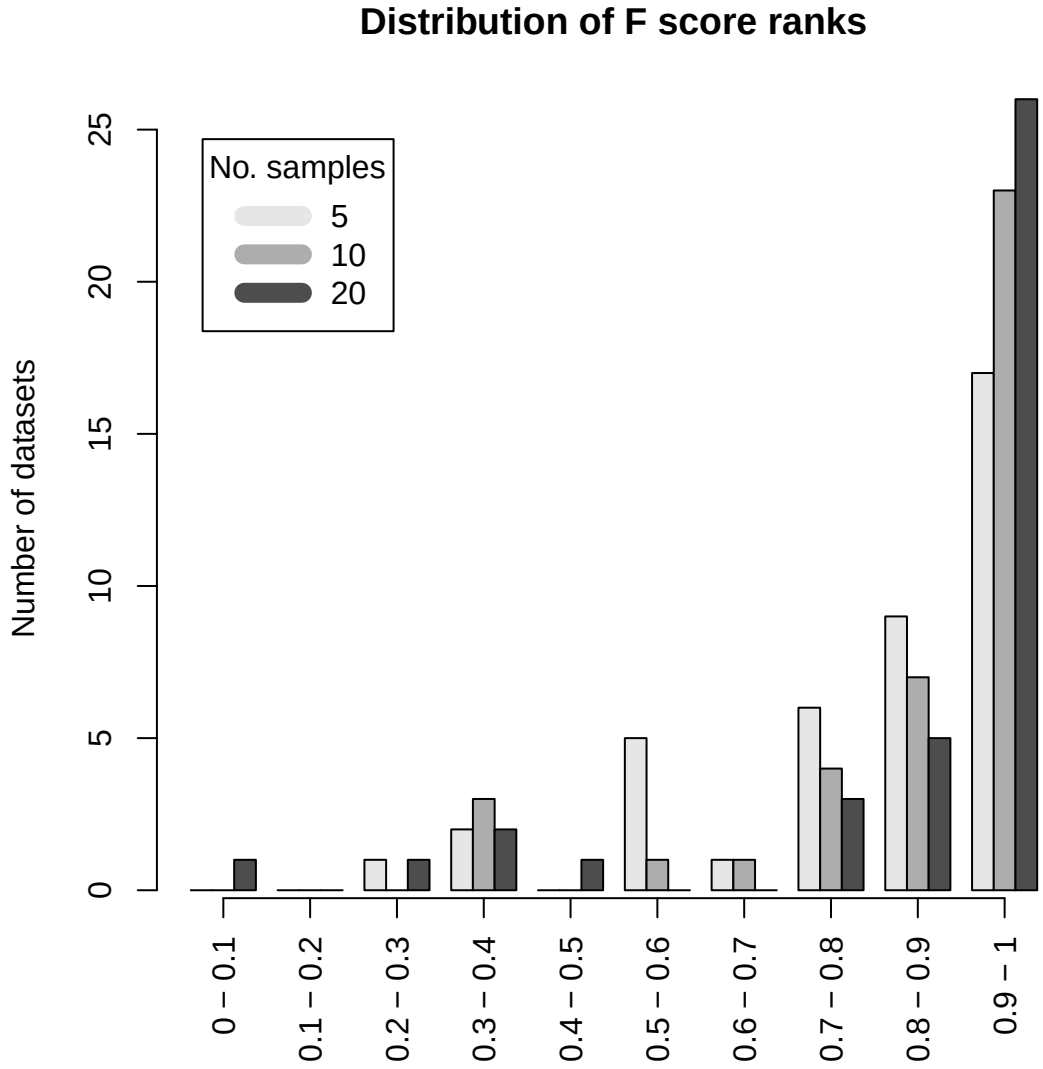


Figure 5.7: Distribution of F-score ranks for DAGs of varying sizes. For the majority of datasets, the DAG-based predictions are more accurate than at least 90% of the predictions on individual alignment samples. Increasing the number of samples used to create the DAG results in a further small improvement.

# alignments	1	5	10	20
F-score	0.486	0.535	0.562	0.566
Sensitivity	0.556	0.616	0.645	0.653
PPV	0.448	0.489	0.512	0.515

Table 5.3: Overall performance across the datasets.

Dataset	# alignments	# unique columns	eq. class size	# paths in DAG	Sensitivity
14	1	82	1	1	0.623
	5	210	1.16	181,818	0.782
	10	318	1.27	311,109,120	0.916
16	1	133	1	1	0.432
	5	216	1.09	12,480	0.457
	10	294	1.11	102,672	0.657

Table 5.4: Improvement in prediction sensitivity as a function of the number of samples, for two specific example datasets. The increase in sensitivity in this case is somewhat larger for Dataset 14, for which the size of the DAG increases much faster as a function of the number of samples.

is likely due to a combination of the increased number of paths in the DAG, and the improved estimates of the column probabilities resulting from the larger sample size.

Examining some specific examples (cf. Table 5.4), we can see that the specificity of the predictions can increase quite sharply as the number of samples is increased. In the two examples shown, the degree of improvement observed appears to be related to the number of additional paths introduced into the DAG with the addition of further samples: in the case where adding further samples does not significantly modify the DAG, the additional samples will make less difference to the predictions. As can also be seen in Table 5.4, the number of paths through the DAG increases very rapidly for some of these examples, for example exceeding 300m with just 10 alignments for Dataset 14. This results in a greatly magnified search space within which to conduct inference, which is part of the reason why the DAG-based results typically perform much better than the predictions on individual samples.

