Statistical Approaches to Phylogenetic Networks, Recombination and Testing of Incongruence

April 2011

Alethea Rea

Supervised by

David Bryant and Rachel Fewster

A dissertation submitted for the degree of
Doctor of Philosophy in Statistics
at the University of Auckland, New Zealand

Abstract

Phylogenetics is the study of relationships between species using Deoxyribose Nucleic Acid (DNA). This thesis takes a statistical approach to two phenomenon which violate the assumption that evolution is treelike, and examines ways of visualising non-treelike signal.

We use networks to display phylogenetic signal as they are robust and capable of displaying uncertainty. Phylogenetic network inference involves estimating discrete (topology) and continuous (branch length) parameters. One particular class of phylogenetic networks, *split networks*, can be viewed as points in Euclidean space of high dimension. In theory, then, phylogenetic analysis become a problem of inferring simple real valued parameters. In this thesis we report on our experiences turning this theory into practice. We use the Least Absolute Shrinkage and Selection Operator (LASSO) approach to regression in the first instance and then extend the LASSO to a partial LASSO.

Within genes, phenomena like recombination (combining genetic material from more than one source) leads to non-treelike evolutionary histories. We introduce two methods for estimating the location of a recombination event. The first method is based on detecting a regime shift in the presence of recombination and the second method models the signal in each pair of DNA sites.

Even if each gene has a treelike evolutionary history, the histories may not be shared. Therefore, we developed an approach to constructing a confidence set of topologies for a set of genes. If this set is empty then the genes do not share an evolutionary history.

We conclude that the new statistical approaches to these phenomena, developed here, can give further insight into an evolutionary history.

In memory of Grandpa

Acknowledgements

I would like to start by thanking my supervisors David Bryant and Rachel Fewster. My particular thanks to David for providing me with the opportunity to live in Germany, for many thoughtful discussions and a great deal of support.

I would like to thank the following hosts and institutions for allowing me to visit: Bill Martin, Tal Dagan and Christian Eßer at Hienrich-Henie Universität, Düsseldorf, Germany; Mike Steel and Marco Reale, University of Canterbury, Christchurch, New Zealand; and David Bryant, University of Otago, Dunedin, New Zealand.

I would like to thank Bill Rea for many helpful discussions and valuable input particularly regarding the work on recombination breakpoint detection. I would also like to thank him for editorial work on the thesis and valuable feedback on my presentations. I would like to thank Jessica Leigh and Peter Tsai for implementing the ideas on confidence sets. I would like to thank Raazesh Sainudiin for helpful discussions. I would like to thank Fraser Dron for third party editing. I would like to thank all those who attended one of my talks at any of the conferences I have been to, particularly those who asked questions and gave me feedback. To each of my office mates, thank you for the great conversations and for the times you supported me.

I would like to thank Avner Bar-Hen for providing me with R code for calculating the influence function and David Bryant for implementing our methods in SplitsTree.

I would like to thank the Tertiary Education Commission, New Zealand, the Marsden Fund, New Zealand and the Humblebolt Fellowship for financial support. Additional thanks to the Royal Society of New Zealand for funding my attendance at the Mathematics, Evolution, and Development Conference, China in 2010.

Thank you to the friends and family who supported me throughout these three years regardless of how often I changed locations. In particular thanks to my parents whose friendship never waivers.

Contents

1

Introduction

1.1 Phylogenetics and its assumptions

Molecular phylogenetics is the study of evolutionary relationships. The main goal of most phylogenetic studies is to reconstruct an evolutionary history; that is, estimate which organisms have the most recent common ancestors, and when those ancestors were on earth.

Statistical inference of an evolutionary history is only possible if one is willing to make several assumptions. One of the standard assumptions is that the genetic sequences being studied all evolved along the same evolutionary 'tree-of-life'. There is more and more evidence that this is not the case in a wider and wider variety of situations. Bacteria have complex evolutionary histories and a network seems more appropriate for representing aspects of of their history (Puigbò et al., 2010). Even eukaryotes can inherit genetic material from non-parental sources (see Andresson (2005); Keeling and Palmer (2008) for

Figure 1.1: Left: An illustrative phylogenetic tree based on five taxa. Right: An illustrative phylogenetic network with a single reticulation, or box, based on four taxa.

reviews.) This creates the problem of **phylogenetic heterogeneity**, where a single tree is unable to fully explain the evolutionary history.

The thesis falls naturally into two parts; the firsts look at ways of visualising heterogeneity using networks, the second looks at sources of heterogeneity: recombination and incongruence.

1.2 Networks

Phylogenetic networks is a broad term describing a range of graphical diagrams built from sequences, distances, genealogies, or trees. Phylogenetic trees are themselves a subset of phylogenetic networks.

Huson and Bryant (2006) classify networks into three categories: reticulate networks, split networks, and 'other'.

The 'other' category describes trees which had at least one more branch added to them. These representations are used to display a piece of DNA which is incorporated directly from a non-ancestral source.

Reticulate networks can display events involving an organism gaining additional genetic material such as an extra set of chromosomes. For some examples, see Maddison (1997); Baroni et al. (2004); Nakleh et al. (2005). Reticulate networks can also display recombination events within a population, for some examples see Hudson (1983).

Split networks can be categorised by the type of data used to construct them: median networks are constructed from sequences; consensus networks are constructed from trees; additionally there is a class of split networks which are constructed from distances. We review each in turn.

The median network (Bandelt et al., 1995) represents all of the trees which use the smallest number of substitutions to explain the site patterns.

Consensus networks are a way to visualise multiple trees at once; they are constructed from the branches of trees. These network methods look at the branches (also called splits) of each tree and draw a network that contains all (or in some cases a subset) of the splits from all of the input trees; see Bandelt et al. (1995); Holland and Moulton (2003).

Split networks are used to visualise a set of splits which do not form a single tree. They do not represent an explicit evolutionary history as the nodes do not represent a common ancestor. Nonetheless, split networks are useful because they provide insight into how tree-like the distances between taxa are. Some split network methods include splits decomposition (Bandelt and Dress, 1992), neighbor-net (Bryant and Moulton, 2004) and Q-net (Grünewald et al., 2007).

We discuss networks in detail in Chapters 2, 4 and 5.

1.3 Incongruence

Incongruence means 'a lack of agreement'. Within phylogenetics, incongruence between genes refers to two or more genes having evolved under at least two different trees.

While genetic data was once scarce, and the use of more than one gene was rare, that is certainly no longer the case. It is now commonplace to get alignments with multiple genes on a set of taxa. It is also common for not all genes to be available for all taxa, and therefore the alignments are often 'patchy'. Sanderson et al. (2010) explores the limits of phylogenomic inference with patchy data.

There are two approaches to this type of data. The first is to concatenate all the genes into a single alignment and use this alignment to build a phylogenetic tree. This was

the approach taken by Baldauf (1999) and many others since. The second approach is to build a tree on each gene and then build a 'supertree' from all of the subtrees (for a review of this approach see Cotton and Wilkinson (2008)).

One concern is that the set of genes may not have evolved on the same topology. When the genes have different evolutionary histories reconstructing a single tree is meaningless. Congruence testing is interested in finding genes which did evolve on the same topology as these genes can form the basis of a multi-gene analysis.

We look at incongruence in Chapter 6.

1.4 Recombination

Recombination occurs when an organism inherits genetic material from more than one source.

Biologists are interested in several aspects of recombination: the presence or absence of recombination; the recombination rate; the presence or absence of hotspots; the location of recombination events; and estimation of the number of recombination events. Each of these are discussed below.

There is a wealth of literature on recombination detection. For reviews of recombination detection methods see Brown et al. (2001), Wiuf et al. (2001), Posada and Crandall (2001), Posada (2002), and Bruen et al. (2006). Posada and Crandall (2001) and Posada (2002) compared 14 methods which detect if recombination is present. They concluded that one method on its own was not sufficient to determine recombination but rather that a wide range of methods should be used and a 'majority rules' approach taken to determine recombination.

One type of recombination event is meiotic crossover, where segments of the maternal and paternal DNA in a sexually reproducing organism, are swapped, leading to a chromosomal reassortment.

Population geneticists use a recombination rate parameter in their models of inheritance for sexually reproducing organisms. If recombination is present, incorporating this parameter into these models gives rise to more accurate population parameters. Work in

this field first involved the description of models with recombination (Hudson, 1983), and second ways of estimating the model parameters. Stumpf and McVean (2003) provide an overview of this field.

Sexually reproducing organisms have recombination 'hotspots'. These are small patches of the chromosome in which the meiotic crossover level is particularly high (for an overview on hotspots see Hey (2004)). When such hotspots are potentially present, they can be searched for using the methodologies of Li and Stephens (2003); Myers and Griffiths (2003); Wall and Pritchard (2003); Crawford et al. (2004); Fearnhead et al. (2004); McVean et al. (2004).