As discussed earlier, the theoretical runtime for the algorithm is cubic in the number of edges in the DAG, although savings can be obtained, for example by considering only pairs

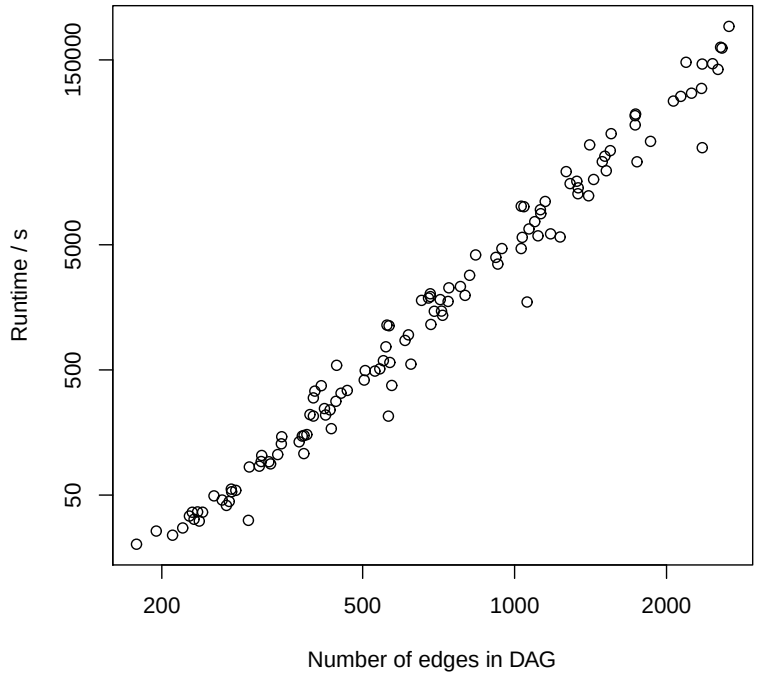


Figure 5.8: Runtime as a function of the number of edges in the DAG. The strong linear relationship on the log-log scale indicates a power law, and a linear model yields an exponent of 3.5.

of columns for which $i \twoheadrightarrow j$. The log-log plot shown Figure 5.8 indicates that the runtime scales with the number of edges in the DAG according to a power law with exponent 3.5. This is slightly above the theoretical value, with the deviation likely due to various overheads associated with maintaining data structures. Although the underlying complexity cannot be improved beyond the basic cubic relationship, the range of applicability of the method could be extended significantly by cutting runtime by a constant factor, hence it would be of interest to investigate ways of improving the efficiency of the software.

In terms of memory requirements, most of the datasets used here ran successfully with under 1Gb of RAM, with some requiring up to 2Gb. As the number of alignment samples is increased up to 50 or 100, the number of columns increases up to several thousand for some of the datasets, leading to memory requirements that exceed 4Gb. It should be stressed,

however, that the current implementation has not been optimised, hence there are likely to be a number of gains in efficiency that could be made beyond this initial implementation.

5.4 Conclusions

In this Chapter we have analysed some common algorithms for multiple sequence analysis which can be modified to operate on the alignment DAG structure; the DAG-based minimum-risk/maximum accuracy predictions are generally superior to the majority of the estimates made on individual alignment samples, often without significant additional computational cost. Indeed, in the case of the HMM analysis, the algorithms on the DAG are no less efficient than repeating the the analysis separately on each of the individual alignment samples, so there is no computational reason not to adopt the type of approach we have presented here.

A similar approach could potentially be used with many other types of algorithms that take as input an alignment and compute a score that is additive over columns or pairs of columns. In conjunction with efficient techniques for generating statistical alignment samples, this type of procedure offers a tractable way to take account of alignment uncertainty in downstream analysis.

The applicability of this type of approach clearly depends on the extent to which alignment uncertainty can be accurately quantified independently of the downstream quantity of interest; as we have discussed, parameters such as trees should ideally be coestimated with alignments, due to their interdependence. However, when joint sampling is not feasible, the two-step methodology described here offers a more robust approach than conducting analysis on a single alignment.

References

Ahola, V., Aittokallio, T., Vihinen, M., and Uusipaikka, E. (2008). Model-based prediction of sequence alignment quality. *Bioinformatics*, **24**(19), 2165–2171.

Ajawatanawong, P., Atkinson, G. C., Watson-Haigh, N. S., MacKenzie, B., and Baldauf, S. L. (2012). SeqFIRE: A web application for automated extraction of indel regions and conserved blocks from protein multiple sequence alignments. *Nucleic Acids Research*, **40**(W1), W340–W347.

Altschuh, D., Vernet, T., Berti, P., Moras, D., and Nagai, K. (1988). Coordinated amino acid changes in homologous protein families. *Protein Engineering*, **2**(3), 193–199.

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

Anderson, J. W. J., Novák, A., Sükösd, Z., Golden, M., Arunapuram, P., Edvardsson, I., and Hein, J. (2013). Quantifying variances in comparative RNA secondary structure prediction. *BMC Bioinformatics*, **14**(1), 149.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient monte carlo computations. *Annals of Statististics*, **37**(2), 697–725.

Aris-Brosou, S. and Yang, Z. (2002). Effects of models of rate evolution on estimation of

divergence dates with special reference to the metazoan 18S ribosomal RNA phylogeny. *Systematic Biology*, **51**(5), 703–714.

Arnosti, D., Barolo, S., Levine, M., and Small, S. (1996). The eve stripe 2 enhancer employs multiple modes of transcriptional synergy. *Development*, **122**(1), 205–214.

Arunapuram, P., Edvardsson, I., Golden, M., Anderson, J. W. J., Novák, A., Sükösd, Z., and Hein, J. (2013). StatAlign 2.0: combining statistical alignment with RNA secondary structure prediction. *Bioinformatics*, **29**(5), 654–655.

Baker, J. K. (1979). Trainable grammars for speech recognition. *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550.

Bergman, C. M., Carlson, J. W., and Celniker, S. E. (2005). Drosophila DNase I footprint database: A systematic genome annotation of transcription factor binding sites in the fruitfly, Drosophila melanogaster. *Bioinformatics*, **21**(8), 1747–1749.

Bernhart, S. H., Hofacker, I. L., Will, S., Gruber, A. R., and Stadler, P. F. (2008). RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, **9**, 474.

Blackburne, B. P. and Whelan, S. (2013). Class of multiple sequence alignment algorithm affects genomic analysis. *Molecular Biology and Evolution*, **30**(3), 642–653.

Bloom, J. D., Drummond, D. A., Arnold, F. H., and Wilke, C. O. (2006). Structural determinants of the rate of protein evolution in yeast. *Molecular Biology and Evolution*, **23**(9), 1751–1761.

Blundell, T. L., Sibanda, B. L., Sternberg, M. J. E., and Thornton, J. M. (1987). Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, **326**(6111), 347–352.

Bouchard-Côté, A. and Jordan, M. I. (2013). Evolutionary inference via the Poisson Indel Process. *Proceedings of the National Academy of Sciences*, **110**(4), 1160–1166.

Bradley, R. K., Roberts, A., Smoot, M., Juvekar, S., Do, J., Dewey, C., Holmes, I., and Pachter, L. (2009). Fast statistical alignment. *PLoS Computational Biology*, **5**(5), e1000392.

Bucka-Lassen, K., Caprani, O., and Hein, J. (1999). Combining many multiple alignments in one improved alignment. *Bioinformatics*, **15**(2), 122–130.

Bujnicki, J. M. (2000). Phylogeny of the restriction endonuclease-like superfamily inferred from comparison of protein structures. *Journal of Molecular Evolution*, **50**(1), 39–44.

Burmester, T., Weich, B., Reinhardt, S., and Hankeln, T. (2000). A vertebrate globin expressed in the brain. *Nature*, **407**(6803), 520–523.

Burmester, T., Ebner, B., Weich, B., and Hankeln, T. (2002). Cytoglobin: A novel globin type ubiquitously expressed invertebrate tissues. *Molecular Biology and Evolution*, **19**(4), 416–421.

Capella-Gutiérrez, S. and Gabaldón, T. (2013). Measuring guide-tree dependency of inferred gaps in progressive aligners. *Bioinformatics*, **29**(8), 1011–1017.

Capella-Gutiérrez, S., Silla-Martínez, J. M., and Gabaldón, T. (2009). trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. *Bioinformatics*, **25**(15), 1972–1973.

Cartwright, R. A. (2005). DNA assembly with gaps (DAWG): Simulating sequence evolution. *Bioinformatics*.

Carvalho, L. E. and Lawrence, C. E. (2008). Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proceedings of the National Academy of Sciences*, **105**(9), 3209–3214.

Castresana, J. (2000). Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Molecular Biology and Evolution*, **17**(4), 540–552.

Challis, C. J. (2013). *Bayesian Structural Phylogenetics*. Ph.D. thesis, Duke University.

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.

Chivian, D. and Baker, D. (2006). Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection. *Nucleic Acids Research*, **34**(17), e112–e112.

Choi, S. C., Hobolth, A., Robinson, D. M., Kishino, H., and Thorne, J. L. (2007). Quantifying the impact of protein tertiary structure on molecular evolution. *Molecular Biology and Evolution*, **24**(8), 1769–1782.

Chothia, C. and Lesk, A. M. (1986). The relationship between the divergence of sequence and structure in proteins. *EMBO Journal*, **5**(4), 823–826.

Churchill, G. A. (1997). Monte Carlo sequence alignment. In *Proceedings of the First Annual International Conference on Computational Molecular Biology*, pages 93–97. ACM.

Collingridge, P. and Kelly, S. (2012). MergeAlign: Improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments. *BMC Bioinformatics*, **13**(1), 117.

Cowell, R., Dawid, P., Lauritzen, S., and Spiegelhalter, D. (2007). *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Information Science and Statistics. Springer, New York.

Cruickshank, D. W. J. (1960). The required precision of intensity measurements for single-crystal analysis. *Acta Crystallographica*, **13**(10), 774–777.

Cruickshank, D. W. J. (1999). Remarks about protein structure precision. *Acta Crystallographica Section D*, **55**(3), 583–601.

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, **5**(suppl 3), 345–351.

DeBlasio, D., Wheeler, T., and Kececioglu, J. (2012). Estimating the accuracy of multiple alignments and its use in parameter advising. In B. Chor, editor, *Research in Computational Molecular Biology*, volume 7262 of *Lecture Notes in Computer Science*, pages 45–59. Springer Berlin Heidelberg.

DePristo, M. A., de Bakker, P. I., and Blundell, T. L. (2004). Heterogeneity and inaccuracy in protein structures solved by X-Ray crystallography. *Structure*, **12**(5), 831 – 838.

Dessimoz, C. and Gil, M. (2010). Phylogenetic assessment of alignments reveals neglected tree signal in gaps. *Genome Biology*, **11**(4), 1–9.

Dickson, R. J. and Gloor, G. B. (2012). Protein sequence alignment analysis by local covariation: Coevolution statistics detect benchmark alignment errors. *PLoS ONE*, **7**(6), e37645.

Dickson, R. J., Wahl, L. M., Fernandes, A. D., and Gloor, G. B. (2010). Identifying and seeing beyond multiple sequence alignment errors using intra-molecular protein covariation. *PLoS ONE*, **5**(6), e11082.

Diebolt, J. and Robert, C. P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, **56**(2), 363–375.

Do, C. B., Mahabhashyam, M. S. P., Brudno, M., and Batzoglou, S. (2005). Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*, **15**(2), 330–340.

Dowell, R. and Eddy, S. (2004). Evaluation of several lightweight stochastic context-free grammars for rna secondary structure prediction. *BMC Bioinformatics*, **5**(1), 71.

Dress, A., Flamm, C., Fritzsch, G., Grunewald, S., Kruspe, M., Prohaska, S., and Stadler, P. (2008). Noisy: Identification of problematic columns in multiple sequence alignments. *Algorithms for Molecular Biology*, **3**(1), 7.

Drummond, A. J., Nicholls, G. K., Rodrigo, A. G., and Solomon, W. (2002). Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics*, **161**(3), 1307–1320.

Dryden, I. L., Hirst, J. D., and Melville, J. L. (2007). Statistical analysis of unlabeled point sets: Comparing molecules in chemoinformatics. *Biometrics*, **63**(1), 237–251.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1999). *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Dutheil, J. Y., Galtier, N., Romiguier, J., Douzery, E. J., Ranwez, V., and Boussau, B. (2012). Efficient selection of branch-specific models of sequence evolution. *Molecular Biology and Evolution*, **29**(7), 1861–1874.

Dwivedi, B. and Gadagkar, S. (2009). Phylogenetic inference under varying proportions of indel-induced alignment gaps. *BMC Evolutionary Biology*, **9**(1), 211.

Ebner, B., Panopoulou, G., Vinogradov, S., Kiger, L., Marden, M., Burmester, T., and Hankeln, T. (2010). The globin gene family of the cephalochordate amphioxus: implications for chordate globin evolution. *BMC Evolutionary Biology*, **10**(1), 370.

Eddy, S. R. (2004). How do RNA folding algorithms work? *Nature Biotechnology*, **22**, 1457–1458.