Estimating the number of recombination events is useful for understanding events at the site level. Many recombination events leave no detectable signal in the data and therefore the majority of methods, which are based on detecting recombination signals, underestimate the occurrence of recombination. Investigations into detecting recombination can be hampered by the recombined fragment being similar to the one it replaces or subsequent mutations weakening the signal (Chan et al., 2006). Some methods look at calculating the minimum possible number of recombination events such as those of Myers and Griffiths (2003) and Song et al. (2007).

There are other events that fall into the category of recombination events. These include gene conversion, transformation, conjugation, and transduction. These types of recombination events move large segments of DNA into the genome and lead to a mosaic of origins in the DNA. We briefly describe these events.

Gene conversion occurs when there is a mismatch in repair during a crossover event. The consequence is a piece of DNA is transfered from one chromosome to another within an organism. Therefore the genetic material incorporated is from a homologous gene. There is a body of work which looks at ways of estimating the extent of crossover and gene conversion simultaneously. For example, Wiuf and Hein (2000) developed an extension of the coalescent that incorporated gene conversion, and Padhukasahasram et al. (2006) provided an example of a method developed to estimate both quantities using single nucleotide polymorphism data.

Bacteria can obtain new genetic information in many ways including: taking up DNA from the surrounding medium (transformation); a direct exchange of genetic material

with a donor (conjugation); and importing material from viruses (transduction).

There is a growing body of biochemical literature which shows that incorporation of new genetic material is not restricted to bacteria and in fact may happen across orders and families. Doolittle et al. (1990) were the first to show that this could occur across kingdoms by demonstrating a probable transfer of a glyceraldehyde-3-phosphate dehydrogenase from a eukaryote to a bacteria. Their finding was based on sequence similarities and phylogenetic analysis.

Recombination events must be detected prior to carrying out phylogenetic analysis. If recombination is detected, one should then carry out further investigations into the taxa and location of the recombination event(s). Alternatively, one can apply any of the methods that account for horizontal gene transfer explicitly; see Birin et al. (2008) for an example.

In the event that recombinations are present but are not detected, they will negatively impact a phylogenetic tree analysis. Schierup and Hein (2000) report that tree estimation procedures applied to data with recombination led to longer terminal branches, larger total tree heights and a smaller time to the most recent common ancestor. Casola and Hahn (2009) show that undetected gene conversion on duplicated genes leads to a high false positive rate when looking for positive selection. When recombination is not accounted for the resulting parameter estimates and inferences are not reliable.

In each of Chapters 7 and 8, we introduce a new method for detecting the location of a recombination event.

1.5 Summary of thesis contributions

In this thesis we first examine ways of visualising heterogeneity using networks.

Chapter 2 focuses on the use of regularisation techniques such as the LASSO to infer split networks.

The main finding of this work is that neighbor-net networks can be simplified, but that this simplification makes little difference to the visualisation of the networks.

Chapter 3 discusses testing for non-treelike data. The null hypothesis we use is that

these sequences arise from a tree-like evolutionary process. The alternative hypothesis is that evolution is not tree-like. The unsuccessful test we developed compared the Akaike information criteria (Akaike, 1974) on a tree and a network to discover whether the network contained more information or explanatory power. The majority of the chapter is devoted to understanding several aspects of the test which led it to being ineffective.

The original contribution of Chapter 3 is the extensive investigation into the sources of its inefficacy.

The main outcome of this work is that this approach to testing for treenesss is ineffective. There is no single component that explains the high level (or type I error) and low power of this test.

Chapter 4 deals with extending the LASSO regression approach (Tibshirani, 1996) to the partial LASSO. The partial LASSO allows the user to define a set of variables for the initial model; the LASSO is then applied to the remaining variables. Therefore, the partial LASSO provides a hybrid between least squares and efficient estimation of a regression model using the LASSO.

The partial LASSO algorithm is novel. The main results are the theorem and proof of the partial LASSO, and its application to neighbor-net networks.

The last network chapter (Chapter 5) describes an extension of consensus networks. The original contribution and main result of the work on consensus networks is a statistically rigorous method for combining a set of phylogenetic trees.

In the second half of the thesis we look at two sources of heterogeneity: incongruence, and recombination.

Chapter 6 discusses our development of a congruency test. The method tests the null hypothesis that all the genes have evolved on the same tree using composite p-value methods. A composite p-value combines p-values that test the null hypothesis that an alignment of just one gene evolved on a specific topology. The method we developed took a p-value for each gene for each topology and combined them to get a p-value for each topology over all the genes. The method used to combine the p-values is the Z-score composite method, or Stouffer's method (Stouffer et al., 1949). If a tree is not rejected as plausible, it is added to the confidence set. In this way, we end up with a confidence set of topologies for the complete set of genes.

The main result of this work is the development of a method that gives a statistically valid confidence set of possible topologies.

Chapters 7 and 8 deal with recombination. Each chapter introduces a recombination breakpoint detection method.

The original contribution of Chapter 7 is the use of breakpoint detection methods from time series analysis on two series that have information regarding recombination. The first series is based on the influence of a single site on the phylogenetic tree estimated by maximum likelihood (Bar-Hen et al., 2008). The second series is based on changes in the neighbor-joining tree branch lengths.

The main finding of this work is that when the sequence divergence increases, the power to detect recombination using these methods increased. The series based on the influence function shows promise as a hotspot detection method.

The original contribution of Chapter 8 is another recombination breakpoint detection method. The model is based on first principles and an understanding of the expected impact a recombination event has on the observed incompatibility score. The model returns an ordering from the most likely to least likely breakpoints, hereby giving an optimal set of breaks for each specified number of potential recombination events.

The main result of this work is that incompatibility shows great potential at detecting recombination event boundaries.

The thesis ends with a discussion chapter that summarises the key findings of the thesis.

Almost every aspect of this thesis has been presented at a conference or workshop. Aspects of the work on networks and treeness was presented at Phylogenetics New Zealand, New Zealand in 2008, the Society for Molecular Biology and Evolution, Spain in 2008, the Australian meeting on Phylogenetic and Evolution, Australia in 2009, and the Joint Statistical Association meeting, Canada in 2010. The work on the independence of the neighbor-net networks and the PHI test was presented at the Evolution meeting, USA, in 2009. The work on the congruence test was presented at Phylogenetics New Zealand, New Zealand in 2010. The work on using the influence function to detect recombination breakpoints was presented at Phylomania, Australia in 2010.

2

Splits selection for phylogenetic networks

2.1 Background and motivation

Visualising heterogeneity is the subject of this chapter, and of Chapters 4 and 5. All three chapters focus on networks. In this chapter we introduce a statistically rigorous way of inferring split networks.

Phylogenetic trees are awkward subjects for statistical analysis. The trees are mixtures of discrete (topology) and continuous (branch length) parameters, and the continuous parameters for one tree generally do not correspond to continuous parameters for another. The combinatorics of the space of trees is itself arbitrarily complex (Semple and Steel, 2003).

Inference of phylogenetic trees is carried out in two steps. The first step is to infer

the topology of the tree; that is to determine which branches represent the evolutionary history. The second step is to infer the branch lengths.

In a similar manner, phylogenetic split network inference is a two step procedure: choosing a set of splits, and estimating the lengths of the splits. The lengths of the splits are called split weights.

Estimating the split weights is of central importance to network methods such as neighbor-net (Bryant and Moulton, 2004) and Q-Net (Grünewald et al., 2007) where poor estimation can lead to an overly complex representation of the alignment. The neighbor-net networks have been criticised as having a high false positive rate; that is, they have too many branches in their networks (Nakleh et al., 2005). Usually, a proportion of the splits are associated with small split weights and including them only clutters the phylogenetic signal.

In this chapter, we use a modelling framework based on linear regression to estimate the split weights, sub-setting some of them to zero. We apply linear regression and the positive LASSO algorithm to the problem of picking the neighbor-net splits and split weights. We carry out several experiments into components of the regression framework and into the effectiveness of the framework in reducing the clutter seen in neighbor-net networks.

2.2 Modelling

In this section we introduce linear models in phylogenetics, estimating the covariance matrix and our method for simulating alignments.

2.2.1 Regression estimation of split weights

In this chapter we focus on distance based phylogenetic methods. The distances are measured on pairs of taxa and therefore they do not contain information on higher order relationships. However, Felsenstein (2004) notes that little information is lost and that the majority of the information on evolutionary relationships in contained in pairwise

distances. Therefore, distance based methods assume the estimated pairwise distances represent the true evolutionary time plus noise.

Given a method of clustering the taxa and defining the tree structure, the distances are used to calculate the branch lengths. Each distance is the sum of the lengths of the branches that separate the two taxa. The branch lengths are typically estimated using least squares (Cavalli-Sforza and Edwards, 1967; Vach, 1989; Rzhetsky and Nei, 1992; Gascuel, 1997).

We also assume that the observed distances are equal to the true distances (or additive distances) plus some error. The true distance between two taxa is the sum of the branch lengths on the path connecting them. Hence, the vector of observed distances $\mathbf{y}$ can be written

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon \tag{2.1}$$

where each row of $\mathbf{y}$ is indexed by a pair of taxa and each row of $\mathbf{X}$ is also indexed by pairs of taxa. The entries of $\mathbf{X}$ are ones and zeros picking out exactly those edges on the corresponding path between the pairs of taxa. The vector $\boldsymbol{\beta}$ is the vector of estimated split weights. Therefore, the expected distances are modelled using $\mu_{ij} = (\mathbf{X}\boldsymbol{\beta})_{ij}$. It is the inferred $\boldsymbol{\beta}_i$ values that contain information on evolutionary relationships.

An example of a split matrix for a five taxa tree can be seen in Figure 2.1.

The neighbor-net method of Bryant and Moulton (2004) used non-negative least squares to estimate the split weights given the distance vector and a set of splits known as the circular splits. We use linear regression. The model of Equation (2.1) extends directly to a split network, so long as $\mathbf{X}$ contains the splits for the network.

The first step of either neighbor-net or Q-Net is to select a set of candidate splits. For neighbor-net these splits come from a circular ordering of the taxa; that is, there is an ordering of the taxa $t_1, t_2, \ldots, t_M$ such that every split is of the form $\{t_i, t_{i+1}, \ldots, t_j\}|T - \{t_i, \ldots, t_j\}$ for some i and j satisfying $1 \leq i \leq j < M$. Our methods however are not limited to this set, and can be applied to any set of splits.

Linear regression makes four assumptions, namely a linear model is appropriate, the errors are independently and identically distributed according to the normal distribution; the columns of $\mathbf{X}$ are linearly independent; and each element of the predictive vector is an

$$X = \begin{array}{c} \\ ab \\ ac \\ ad \\ ae \\ bc \\ bd \\ be \\ cd \\ ce \\ de \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

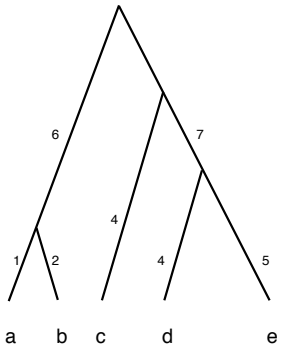

Figure 2.1: Splits labelled one to seven. Left: Matrix of seven splits for a five taxa tree where 0 means the two taxa are on the same side of the split and 1 means that one of the taxa is on each side. Right: A symbolic tree on five taxa.

independent observation. While the first three pose no problems in the implementation of the framework the fourth does not hold as the taxa have evolutionary history in common and therefore are not independent. One potential way to deal with this assumption is to estimate a covariance matrix for the error in the distances and apply a transformation to the regression problem. Therefore we considered two ways of estimating this covariance matrix, both presented in Bulmer (1991).

2.2.2 Estimating the covariance

The first covariance estimator makes no assumptions about the evolutionary structure of the data. It only assumes that the distances have been corrected using the Jukes-Cantor distance correction (Jukes and Cantor, 1969). The formula for the variance was also presented in Kimura and Ohta (1972). It applies the delta method (Stuart and Ord, 1987) to give approximations for the variance and covariances. These are

$$\mathrm{var}(d_{ij}) = \frac{p_{ij}(1-p_{ij})}{L}\left(\frac{1}{1-p_{ij}/b_{ij}}\right)^2, \quad \text{and} \tag{2.2}$$

$$\mathrm{cov}(d_{ij}, d_{kl}) = \frac{1}{\left(1-\frac{p_{ij}}{b_{ij}}\right)}\frac{1}{\left(1-\frac{p_{kl}}{b_{kl}}\right)}\left[\frac{(p_{ij,kl}-p_{ij}p_{kl})}{L}\right], \tag{2.3}$$

where L is the number of sites, $p_{ij,kl}$ is the proportion of site where i differs from j and k differs from l, and p_{ij} is the proportion of site where i differs from j. The quantity p_{ij} has been assumed to be a proportion from a binomial distribution. The parameter b is the expected proportion of sites differences when the two sequences are independent of each other. These formulae can be extended to handle missing data.

The second estimator, also from Bulmer (1991), is based on a measure of the shared history between two taxa. Therefore it assumes there is an evolutionary relationship between the taxa and that this is represented by the common path. The formula was also presented in Nei and Jin (1989):

$$\operatorname{cov}(d_{ij}, d_{kl}) = b\left[(1-b)\exp\frac{2\delta}{d} + (2b-1)\exp\frac{\delta}{b} - b\right]/L \tag{2.4}$$

where δ is a measure of the distance in the common path between i, j, k and l, b is the expected proportion of sites differences when two sequences are independent of each other and L is the number of sites. This formula can also be adapted to account for missing data.

The value, δ is given by

$$\delta = \sum_r \mathbf{X}_{ij;r}\mathbf{X}_{kl;r}\boldsymbol{\beta}_r \tag{2.5}$$

since this equals the sum of the split weights over all splits that separate both i and j and k and l. The notation in Equation (2.5) differs slightly from that of Bulmer (1991) in that $\boldsymbol{\beta}$ is used to reflect the inferred branch lengths.

Equivalently,

$$\delta = (\mathbf{XWX})_{ij;kl}, \tag{2.6}$$

where

$$\mathbf{W} = \operatorname{diag}(\boldsymbol{\beta}). \tag{2.7}$$

Conveniently, this formula makes no assumption that the splits are from a tree; therefore, the formula can be used for both trees and split networks. This is valid as distance based tree methods can be extended by split networks as shown in Bryant (2005).

When applying covariance matrix, that is when we transform the data, we use the inverse of the upper triangular Cholesky decomposition matrix.

2.2.3 Data simulation

There are standard ways of simulating sequence alignments on a tree. It is possible to construct a topology manually; or you can construct one according to a distribution. Once the tree has been chosen the alignment is created by evolving sites down the tree.

Simulating networks is more complex. One widely-used approach is to use recombination within a single population to simulate alignments which are non-treelike. One algorithm for simulating recombination is that of Hudson (1983).

The procedure has two steps. The first step generates an ancestral recombination graph; that is, a directed acyclic graph with branch lengths, in which most nodes have a single parent, but some have two parents and these are the recombinants. The second step simulates the characters in the alignment. At each of the nodes with two parents, part of the sequence is inherited from one parent and the remainder from the other parent. The number of recombinant nodes is random and controlled by a recombination rate parameter. As the parameter increases the number of expected recombinations increases. Because of the framework, it is possible that the alignment may not have detectable signs of non-treelike behaviour, even when a non-zero recombination rate is used.

The simulator uses four parameters. The recombination rate determines whether the data is simulated on a tree (a recombination of zero) or on a network (a non-zero recombination rate). The other three parameters are the mutation rate or sequence divergence rate, the number of taxa, and the sequence length.

We used our own implementation of Hudson's algorithm.

2.3 Methodology

The particular regression approach we took is the **least absolute shrinkage and selection operator algorithm**, or LASSO (Tibshirani, 1996). It applies a tuning parameter restriction to the full least squares solution, allowing regression to be carried out in a way that subsets the data by including variables a few at a time. Like ridge regression (Hoerl and Kennard, 1970), it reduces the variance of the parameter estimates by adding small amounts of bias.

For each value of the tuning parameter λ, LASSO finds a vector $\boldsymbol{\beta}$ that minimises $||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||$ such that $\mathbf{1}^T\boldsymbol{\beta} \leq \lambda$. Different sets of solutions are constructed by varying λ, adding and removing variables as required to satisfy the constraints.

We also investigated an NNLS-LASSO hybrid. The LASSO algorithm picks a suite of subsets of variables subject to the constraint $\mathbf{1}^T\boldsymbol{\beta} \leq \lambda$, as λ varies. For the hybrid approach we use the LASSO algorithm to determine the subset of variables but the final $\boldsymbol{\beta}$ coefficients are calculated for the subset using non-negative least squares (NNLS).

One practical advantage of the LASSO framework is that the set of solutions for varying λ can be computed efficiently. The first efficient algorithm was due to Osborne et al. (2000a) and Osborne et al. (2000b). Efron et al. (2004) showed that LASSO solutions can be computed using a variation of their LARS algorithm.

The LASSO solutions do not provide a single model but rather a set of solutions from which an optimal model is chosen. Two popular criteria for making such decisions are the Akaike Information Criteria (AIC) (Akaike, 1974) and the Bayesian Information Criteria (BIC) (Schwarz, 1978).