Edgar, R. C. (2004). Muscle: A multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**(1), 113.

Eidhammer, I., Jonassen, I., and Taylor, W. R. (2000). Structure comparison and structure patterns. *Journal of Computational Biology*, **7**, 685–716.

England, J. L. and Shakhnovich, E. I. (2003). Structural determinant of protein designability. *Physical Review Letters*, **90**, 218101.

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

Felsenstein, J. and Churchill, G. A. (1996). A Hidden Markov Model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, **13**(1), 93–104.

Feng, D. F. and Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, **25**(4), 351–360.

Fletcher, W. and Yang, Z. (2010). The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection. *Molecular Biology and Evolution*, **27**(10), 2257–2267.

Garau, G., Di Guilmi, A. M., and Hall, B. G. (2005). Structure-based phylogeny of the metallo-β-lactamases. *Antimicrobial Agents and Chemotherapy*, **49**(7), 2778–2784.

Gardner, P. P. and Giegerich, R. (2004). A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, **30**(5), 140.

Gardner, P. P., Wilm, A., and Washietl, S. (2005). A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Research*, **33**(8), 2433–2439.

Garrocho-Villegas, V., Gopalasubramaniam, S. K., and Arredondo-Peter, R. (2007). Plant hemoglobins: What we know six decades after their discovery. *Gene*, **398**(1–2), 78 – 85.

Gatesy, J., DeSalle, R., and Wheeler, W. (1993). Alignment-ambiguous nucleotide sites and the exclusion of systematic data. *Molecular Phylogenetics and Evolution*, **2**(2), 152–157.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC.

Godzik, A. (1996). The structural alignment between two proteins: is there a unique answer? *Protein Science*, **5**(7), 1325–1338.

Goodall, C. R. and Mardia, K. V. (1993). Multivariate aspects of shape theory. *The Annals of Statistics*, **21**(2), 848–866.

Gopalasubramaniam, S. K., Kovacs, F., Violante-Mota, F., Twigg, P., Arredondo-Peter, R., and Sarath, G. (2008). Cloning and characterization of a caesalpinoid (chamaecrista fasciculata) hemoglobin: The structural transition from a nonsymbiotic hemoglobin to a leghemoglobin. *Proteins: Structure, Function, and Bioinformatics*, **72**(1), 252–260.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, **162**(3), 705–708.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**(4), 711–732.

Green, P. J. and Mardia, K. V. (2006). Bayesian alignment using hierarchical models, with applications in protein bioinformatics. *Biometrika*, **93**(2), 235–254.

Green, P. J., Mardia, K. V., Nyirongo, V. B., and Ruffieux, Y. (2010a). *Bayesian modelling for matching and alignment of biomolecules*, pages 27–50. The Oxford Handbook of Applied Bayesian Analysis. Oxford University Press, Oxford.

Green, P. J., Mardia, K. V., Nyirongo, V. B., and Ruffieux, Y. (2010b). *Bayesian modelling for matching and alignment of biomolecules*, pages 27–50. The Oxford Handbook of Applied Bayesian Analysis. Oxford University Press, Oxford.

Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R., and Bateman, A. (2005). Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Research*, **33**(suppl 1), D121–D124.

Grishin, N. V. (1997). Estimation of evolutionary distances from protein spatial structures. *Journal of Molecular Evolution*, **45**, 359–369.

Groussin, M., Boussau, B., and Gouy, M. (2013). A branch-heterogeneous model of protein evolution for efficient inference of ancestral sequences. *Systematic Biology*, **62**(4), 523–538.

Gunčar, G., Podobnik, M., Pungerčar, Jože, Štrukelj, B., Turk, V., and Turk, D. (1998). Crystal structure of porcine cathepsin H determined at 2.1Å resolution: location of the mini-chain C-terminal carboxyl group defines cathepsin H aminopeptidase function. *Structure*, **6**(1), 51 – 61.

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

Hall, B. G. (2008). How well does the HoT score reflect sequence alignment accuracy? *Molecular Biology and Evolution*, **25**(8), 1576–1580.

Hamada, M. and Asai, K. (2012). A classification of bioinformatics algorithms from the viewpoint of maximizing expected accuracy (MEA). *Journal of Computational Biology*, **19**(5), 532–549.

Hamada, M., Sato, K., Kiryu, H., Mituyama, T., and Asai, K. (2009). CentroidAlign: fast and accurate aligner for structured RNAs by maximizing expected sum-of-pairs score. *Bioinformatics*, **25**(24), 3236–3243.

Hamada, M., Kiryu, H., Iwasaki, W., and Asai, K. (2011). Generalized centroid estimators in bioinformatics. *PLoS ONE*, **6**(2), e16450.

Hansen, T. F. and Martins, E. P. (1996). Translating between microevolutionary process and macroevolutionary patterns: The correlation structure of interspecific data. *Evolution*, **50**(4), 1404–1417.

Hasegawa, H. and Holm, L. (2009). Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology*, **19**(3), 341–8.

Hastie, D. I. and Green, P. J. (2012). Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, **66**(3), 309–338.

Hein, J., Wiuf, C., Knudsen, B., Møller, M. B., and Wibling, G. (2000). Statistical alignment: computational properties, homology testing and goodness-of-fit. *Journal of Molecular Biology*, **302**(1), 265–279.

Herman, J. L., Novák, A., Lyngsø, R., Szabó, A., Miklós, I., and Hein, J. (2014a). Efficient representation of uncertainty in multiple sequence alignments using directed acyclic graphs. *(submitted)*.

Herman, J. L., Aştefănoaei, M. S., Gratie, D.-E., Rich, C., Novák, A., Anderson, J. W. J., Lyngsø, R., and Hein, J. (2014b). RNA structure prediction in the presence of alignment uncertainty. *(in preparation)*.

Herman, J. L., Novák, A., and Hein, J. (2014c). Sequence annotation in the presence of alignment uncertainty. *(in preparation)*.

Herman, J. L., Challis, C. J., Novák, A., Hein, J., and Schmidler, S. C. (2014d). Simultaneous Bayesian estimation of alignment and phylogeny under a joint model of protein sequence and structure. *(in press)*.