The model we chose is the one with the smallest AIC or BIC because it has low residuals given the number of parameters in the model. The information criterion approach provided a way to objectively balance the desire for fewer splits with the desire for the distances to be modelled well.

The AIC criterion in this context is given by

$$AIC = \frac{||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||^2}{n\sigma^2} + \frac{2}{n}k \tag{2.8}$$

and the BIC is given by

$$BIC = \frac{||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||^2}{n\sigma^2} - k\log(n). \tag{2.9}$$

The vector $\mathbf{y}$ is the vector of pairwise distances, k is the number of parameters in the model, and n is the total number of splits (in this case number of pairs). The parameter σ^2 is the variance of the predictor error vector, a fixed but unknown constant that needs to be estimated. Smaller σ^2 values favour larger models; that is, models with a greater number of parameters.

We estimate σ^2 using the ordinary least squares estimator (Rao, 1970). While the estimators of the covariance matrix above contain estimators of the predictor error vector on the diagonal, we chose to use standard estimators. Further investigations into estimators for σ^2 are presented in Section 3.2.3.

The σ^2 estimator from the OLS framework is

$$\hat{\sigma}^2 = \frac{RSS_{fin}}{n-k} \tag{2.10}$$

where RSS_{fin} is the residuals sum of squares of the final model when the parameters are estimated using an ordinary least squares approach. This is how σ^2 is estimated in a linear regression framework.

An important issue is which model should be used when σ^2 is estimated. One could estimate σ^2 based on $\boldsymbol{\beta}$ values from the full network, or from a tree, or according to some other objective. Throughout this thesis we use $\hat{\sigma}^2_T$ to describe σ^2 estimated under the tree-based null model, and $\hat{\sigma}^2_N$ to describe σ^2 estimated under the network-based null model. Note that, in general, $\hat{\sigma}^2_N \leq \hat{\sigma}^2_T$ so choosing a tree-based null model will in general, favour selection of smaller models than choosing a network-based null model.

The difference between the estimators $\hat{\sigma}^2_T$ and $\hat{\sigma}^2_N$ is the set of residuals used in calculating $\hat{\sigma}^2$. For $\hat{\sigma}^2_N$ the residuals come from fitting a non-negative least squares model to the set of circular splits, and for $\hat{\sigma}^2_T$ the residuals come from fitting a neighbor-joining tree (Saitou and Nei, 1987). Below, in Section 2.4.2, we quantify the difference this makes; later, we discuss this in more detail.

It is worth noting that the corrected AIC may be useful in future investigations. While n is large, the ratio of n/k is not and therefore according to Burnham and Anderson (2004) the AIC may lead to over fitted models.

2.4 Experiments

In this section we discuss two experiments. The first experiment investigates the covariance matrix and the second experiment looks at neighbor-net networks as estimated by the LASSO.

2.4.1 An investigation into the covariance matrices

The ideal covariance matrix estimator has three desirable properties. First, it should minimise or eliminate the correlation in the distance vector; second it should be positive definite; and third, it should be well-conditioned. While we investigate these three properties separately we could have chosen to analyse them simultaneously and optimise them using an iterative method.

Reducing correlation

We tested whether the data can be made correlation free by measuring the empirical levels of correlations. We tested both of the covariance estimators reviewed in Section 2.2.

To test which transformation reduced the correlations the most, we carried out the following procedure.

1. Set the parameters and generate an ancestral recombination graph.
   (a) For 100 replications:
      i. Simulate an alignment of 1000 base pairs on the ancestral recombination graph.
      ii. Compute the distances.
      iii. Calculate the two covariance matrix estimators.
      iv. If the matrix is positive definite, then:
         A. Transform the distances using each covariance matrix estimate.
         B. Store the distance vector before and after transformations.
   (b) For the matrix of up to 100 stored distance vectors, calculate the correlation matrix (before and after transformations). Correlations greater than 0.1 are considered significant.

The results of this experiment are summarised in Table 2.1. Note that in some cases the covariance matrices estimated by Equation (2.3) are not positive definite and this gives rise to the differences in the 'before' column.

The transformation calculated using the formula in Equation (2.3) increased the amount of correlation between the distances and is therefore not suitable as a transformation estimator. This estimator of the covariance matrix made very few assumptions, and while the estimator is asymptotically unbiased, it seems that for short sequences the estimator is not effective.

Recombination rate	Sequence Divergence	Equation (2.3) Distances		Equation (2.4) Distances	
		Before	After	Before	After
0	0.01	0.51	0.76	0.54	0.21
0	0.05	0.52	0.85	0.55	0.14
0	0.1	0.53	0.82	0.55	0.13
4	0.01	0.51	0.77	0.55	0.23
4	0.05	0.54	0.89	0.55	0.23
4	0.1	0.55	0.90	0.58	0.14

Table 2.1: Average proportion of significant correlations in the pairwise distances, before and after transformation, sequence length 1000.

The transformation estimator based on the shared path lengths from Equation (2.4) reduced the correlation in the distances. Before transformation, over half of the correlations were statistically significant. After transformation, this number was much lower, ranging from 13% to 23%.

Based on our investigation, the formula based on shared path length outperformed the general formula.

Positive definiteness and numerical stability

In all of the experiments, we found that the shared path covariance estimation (Equation (2.4)) gave positive definite matrices. However, positive definiteness does not guarantee numerical stability.

The condition number is the ratio of the smallest to largest eigenvalues, and it indicates whether the inverse of the matrix is likely to be stable.

In our experiments, the condition number of the covariance matrix before stabilisation was on the order of 100,000. This means there was a difference of six orders of magni-

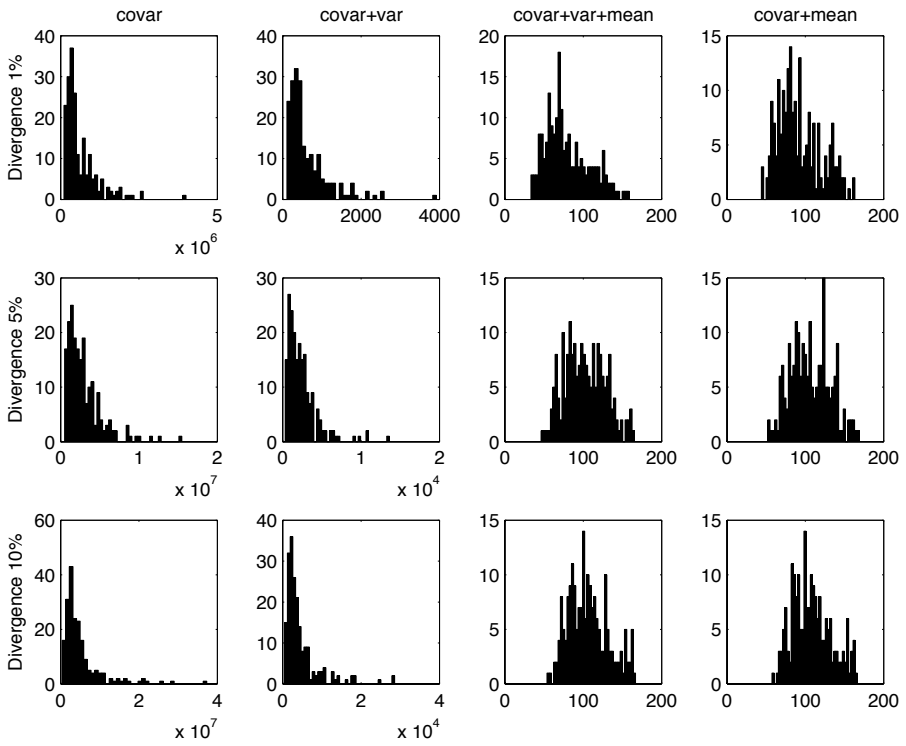

Figure 2.2: Histograms of the condition numbers when the recombination rate is zero, sequence length 1000. 'covar' is without stabilisation, 'covar + var' is the stabilisation by the variance matrix, 'covar + var + mean' is the stabilisation using the variance matrix and the mean of the variances, 'covar +mean' is the stabilisation using the mean of the variances.

tude between the largest and the smallest eigenvalue. This posed a considerable risk of instability.

The first method of stabilisation we tested was adding the variance matrix to the covariance matrix and dividing all the elements by two. This scenario is like fitting a mixture model with equal weights on a full covariance transformation and weighted least squares, where the weights are the variances. This assisted considerably in stablising the matrix, leading to condition numbers of the order of 1,000 rather than 100,000. However, this still posed a risk of numerical issues, so we investigated two further options.

The second method of stabilisation we tested was adding the variance and the mean of the variances to the diagonal elements of the matrix and dividing all the elements by three. This scenario is like fitting a mixture model with equal weights on a full transformation, weighted least squares, and ordinary least squares. This covariance transformation

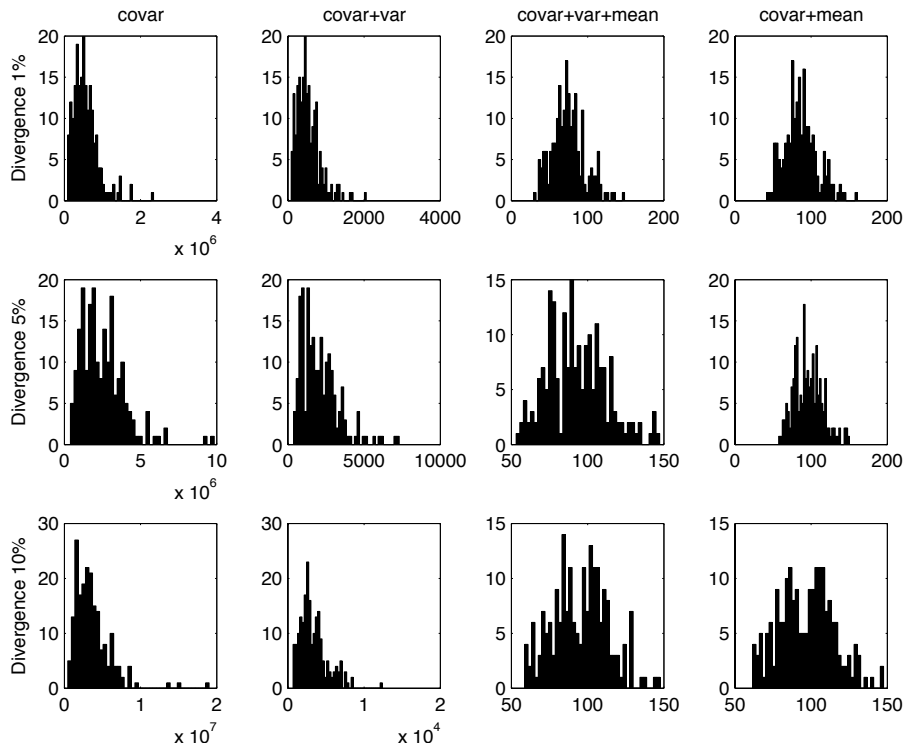

Figure 2.3: Histograms of the condition numbers when the recombination rate is four, sequence length 1000. 'covar' is without stabilisation, 'covar + var' is the stabilisation by the variance matrix, 'covar + var + mean' is the stabilisation using the variance matrix and the mean of the variances, 'covar +mean' is the stabilisation using the mean of the variances.

reduced the extreme differences in the diagonal elements. This assisted considerably in stablising the matrix, leading to condition numbers of the order of 100, which is acceptable as it poses minimal risk of numerical stability issues.

The third method of stabilisation we tested was very similar to the second, and involved adding the mean of the variances to the diagonal elements of the matrix and dividing all the elements by two. This also reduced the extreme differences in the diagonal elements, and led to condition numbers of the order of 100 which is acceptable.

Therefore, we chose to apply the simpler of the two most effective stabilisation methods, where the mean of the variances was added to the diagonal and all the elements of the matrix divided by two. This is in line with the recommended methods of stabilisation discussed in Schäfer and Strimmer (2005).

In summary, the chosen covariance estimator is the formula based on the shared path

Equation (2.4) stablised by the addition of the mean of the variances to the diagonal.

2.4.2 Assessing the effectiveness of the LASSO approach

We term the neighbor-net network with the β_i values chosen by the LASSO algorithm the **reduced neighbor-net**. The desired properties of the reduced neighbor-net network are, first, a close fit of the distance vector, and second, a reduction of the number of splits.

Ideally, the trade-off between fit and the reduction in the number of parameters should result in only a small compromise in the fit.

The measure of the bias of the fits is

$$\text{difference in fit} = \sum_{ij} \frac{d_{ij} - \hat{d}_{ij}}{d_{ij}} \tag{2.11}$$

where d is the pairwise distances and $\hat{d}$ is the modelled pairwise distances. Ideally this difference should be very small. We also investigate the sum of the absolute difference in fits.

The measure we used to study the reduction in the number of splits was a comparison of the number of splits in the reduced neighbor-net network with the number of splits in the original neighbor-net network (with split weight estimated using non-negative least squares).

The simulation study follows that of Wiuf et al. (2001). Trees were generated according to the coalescent model of Hudson (1983). In all cases the number of taxa used was 20. The sequence lengths used were 500, 1000, and 2000 base pairs. The recombination parameter values used were zero, two, four, and eight. Higher recombination parameters should give rise to less tree-like data. The expected sequence divergence rates (sometimes referred to as divergence rate) were one, five, and ten percent site differences. All sets of simulations had 1000 replications.

See Appendices A, B, D and E for graphical results.

The general observation was that the bias in the fit of the distances:

- decreased as the recombination rate increased;
- decreased as the divergence rate increased;
- decreased as the sequence length increased;
- was closer for AIC than BIC; and
- was closer for when using $\hat{\sigma}_N^2$ rather than $\hat{\sigma}_T^2$ as an estimator.

The untransformed scenario fits were small; most were within 1% with $\hat{\sigma}_N^2$ and within 2% for $\hat{\sigma}_T^2$. The transformed scenario fits were very large (up to 20%), especially for BIC based measures and $\hat{\sigma}_N^2$.

The most significant factor in determining the fit of the distances was whether a transform was applied. When a transform was applied, the variation in the fits was much larger. This could be the result of a small number of data sets having unstable transformation matrices. While the simulations above suggested that the condition number was significantly reduced by the stabilisation procedure, it is possible that this stabilisation was not sufficient for some data sets. It could also be the case that the covariance matrix occasionally poorly represented the true correlation, and that this influenced the fitted model. Based on the fit of distances alone, transforming does not appear to give reliable results.

For the rest of the discussion on the fit of the distances we refer to results based on the simulations without transformation.

When $\hat{\sigma}_N^2$ was applied rather than $\hat{\sigma}_T^2$, the variance of fits was much lower. This was expected as the network models will, on average, have more parameters and therefore a closer fit to the distances. The more important question is whether this trade-off (increased variance in the fits for a reduction in parameters) is justifiable. From the perspective of the fit of the distances it seems that the increase in unfitted distances was low and remained acceptable.

The BIC led to a greater variation in the fits compared with the AIC. This was also to be expected, as the BIC has a larger penalty term, potentially leading to smaller models. The BIC will compromise some of the fit of the distances for the reduction in the number of parameters. Once again the more important question is whether this trade-off is justifiable. From the perspective of the fit of the distances it seems that the increase

was low and remained acceptable.

Therefore, the three factors we have control over in an application do influence the variance of the fitted distances. Applying a transformation increases the variance substantially and is therefore unlikely to be justifiable, while the use of $\hat{\sigma}_T^2$ and the BIC may be justifiable, depending in the reduction in the number of splits.

Of the factors we **cannot** control in an application, (sequence length, sequence divergence rate, and recombination rate), the most influential factor is the recombination rate. As the amount of recombination increased, the variance in the fits also increased. This implied that the neighbor-net networks over-fitted the distances to a larger extent as recombination increased. This could be because the distance measure itself was not capturing all the evolutionary aspects; or because the set of circular splits did not contain sufficient flexibility. Sequence length and sequence divergence both had very minor influences on the variance of the fits.

This measure of fit reflects only the total fit rather than the fits of the individual pairwise distances. As we ran 1000 replications, it was not feasible to look at the fits of the pairwise distances for each of the models. We looked at a few fits of the individual distances; see Figure 2.4 for an example based on the AIC criteria and σ^2 estimated by $\hat{\sigma}_T^2$. It was based on ten taxa, a sequence length of 1000 base pairs, a recombination rate of zero, and a sequence divergence rate of 10%.

The measure of fit showed two clear groupings. The first group was shorter distances which were overestimated, and the second group was larger distances generally were underestimated. This is worth further investigation.

We compared the plots of the sum of the absolute difference of the fits is

$$\text{difference in fit} = |d_{ij} - \hat{d}_{ij}| \tag{2.12}$$

and ideally this difference should not be too big.