Herman, J. L., Novák, A., Challis, C. J., Schmidler, S. C., and Hein, J. (2014e). StatAlign 3: Bayesian alignment and phylogenetics with protein structures. *(submitted)*.

Herman, J. L., Challis, C. J., Schmidler, S. C., Hein, J., Storz, J. F., Vinogradov, S. N., and Hoogewijs, D. (2014f). A structure-based molecular phylogeny identifies an ancient globin lineage from bacteria, to plants and animals. *(in preparation)*.

Hoffmann, F. G., Opazo, J. C., and Storz, J. F. (2010). Gene cooption and convergent evolution of oxygen transport hemoglobins in jawed and jawless vertebrates. *Proceedings of the National Academy of Sciences*, **107**(32), 14274–14279.

Hoffmann, F. G., Opazo, J. C., Hoogewijs, D., Hankeln, T., Ebner, B., Vinogradov, S. N., Bailly, X., and Storz, J. F. (2012a). Evolution of the globin gene family in deuterostomes: Lineage-specific patterns of diversification and attrition. *Molecular Biology and Evolution*, **29**(7), 1735–1745.

Hoffmann, F. G., Opazo, J. C., and Storz, J. F. (2012b). Whole-genome duplications spurred the functional diversification of the globin gene superfamily in vertebrates. *Molecular Biology and Evolution*, **29**(1), 303–312.

Holder, M. T., Lewis, P. O., Swofford, D. L., and Larget, B. (2005). Hastings ratio of the LOCAL proposal used in Bayesian phylogenetics. *Systematic Biology*, **54**(6), 961–965.

Holmes, I. and Durbin, R. (1998). Dynamic programming alignment accuracy. *Journal of Computational Biology*, **5**(3), 493–504.

Hopf, T., Colwell, L., Sheridan, R., Rost, B., Sander, C., and Marks, D. (2012). Three-

dimensional structures of membrane proteins from genomic sequencing. *Cell*, **149**(7), 1607–1621.

Hoy, J. A., Robinson, H., Trent III, J. T., Kakar, S., Smagghe, B. J., and Hargrove, M. S. (2007). Plant hemoglobins: A molecular fossil record for the evolution of oxygen transport. *Journal of Molecular Biology*, **371**(1), 168 – 179.

Hughes, A. L. (1994). The evolution of functionally novel proteins after gene duplication. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, **256**(1346), 119–124.

Höhl, M. and Ragan, M. A. (2007). Is multiple-sequence alignment required for accurate inference of phylogeny? *Systematic Biology*, **56**(2), 206–221.

Illergård, K., Ardell, D. H., and Elofsson, A. (2009). Structure is three to ten times more conserved than sequence: A study of structural response in protein cores. *Proteins: Structure, Function, and Bioinformatics*, **77**(3), 499–508.

Jagota, A., Lyngsø, R. B., and Pedersen, C. N. S. (2001). Comparing an HMM and an SCFG. In *Proceedings of the 1st Workshop on Algorithms in Bioinformatics (WABI)*, pages 69–84.

Johnson, M. S., Sali, A., and Blundell, T. L. (1990). Phylogenetic relationships from three-dimensional protein structures. volume 183 of *Methods in Enzymology*, pages 670 – 690. Academic Press.

Jordan, G. and Goldman, N. (2012). The effects of alignment error and alignment filtering on the sitewise detection of positive selection. *Molecular Biology and Evolution*, **29**(4), 1125–1139.

Karlin, S. and Altschul, S. F. (1993). Applications and statistics for multiple high-scoring

segments in molecular sequences. *Proceedings of the National Academy of Sciences*, **90**(12), 5873–5877.

Katoh, K. and Standley, D. M. (2013). MAFFT Multiple Sequence Alignment Software Version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, **30**(4), 772–780.

Kim, J. and Ma, J. (2011). PSAR: measuring multiple sequence alignment reliability by probabilistic sampling. *Nucleic Acids Research*, **39**(15), 6359–6368.

Kim, J., Pramanik, S., and Chung, M. J. (1994). Multiple sequence alignment using simulated annealing. *Computer Applications in the Biosciences : CABIOS*, **10**(4), 419–426.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, **16**(2), 111–120.

Kleinman, C. L., Rodrigue, N., Lartillot, N., and Philippe, H. (2010). Statistical potentials for improved structurally constrained evolutionary models. *Molecular Biology and Evolution*, **27**(7), 1546–1560.

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

Knudsen, B. and Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, **31**(13), 3423–3428.

Kumar, S., Filipski, A. J., Battistuzzi, F. U., Kosakovsky Pond, S. L., and Tamura, K. (2012). Statistics and truth in phylogenomics. *Molecular Biology and Evolution*, **29**(2), 457–472.

Lake, J. A. (1991). The order of sequence alignment can bias the selection of tree topology. *Molecular Biology and Evolution*, **8**(3), 378–385.

Lakner, C., van der Mark, P., Huelsenbeck, J. P., Larget, B., and Ronquist, F. (2008). Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Systematic Biology*, **57**(1), 86–103.

Lamy, J. N., Green, B. N., Toulmond, A., Wall, J. S., Weber, R. E., and Vinogradov, S. N. (1996). Giant hexagonal bilayer hemoglobins. *Chemical Reviews*, **96**(8), 3113–3124.

Landan, G. and Graur, D. (2007). Heads or tails: A simple reliability check for multiple sequence alignments. *Molecular Biology and Evolution*, **24**(6), 1380–1383.

Landan, G. and Graur, D. (2008). Local reliability measures from sets of co-optimal multiple sequence alignments. In *Pacific Symposium on Biocomputing*, volume 13, pages 15–24.

Landsmann, J., Dennis, E. S., Higgins, T. J. V., Appleby, C. A., Kortt, A. A., and Peacock, W. J. (1986). Common evolutionary origin of legume and non-legume plant haemoglobins. *Nature*, **324**, 166–168.

Larget, B. (2013). The estimation of tree posterior probabilities using conditional clade probability distributions. *Systematic Biology*, **62**(4), 501–511.

Larget, B. and Simon, D. (1999). Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, **16**(6), 750.

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

Lee, M. S. Y. (2001). Unalignable sequences and molecular evolution. *Trends in Ecology and Evolution*, **16**(12), 681–685.

Levy, E. D., Boeri Erba, E., Robinson, C. V., and Teichmann, S. A. (2008). Assembly reflects evolution of protein complexes. *Nature*, **453**(7199), 1262–1265.