The general observation is that the sum of the absolute difference of the fits:

- increased when the sequence divergence rate increase;
- remained unchanged with sequence length;

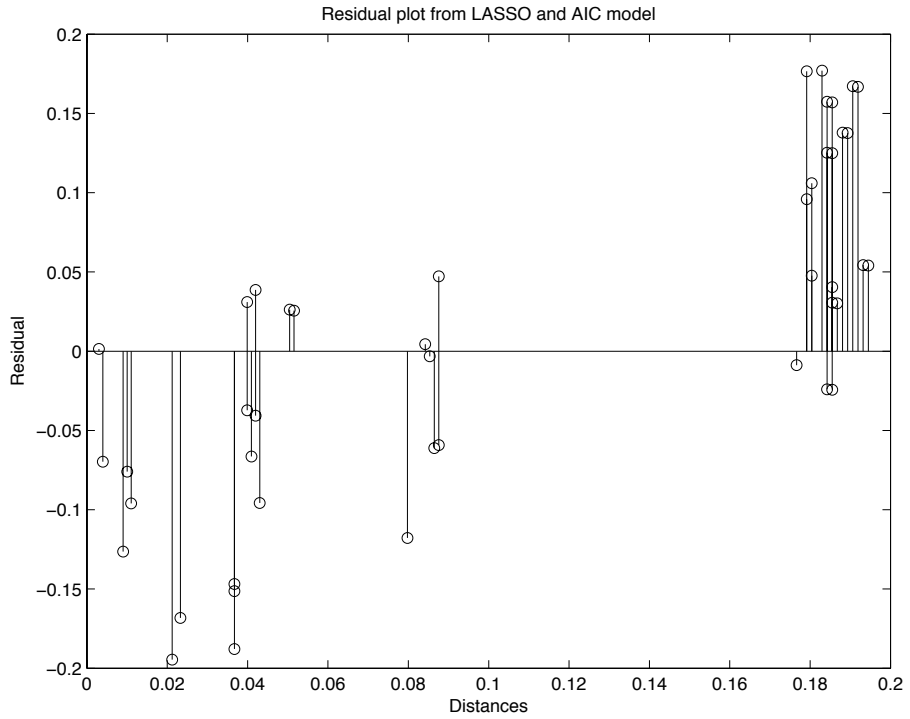

Figure 2.4: A plot of pairwise distance against residual (pairwise distance-fitted pairwise distance) based on fitting a network using circular splits and the LASSO algorithm. Based on ten taxa, a sequence 1000 base pairs long, a recombination rate of zero, a sequence divergence rate of 10%.

- increased when the recombination rate increased;
- remained basically unchanged when $\hat{\sigma}_T^2$ was used instead of $\hat{\sigma}_N^2$; and
- increased when the transformed scenario was used rather than the untransformed scenario.

Therefore, the three factors we have control over in an application do influence the sum of the absolute differences between the fitted distances and the observed distances. Applying a transformation increases the different in the sum of the absolute fit of the distances, while the use of $\hat{\sigma}_T^2$ and BIC make little difference. Therefore subject to the effect on the number of splits the use of $\hat{\sigma}_T^2$ and BIC would seem appropriate.

Of the factors we **cannot** control in an application, (sequence length, sequence divergence rate, and recombination rate), the most influential factor is the recombination rate. As the amount of recombination increased, the sum of the absolute difference in the fits

also increased. This implied that the neighbor-net networks over-fitted the distances to a larger extent as recombination increased. Once again, sequence length and sequence divergence both had very minor influences.

The general observation is that the number of splits chosen relative to the neighbor-net model decreased when:

- the sequence divergence rate decreased;
- the sequence length decreased;
- the recombination rate increased; and
- $\hat{\sigma}_T^2$ was used instead of $\hat{\sigma}_N^2$.

The difference between the number of splits chosen with the AIC and the BIC criteria is discussed below.

The number of splits removed when the transformation was applied was considerably fewer than when there was no transformation. Often, the reduced neighbor-net networks and the original neighbor-net networks were the same. As using the transform significantly increased the variance in the fitted distances, it seems that any reduction in the number of splits came at a high price. These two observations seem at odds with one another; however, the AIC and estimators of σ^2 were calculated while the data was transformed which led to models which did not reduce the number of splits by much. The β_i's of the chosen model, and consequently $\mathbf{X}\boldsymbol{\beta}$, were compared to the original distances, and often this fit was very poor. Therefore, we do not recommend the transformation.

Using $\hat{\sigma}^2 = \hat{\sigma}_T^2$ in the AIC formula trimmed more variables from the split network than $\hat{\sigma}_N^2$. This was expected as it decreases the likelihood component of the AIC formula, thereby favouring smaller models. As we aimed to reduce the level of clutter in the networks, smaller models were preferable unless they were too small to display the key features of the split network. Working with some examples suggested that $\hat{\sigma}_T^2$ was an appropriate choice, and that the reduction in the number of splits chosen was not excessive. This was particularly appropriate here, since tree-like evolution is customarily the standard null hypothesis in phylogenetic inference.

Of the factors we did not have control over, that is, sequence divergence rate, sequence length and recombination rate, all influenced the number of splits in the reduced neighbor-

net networks. The first two of these had counter-intuitave effects, but they both suggested that when there is little information in the alignment (because it is shorter or contains fewer variable sites), the reduced neighbor-net networks are smaller than the original neighbor-net network. For an increase in the recombination rate the reduction in the number of splits was greater, which suggested that much simpler models can be used to capture the key features of the split networks.

The general observation, which was in line with expectation, was that the number of splits chosen by AIC was greater than or equal to the number of splits chosen by the BIC. The difference between the numbers of splits selected with the two criteria was, on average, larger when

- the divergence rate was higher;
- the sequence length was longer;
- the recombination rates was higher; and
- $\hat{\sigma}_T^2$ was used instead of $\hat{\sigma}_N^2$.

The number of splits chosen by BIC can be up to 20 or more fewer than the AIC. It seems unlikely that this would remove too many splits from the network; however, we decided to implement the procedure with the option for the user to choose between the AIC and the BIC. The BIC was the default option.

When the recombination rate is zero, the data is tree-like and we know that the number of true splits is $2n-3$. We compared the number of splits chosen by the LASSO approach for tree-like data. We found the number of splits chosen was closer to the correct number when σ_T was used compared with σ_N and when the BIC was used compared with AIC. See Appendix C for graphical results.

Therefore the implementation used the untransformed scenario, the $\hat{\sigma}_T^2$ estimate for $\hat{\sigma}$, and left the user to choose between the AIC and the BIC.

2.5 Data Analysis

We re-analysed a data set of 135 human mitochondrial sequences studied by Vigilant et al. (1991). A phylogeny for these sequences was used as supporting evidence for an African origin of humans. The validity of this study was later questioned. An extensive study of the large-scale landscape of the space of trees by Penny et al. (1995) indicates that the data support the phylogenetic hypotheses put forward by Vigilant et al. (1991).

This data set has a large number of sequences and a small number of sites, and many sites are known to be fast-evolving. These fast-evolving sites often appear non-tree-like and can bias a tree-based analysis. Therefore, it is interesting to look at a network for the distances to get some idea of the noise and conflicting signals.

Following Penny et al. (1995) we estimated distances from the mitochondrial sequences using K2P + Γ. We compared the models using $\hat{\sigma}^2_N$ and $\hat{\sigma}^2_T$ values and the AIC and BIC criteria.

The networks are shown in Figure 2.5.

We found with $\hat{\sigma}^2_N$ there were 318 splits under the AIC criteria, coinciding with the original neighbor-net model, and 242 under the BIC criteria. With $\hat{\sigma}^2_T$, there were 247 splits under the AIC criteria and 188 under the BIC criteria. The original neighbor-net model also had 318 splits.

One of the key differences between the intermediate model and the final models as chosen by the AIC and the BIC was that the intermediate model did not have many 'trivial' splits; that is those splits which separate one taxa from the remaining taxa. We examine this issue further in Chapter 4.

The visual appearance of the three displayed final models was virtually the same. This suggests that the additional splits were small and did not contribute to the overall appearance of the network.

The network showed a reasonable amount of deviation from a tree like pattern. This implies that tree-based methods should be used with caution and that some conclusions from a tree based analysis could be artifacts of noise and fast evolving sites.

The fit of the distances was not close until quite late in LASSO algorithm. Therefore,

there was a comprise in the fit for the models with fewer splits, but this did not alter the visual appearance of the network by much. See Figures 2.5 and 2.6.

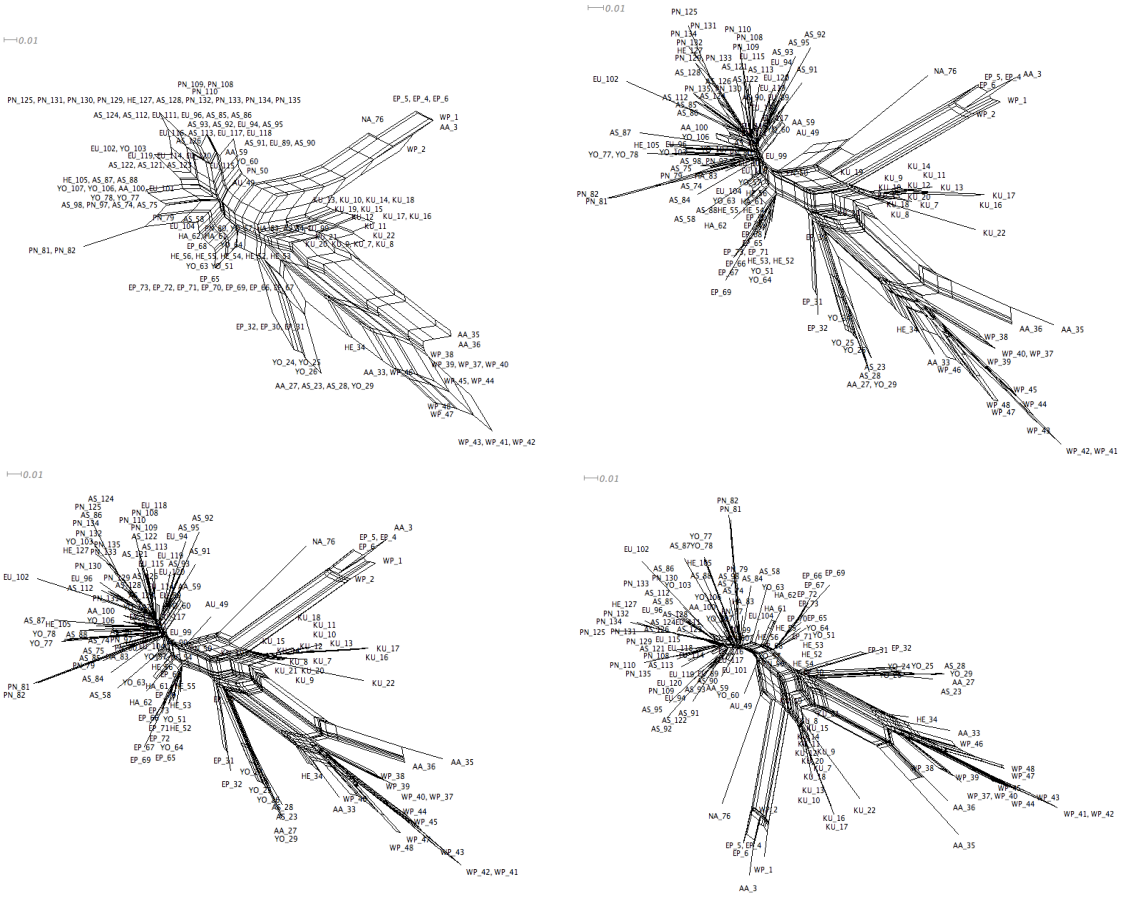

Figure 2.5: Top left: An intermediate network of 60 splits. Top right: The LASSO network for 135 Human mitochondrial sequences given by the $\hat{\sigma}^2_T$ and the BIC, 188 splits. Bottom left: The LASSO network for 135 Human mitochondrial sequences given by the $\hat{\sigma}^2_N$ and the BIC, 242 splits, Bottom right: The original neighbor-net network and LASSO network given by the $\hat{\sigma}^2_N$ and the AIC, 318 splits.

2.6 Discussion

We applied the tools of linear regression to the problem of estimating weights in a split network. This involved investigating two estimates of covariance between distances, and some methods for estimating $\hat{\sigma}^2$ used for the information criteria. In the end, a trans-

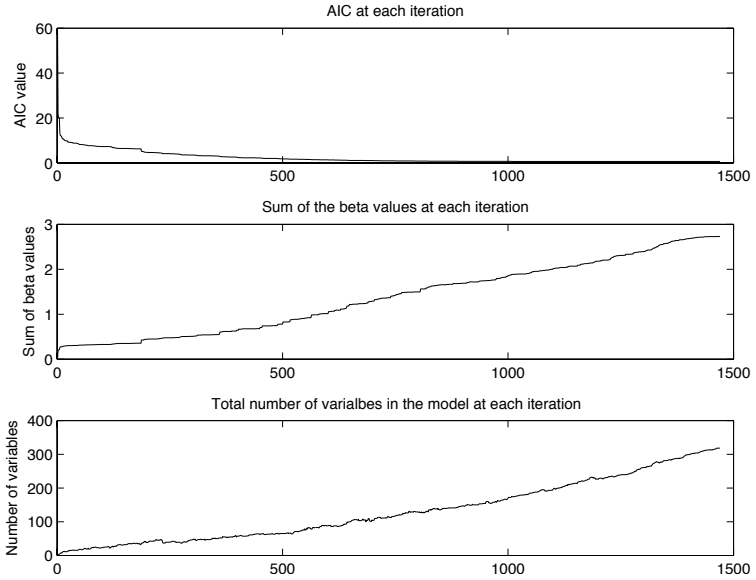

Figure 2.6: Summary statistics based on fitting a network using the LASSO to 135 Human mitochondrial. AIC, sum of beta values and total number of variables in the model for each iteration of the LASSO.

formation seemed undesirable. The chosen method for estimating σ^2 was based on the null model of a tree. This seemed to better reflect the variance of $\mathbf{y}$, and lead to smaller models. However, the apparent objectivity of the information criterion was undermined by this subjective choice of σ^2 estimate.

Our application showed that the benefit of introducing LASSO for the calculation of the split weights was the reduction in the number of splits. The $\hat{\sigma}_T^2$ and the BIC network maintained all the key features of the network and had only 60% of the original splits.

3

A failed test for 'tree-likeness'

This chapter focusses on developing a test for phylogenetic heterogeneity, regardless of its source. Unfortunately, our test was not successful, but some of the investigations into the sources of the inefficacy of the test are of interest.

Not all evolution is tree-like. Phenomena like lateral gene transfer, hybridisation, and recombination, all led to non-tree-like evolution.

Lateral gene transfer, or horizontal gene transfer, is where one strand of DNA has inherited material from a source that is not its parent. There are methods which specifically look at testing for, detecting, and displaying lateral gene transfer, such as the recently published methods of Than et al. (2007); Abby et al. (2010); Cohen and Pupko (2010); Boc et al. (2010). Also see Becq et al. (2010) for a comparative study of 16 methods of lateral gene transfer detection.

Hybridisation is common in plants as polyploidy (having more than two copies of each chromosome) can produce viable offspring. See Soltis and Soltis (2009) for a review on

plant hybridisation. There are methods which specifically look at testing for, detecting and displaying hybridisation such as the recently published methods of Holland et al. (2008); Joly et al. (2009); Chen and Wang (2010).

A third cause of sequence heterogeneity is recombination. Recombination and recombination detection methods are discussed at length in Chapter 7.

All of the methods for detecting phenomena like lateral gene transfer, hybridisation, and recombination have very specific alternative hypotheses. They are based on the assumption that the data is either tree-like or non-tree-like because of a specific type of event in the evolutionary history.

Biologists also want more general tests where the null hypothesis is that the data is tree-like and the alternative hypothesis is that the data is non-tree-like. This alternative hypothesis makes no reference to a specific cause of the non-tree-like behaviour.

The demand for such tests arises from the underlying assumption of many phylogenetic methods that evolution is tree-like. An all-encompassing test would provide a useful initial screening tool. A rejection of the null would provide an opportunity to investigate the specific cause of non-tree-like evolution.

Only a few such tests exist: Bulmer (1991) fitted a tree to the matrix of pairwise distances and then tested the residual sum of squares using a chi-squared distribution. Goldman (1993) investigated whether data evolved in a tree-like manner using maximum likelihood and likelihood ratio tests. Lyons-Weiler et al. (1996) developed Relative Apparent Synapomorphy Analysis (RASA), a method for statistically detecting phylogenetic signal. A failure to detect phylogenetic signal may indicate non-tree-likeness. Makarenkov and Legendre (2004) considered trees fitted by least squares and then added another branch to the tree, such that it reduced the value of the least squares fit the most. If adding a branch improved the fit, then this was evidence that the data was not tree-like.

In the absence of a range of effective tests, several authors have developed methods of representing the data in such a way as to indicate non-tree-like behaviour. The first two of these methods display features of the data which show how resolved subsets of taxa are.

Strimmer and von Haeseler (1997) used quartets to investigate tree-likeness. They assigned a probability to each of the three resolved topologies on four taxa using the ratio

of the likelihood for the ith model divided by the total of the three likelihoods. The quartets were plotted on a triangle of probability space. Points near the center implied an inability to distinguish between the topologies, while points in the corners showed strong support for one of the trees. A plot of all quartets showed whether the quartets were well resolved.