Liu, K., Raghavan, S., Nelesen, S., Linder, C. R., and Warnow, T. (2009). Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, **324**(5934), 1561–1564.

Liu, K., Warnow, T. J., Holder, M. T., Nelesen, S. M., Yu, J., Stamatakis, A. P., and Linder, C. R. (2012). SATé-II: Very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Systematic Biology*, **61**(1), 90–106.

Liu, Y., Schmidt, B., and Maskell, D. L. (2010). MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. *Bioinformatics*, **26**(16), 1958–1964.

Löytynoja, A. and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(30), 10557–10562.

Lukatsky, D., Shakhnovich, B., Mintseris, J., and Shakhnovich, E. (2007). Structural similarity enhances interaction propensity of proteins. *Journal of Molecular Biology*, **365**(5), 1596 – 1606.

Lundin, D., Poole, A. M., Sjöberg, B.-M., and Högbom, M. (2012). Use of structural phylogenetic networks for classification of the ferritin-like superfamily. *Journal of Biological Chemistry*, **287**(24), 20565–20575.

Lunter, G. (2007). Probabilistic whole-genome alignments reveal high indel rates in the human and mouse genomes. *Bioinformatics*.

Lunter, G., Miklós, I., Drummond, A. J., Jensen, J. L., and Hein, J. (2003a). Bayesian phylogenetic inference under a statistical insertion-deletion model. In G. Benson and R. Page, editors, *Algorithms in Bioinformatics*, volume 2812 of *Lecture Notes in Computer Science*, pages 228–244. Springer Berlin Heidelberg.

Lunter, G., Drummond, A. J., Miklós, I., and Hein, J. (2005a). Statistical alignment: Recent progress, new applications, and challenges. In *Statistical Methods in Molecular Evolution*, Statistics for Biology and Health, pages 375–405. Springer New York.

Lunter, G., Rocco, A., Mimouni, N., Heger, A., Caldeira, A., and Hein, J. (2008). Uncertainty in homology inferences: Assessing and improving genomic sequence alignment. *Genome Research*, **18**(2), 298–309.

Lunter, G. A., Miklós, I., , Song, Y., and Hein, J. (2003b). An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *Journal of Computational Biology*, **10**, 869–889.

Lunter, G. A., Miklós, I., Drummond, A., Jensen, J. L., and Hein, J. (2005b). Bayesian coestimation of phylogeny and sequence alignment. *BMC Bioinformatics*, **6**(1), 83.

Lupyan, D., Leo-Macias, A., and Ortiz, A. R. (2005). A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics*, **21**(15), 3255–3263.

Löytynoja, A. and Goldman, N. (2008). Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, **320**(5883), 1632–1635.

Löytynoja, A. and Milinkovitch, M. C. (2001). Soap: cleaning multiple alignments from unstable blocks. *Bioinformatics*, **17**(6), 573–574.

Manning, G., Whyte, D. B., Martinez, R., Hunter, T., and Sudarsanam, S. (2002). The protein kinase complement of the human genome. *Science*, **298**(5600), 1912–1934.

Metzler, D. (2003). Statistical alignment based on fragment insertion and deletion models. *Bioinformatics*, **19**(4), 490–499.

Metzler, D., Fleissner, R., Wakolbinger, A., and von Haeseler, A. (2001). Assessing variability by joint sampling of alignments and mutation rates. *Journal of Molecular Evolution*, **53**(6), 660–669.

Mevissen, H.-T. and Vingron, M. (1996). Quantifying the local reliability of a sequence alignment. *Protein Engineering*, **9**(2), 127–132.

Meyer, I. M. and Miklós, I. (2007). SimulFold: Simultaneously inferring RNA structures including pseudoknots, alignments, and trees using a Bayesian MCMC framework. *PLoS Computational Biology*, **3**(8), e149.

Miklós, I., Novák, A., Dombai, B., and Hein, J. (2008). How reliably can we predict the reliability of protein structure predictions? *BMC Bioinformatics*, **9**(137).

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

Misof, B. and Misof, K. (2009). A Monte Carlo approach successfully identifies randomness in multiple sequence alignments: A more objective means of data exclusion. *Systematic Biology*.

Miyazawa, S. (1995). A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Engineering*, **8**(10), 999–1009.

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.

Morrison, D. A. and Ellis, J. T. (1997). Effects of nucleotide sequence alignment on phylogeny estimation: a case study of 18s rDNAs of apicomplexa. *Molecular Biology and Evolution*, **14**(4), 428–441.

Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48**(3), 443–453.

Nicholls, G. K., Fox, C., and Muir Watt, A. (2012). Coupled MCMC with a randomized acceptance probability. *arXiv:1205.6857*.

Notredame, C. and Higgins, D. G. (1996). Saga: sequence alignment by genetic algorithm. *Nucleic Acids Research*, **24**(8), 1515–1524.

Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, **302**(1), 205–217.

Novák, A., Miklós, I., Lyngsø, R., and Hein, J. (2008). StatAlign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees. *Bioinformatics*, **24**(20), 2403–2404.

Ogden, T. H. and Rosenberg, M. S. (2006). Multiple sequence alignment accuracy and phylogenetic inference. *Systematic Biology*, **55**(2), 314–328.

Panchenko, A. R., Wolf, Y. I., Panchenko, L. A., and Madej, T. (2005). Evolutionary plasticity of protein families: Coupling between sequence and structure variation. *Proteins: Structure, Function, and Bioinformatics*, **61**(3), 535–544.

Penn, O., Privman, E., Landan, G., Graur, D., and Pupko, T. (2010a). An alignment confidence score capturing robustness to guide tree uncertainty. *Molecular Biology and Evolution*, **27**(8), 1759–1767.

Penn, O., Privman, E., Ashkenazy, H., Landan, G., Graur, D., and Pupko, T. (2010b). GUIDANCE: a web server for assessing alignment confidence scores. *Nucleic Acids Research*, **38**(suppl 2), W23–W28.

Privman, E., Penn, O., and Pupko, T. (2012). Improving the performance of positive selection inference by filtering unreliable alignment regions. *Molecular Biology and Evolution*, **29**(1), 1–5.

Rannala, B., Zhu, T., and Yang, Z. (2012). Tail paradox, partial identifiability, and influential priors in Bayesian branch length inference. *Molecular Biology and Evolution*, **29**(1), 325–335.

Rastogi, S. and Liberles, D. (2005). Subfunctionalization of duplicated genes as a transition state to neofunctionalization. *BMC Evolutionary Biology*, **5**(1), 28.

Redelings, B. D. and Suchard, M. A. (2005a). Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, **54**(3), 401–418.

Redelings, B. D. and Suchard, M. A. (2005b). Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, **54**, 401–418.

Redelings, B. D. and Suchard, M. A. (2011). *Robust inferences from ambiguous alignments*, pages 209–271. Sequence Alignment: Methods, Models, Concepts and Strategies. University of California Press.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society Series B (Statistial Methodology)*, **60**(1), 255–268.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, **7**, 110–120.

Robinson, D. and Foulds, L. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, **53**(1–2), 131 – 147.

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

Rodrigue, N., Lartillot, N., Bryant, D., and Philippe, H. (2005). Site interdependence attributed to tertiary structure in amino acid sequence evolution. *Gene*, **347**(2), 207 – 217.

Rodriguez, A. and Schmidler, S. C. (2014). Bayesian protein structure alignment. (*submitted*).

Ronquist, F. and Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, **19**(12), 1572–1574.

Roshan, U. and Livesay, D. R. (2006). Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinformatics*, **22**(22), 2715–2721.

Rueda, M., Ferrer-Costa, C., Meyer, T., Pérez, A., Camps, J., Hospital, A., Gelpí, J. L., and Orozco, M. (2007). A consensus view of protein dynamics. *Proceedings of the National Academy of Sciences*, **104**(3), 796–801.

Ruffieux, Y. and Green, P. J. (2009). Alignment of multiple configurations using hierarchical models. *Journal of Computational and Graphical Statistics*, **18**(3), 756–773.

Sahraeian, S. M. E. and Yoon, B.-J. (2010). PicXAA: greedy probabilistic construction of maximum expected accuracy alignment of multiple sequences. *Nucleic Acids Research*, **38**(15), 4917–4928.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjölander, K., Underwood, R. C., and Haussler, D. (1994). Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, **22**(23), 5112–5120.

Sali, A. and Blundell, T. (1993). Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, **234**(3), 779–815.

Satija, R., Pachter, L., and Hein, J. (2008). Combining statistical alignment and phylogenetic footprinting to detect regulatory elements. *Bioinformatics*, **24**(10), 1236–1242.

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

Schmidler, S. C. (2006). Fast Bayesian shape matching using geometric algorithms (with discussion). In J. M. Bernardo, S. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 8*, pages 471–490, Oxford. Oxford University Press.

Schneider, T. R. (2000). Objective comparison of protein structures: error-scaled difference distance matrices. *Acta Crystallographica Section D*, **56**(6), 714–721.

Schwartz, A. S. (2007). *Posterior decoding methods for optimization and accuracy control of multiple alignments*. Ph.D. thesis, University of California, Berkeley.

Schwartz, A. S. and Pachter, L. (2007). Multiple alignment by sequence annealing. *Bioinformatics*, **23**(2), e24–e29.

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

Schwarzenbacher, R., Godzik, A., Grzechnik, S. K., and Jaroszewski, L. (2004). The importance of alignment accuracy for molecular replacement. *Acta Crystallographica Section D*, **60**(7), 1229–1236.

Schwikowski, B. and Vingron, M. (2003). Weighted sequence graphs: boosting iterated dynamic programming using locally suboptimal solutions. *Discrete Applied Mathematics*, **127**(1), 95–117.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., Weinstock, G. M., Wilson, R. K., Gibbs,

R. A., Kent, W. J., Miller, W., and Haussler, D. (2005). Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research*, **15**(8), 1034–1050.

Simmons, M. P., Müller, K. F., and Norton, A. P. (2010). Alignment of, and phylogenetic inference from, random sequences: The susceptibility of alternative alignment methods to creating artifactual resolution and support. *Molecular Phylogenetics and Evolution*, **57**(3), 1004–1016.

Sinha, S. and He, X. (2007). MORPH: Probabilistic alignment combined with hidden Markov models of cis-regulatory modules. *PLoS Computational Bioloy*, **3**(11), e216.

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

Storz, J. F., Opazo, J. C., and Hoffmann, F. G. (2013). Gene duplication, genome duplication, and the functional diversification of vertebrate globins. *Molecular Phylogenetics and Evolution*, **66**(2), 469–478.

Suchard, M. A. and Redelings, B. D. (2006). Bali-phy: simultaneous bayesian inference of alignment and phylogeny. *Bioinformatics*, **22**(16), 2047–2048.

Szabó, A., Novák, A., Miklós, I., and Hein, J. (2010). Reticular alignment: A progressive corner-cutting method for multiple sequence alignment. *BMC Bioinformatics*, **11**(1), 570.

Talavera, G. and Castresana, J. (2007). Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Systematic Biology*, **56**(4), 564–577.

Tavaré, S. (1986). Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences*, **17**, 57–86.

Teh, A.-H., Saito, J. A., Baharuddin, A., Tuckerman, J. R., Newhouse, J. S., Kanbe, M., Newhouse, E. I., Rahim, R. A., Favier, F., Didierjean, C., Sousa, E. H., Stott, M. B., Dunfield, P. F., Gonzalez, G., Gilles-Gonzalez, M.-A., Najimudin, N., and Alam, M. (2011). Hell's Gate globin I: An acid and thermostable bacterial hemoglobin resembling mammalian neuroglobin. *FEBS Letters*, **585**(20), 3250 – 3258.

Terwilliger, N. (1992). Molecular structure of the extracellular heme proteins. In C. Mangum, editor, *Blood and Tissue Oxygen Carriers*, volume 13 of *Advances in Comparative and Environmental Physiology*, pages 193–229. Springer Berlin Heidelberg.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, **22**(22), 4673–4680.

Thompson, J. D., Plewniak, F., and Poch, O. (1999). BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15**(1), 87–88.

Thompson, J. D., Koehl, P., Ripp, R., and Poch, O. (2005). Balibase 3.0: Latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, **61**(1), 127–136.

Thompson, J. D., Linard, B., Lecompte, O., and Poch, O. (2011). A comprehensive benchmark study of multiple sequence alignment methods: Current challenges and future perspectives. *PLoS ONE*, **6**(3), e18093.

Thorne, J. L. and Churchill, G. A. (1995). Estimation and reliability of molecular sequence alignments. *Biometrics*, **51**(1), 100–113.

Thorne, J. L. and Kishino, H. (1992). Freeing phylogenies from artifacts of alignment. *Molecular Biology and Evolution*, **9**(6), 1148–1162.

Thorne, J. L., Kishino, H., and Felsenstein, J. (1991). An evolutionary model for maximum likelihood alignment of dna sequences. *Journal of Molecular Evolution*, **33**(2), 114–124.

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.

Thorne, J. L., Kishino, H., and Painter, I. S. (1998). Estimating the rate of evolution of the rate of molecular evolution. *Molecular Biology and Evolution*, **15**(12), 1647–1657.

Tiana, G., Shakhnovich, B. E., Dokholyan, N. V., and Shakhnovich, E. I. (2004). Imprint of evolution on protein structures. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(9), 2846–2851.

Tramontano, A., Leplae, R., and Morea, V. (2001). Analysis and assessment of comparative modeling predictions in CASP4. *Proteins: Structure, Function, and Bioinformatics*, **45**(S5), 22–38.

Tweedie, S., Ashburner, M., Falls, K., Leyland, P., McQuilton, P., Marygold, S., Millburn, G., Osumi-Sutherland, D., Schroeder, A., Seal, R., Zhang, H., and Consortium, T. F. (2009). FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucleic Acids Research*, **37**(suppl 1), D555–D559.

Vázquez-Limón, C., Hoogewijs, D., Vinogradov, S. N., and Arredondo-Peter, R. (2012). The evolution of land plant hemoglobins. *Plant Science*, **191 - 192**, 71 – 81.

Vingron, M. (1996). Near-optimal sequence alignment. *Current Opinion in Structural Biology*, **6**(3), 346–352.

Vingron, M. and Argos, P. (1990). Determination of reliable regions in protein sequence alignments. *Protein Engineering*, **3**(7), 565–569.

Vinogradov, S. N., Hoogewijs, D., Bailly, X., Arredondo-Peter, R., Guertin, M., Gough, J., Dewilde, S., Moens, L., and Vanfleteren, J. R. (2005). Three globin lineages belonging to two structural classes in genomes from the three kingdoms of life. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(32), 11385–11389.

Wallace, I. M., O'Sullivan, O., Higgins, D. G., and Notredame, C. (2006). M-Coffee: combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Research*, **34**(6), 1692–1699.

Wang, L. and Jiang, T. (1994). On the complexity of multiple sequence alignment. *Journal of Computational Biology*, **1**(4), 337–348.

Wang, L.-S., Leebens-Mack, J., Wall, P. K., Beckmann, K., de Pamphilis, C. W., and Warnow, T. (2011). The impact of multiple protein sequence alignment on phylogenetic estimation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **8**(4), 1108–1119.

Wang, R. and Schmidler, S. C. (2014). Bayesian multiple protein structure alignment. In R. Sharan, editor, *Research in Computational Molecular Biology*, volume 8394 of *Lecture Notes in Computer Science*, pages 326–339. Springer International Publishing.

Warnow, T. (2012). Standard maximum likelihood analyses of alignments with gaps can be statistically inconsistent. *PLoS Currents*, **4**.

Washietl, S., Pedersen, J. S., Korbel, J. O., Stocsits, C., Gruber, A. R., Hackermüller, J., Hertel, J., Lindemeyer, M., Reiche, K., Tanzer, A., Ucla, C., Wyss, C., Antonarakis, S. E., Denoeud, F., Lagarde, J., Drenkow, J., Kapranov, P., Gingeras, T. R., Guigó, R., Snyder, M., Gerstein, M. B., Reymond, A., Hofacker, I. L., and Stadler, P. F. (2007). Structured RNAs in the ENCODE selected regions of the human genome. *Genome Research*, **17**, 852–864.

Waterman, M. S. and Byers, T. H. (1985). A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences*, **77**(1–2), 179–188.

Webb, B.-J. M., Liu, J. S., and Lawrence, C. E. (2002). Balsa: Bayesian algorithm for local sequence alignment. *Nucleic Acids Research*, **30**(5), 1268–1277.

Westesson, O., Lunter, G., Paten, B., and Holmes, I. (2012). Accurate reconstruction of insertion-deletion histories by statistical phylogenetics. *PLoS ONE*, **7**(4), e34572.

Wheeler, W. C. (1995). Sequence alignment, parameter sensitivity, and the phylogenetic analysis of molecular data. *Systematic Biology*, **44**(3), 321–331.

Wise, M. J. (2010). Not so HoT? Heads or tails is not able to reliably compare multiple sequence alignments. *Cladistics*, **26**(4), 438–443.

Wolfsheimer, S., Hartmann, A., Rabus, R., and Nuel, G. (2012). Computing posterior probabilities for score-based alignments using ppalign. *Statistical Applications in Genetics and Molecular Biology*, **11**(4), Article 1.

Wong, K. M., Suchard, M. A., and Huelsenbeck, J. P. (2008). Alignment uncertainty and genomic analysis. *Science*, **319**(5862), 473–476.

Wood, T. C. and Pearson, W. R. (1999). Evolution of protein sequences and structures. *Journal of Molecular Biology*, **291**(4), 977 – 995.

Wu, M., Chatterji, S., and Eisen, J. A. (2012). Accounting for alignment uncertainty in phylogenomics. *PLoS ONE*, **7**(1), e30288.

Yau, C. and Holmes, C. C. (2013). A decision-theoretic approach for segmental classification. *The Annals of Applied Statistics*, **7**(3), 1814–1835.

Yu, L. and Smith, T. (1999). Positional statistical significance in sequence alignment. *Journal of Computational Biology*, **6**(2), 253–259.

Zhu, J., Liu, J. S., and Lawrence, C. E. (1998). Bayesian adaptive sequence alignment algorithms. *Bioinformatics*, **14**(1), 25–39.

Zuker, M. (1991). Suboptimal sequence alignment in molecular biology: Alignment with error analysis. *Journal of Molecular Biology*, **221**(2), 403–420.