Holland et al. (2002) introduced quartet based delta plots. In this method, the plot is based on the assumption that estimated pairwise distances from tree-like data will satisfy the four-point condition that the two larger split based distances are equal; that is, the larger two of $d_{xy|uv}, d_{xu|yv}$ and $d_{xv|yu}$ are equal where x, y, v and u are taxa of quartet q and $xy|uz$ denotes the split with x and y on one side and u and v on the other. The δ score measures the deviation from the four point condition, and is given by

$$\delta_q = \frac{d_{xv|yu} - d_{xu|yv}}{d_{xv|yu} - d_{xy|uv}} \tag{3.1}$$

where $d_{xy|uv} \leq d_{xu|yv} \leq d_{xv|yu}$. A δ score of zero indicates that the four-point condition is satisfied. One measurement used to assess tree-likeness is the average δ score for all quartets containing that taxa. High average scores indicate potential recombinants.

Both methods visualise some aspect of the data which indicates tree-likeness. With larger taxa sets these quartet methods of Strimmer and von Haeseler (1997) and Holland et al. (2002) will struggle, and usually in these situations a random subset of quartets is analysed.

The other class of representations is networks. For an overview of phylogenetic networks see Huson et al. (2010).

In this chapter we discuss how we developed another general test for non-tree-like evolution. The test we developed compared an information criterion on a tree based model (neighbor-joining) with an information criterion on a network (neighbor-net). The AIC as an information criterion is appropriate for comparing non-nested models, and therefore it can be used to compare a network and a tree even if the splits of the tree are not contained within the network. If the models are nested, then the F-test is also appropriate.

This chapter is arranged as follows. The first section describes the failed test. The second section contains the simulation setup investigating the power and α-level of the test, and

the third section contains the results of the simulation.

We investigated the test to find the sources of its inefficacy. We investigated the AIC criterion, the model of recombination, and the effect of increasing the power.

3.1 An information criterion approach to testing for tree-likeness

We investigate a statistical test for testing tree-likeness based on comparing information in networks and trees.

The procedure had three steps. First, we estimated a tree using neighbor-joining and a splits network using neighbor-net and the LASSO (see Chapter 2). Second, we compared the information criteria on both the tree and the network. Third, we rejected the null hypothesis that the sequences evolved under tree-like evolution if the information criteria for the network was significantly lower.

The splits of the neighbor joining tree were estimated using the MATLAB functions of Cai et al. (2005). The β_i's were based on the non-negative least squares solution given the neighbor-joining topology. The splits wieghts in the neighbor-net network were estimated using the LASSO-NNLS hybrid as described in Section 2.3. We used untransformed and transformed distances and splits.

Suitable thresholds for deciding there is a significant difference between the information in the network and the information in the tree are -2 and -10. A difference of -10 means there is essentially no support for the hypothesis that the model with the higher AIC fits the data better (there is approximately a 1 in 100 chance of the model with the higher AIC being the better model to explain the data). A difference of -2 means there is little support for the model with the higher AIC (Burnham and Anderson, 1998). Small differences mean that both models explain the data well. The cutoff does not depend on the parameters of evolution such as sequence divergence rate, as these are reflected in the AIC values themselves.

We use these cutoff values as an initial guide into values which may be appropriate to use as a cutoff. We had hoped that these cutoff values would give an appropriate level.

The AIC was calculated for both trees and networks with the same σ^2 estimator as defined in Section 2.3. The examiner duly noted that Equation 2.8 differs from the specification of (Burnham and Anderson, 1998) by a factor of n. All of the results are proportional to the results we would have gotten had we use this particular form of the AIC. This is noted for future development of this work.

We investigated the α-level and power of our test through simulation. The α-level was the rate at which the null hypothesis of tree-like evolution is rejected when the data was generated according to tree-like evolution. The power was the ability to reject the null hypothesis when the data was non-treelike.

3.1.1 Assessing power and level by simulation

The simulation study follows that of Wiuf et al. (2001). Sequences were generated according to the coalescent model of Hudson (1983). In all cases the number of taxa used was 20. The sequence lengths used were 500, 1000, and 2000 base pairs. The recombination parameters used were zero, two, four, and eight. The expected sequence divergence rates were one, five, and ten percent.

The distances were calculated using a Jukes-Cantor model (Jukes and Cantor, 1969) with a Wilson adjustment (Wilson, 1927). All sets of simulations had 1000 replications.

3.1.2 Results and discussion

Results for the untransformed scenario

The α-level of the test was the frequency with which the AIC of the network was lower than the AIC of the tree when the input alignment was tree-like. For a cutoff of -2, it ranged from 14% to 57% while for a cutoff of -10, it ranged from 1% to 40%. Often, the level was well over 20%. The ideal level for a test would be 5% to 10%; therefore, this was generally too high.

The two factors we have no control over in an application were sequence divergence rate and sequence length. Curiously, in the majority of cases the alignments with a 5% sequence divergence rate had a higher (poorer) level than those with a 1% sequence

divergence rate or a 10% sequence divergence rate. Increasing the sequence length also increased the level.

The power of the test was the frequency with which the AIC of the network was lower than the AIC of the tree when the input alignment was not tree-like. The power was generally low. The power was between 42% and 78% when the cutoff was -2, and between 18% and 62% when the cutoff was -10. Therefore, as expected, the lower cutoff increased power. The AIC had more power than the BIC. The power sometimes increased and sometimes decreased with an increase in the recombination rate.

The results for power and α-level are summarised in Tables 3.1 and 3.2.

Cutoff	Sequence Length		500			1000			2000		
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	$\rho = 0$	AIC	14	57	34	27	50	27	48	36	22
		BIC	14	54	28	27	46	22	47	31	18
-10	$\rho = 0$	AIC	12	30	4	23	17	1	40	6	1
		BIC	12	29	4	23	17	1	40	6	1

Table 3.1: Level of the test. Percentage of instances in which the AIC difference is greater than the cutoff. Results for sequence lengths 500, 1000, and 2000, sequence divergence rates 1%, 5%, and 10%. AIC and BIC. Untransformed, 20 taxa, 1000 replications.

Cutoff	Sequence Length		500			1000			2000		
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	$\rho = 2$	AIC	43	75	62	65	77	71	78	76	73
		BIC	42	73	60	65	74	68	77	74	70
	$\rho = 4$	AIC	60	71	65	71	73	69	78	77	73
		BIC	59	70	64	70	72	68	77	77	72
	$\rho = 8$	AIC	61	58	55	64	60	56	63	60	57
		BIC	60	57	55	63	60	56	62	60	58
-10	$\rho = 2$	AIC	40	40	26	56	45	33	62	45	38
		BIC	39	40	25	56	44	32	62	44	38
	$\rho = 4$	AIC	50	37	28	55	37	31	56	41	39
		BIC	50	36	27	55	37	31	55	41	39
	$\rho = 8$	AIC	43	22	18	37	21	20	31	24	20
		BIC	43	21	18	37	21	20	30	24	20

Table 3.2: Power of the test. Percentage of instances in which the AIC difference is greater than the cutoff. Results for sequence lengths 500, 1000, and 2000, sequence divergence rates 1%, 5%, and 10%, recombination rates (ρ), 2, 4 and 8 and AIC and BIC. Untransformed, 20 taxa, 1000 replications.

Discussion on the untransformed scenario

While the performance was poor across the range of parameters it is possible to comment on the framework with the more appropriate α-level. With the cutoff at -10 instead of -2 the α-level was closer to that of 5%; therefore, the larger cutoff gave a more acceptable level. The BIC performed marginally better than the AIC, but the difference was inconsequential. Therefore of the factors we can control, we would initially recommend a larger cutoff and the use of AIC criteria.

The power tended to increase as the sequence length increased, but only by a few percentage points. When the cutoff was -2, the sequence divergence rate showed a lack of monotone trend with the sequence divergence rate of 5% often reporting the highest power. When the cutoff was -10, the lowest sequence divergence rate often had the highest power. The non-uniform behaviour of the power with increasing recombination rate was unexpected and is difficult to explain.

The information criterion approach is not suitable as a test for tree-likeness with untransformed data.

Results for the transformed scenario

We repeated the experiment, first transforming the distances and splits to remove the correlation.

The α-level of the test was the frequency with which the AIC of the network was lower than the AIC of the tree when the input alignment was tree-like. The level ranged from 14% to 55% for the cutoff of -2, and 0% and 41% for the cutoff of -10. The ideal α-level for a test would be 5% to 10%; therefore, once again, the level was higher than is recommended for a level of a test when the cutoff was -2, and was sometimes too high when the cutoff was -10. The BIC performed marginally better than the AIC, but the difference was inconsequential.

The two factors we had no control in an application were sequence divergence rate and sequence length. Once again, in the majority of cases the 5% sequence divergence had a higher (poorer) level than either sequence divergence rates of 1% or 10%. Increasing the sequence length increased the level.

The power could either increase or decrease when the recombination rate was increased. The power tended to increase as the sequence lengths increased. Once again, the sequence divergence rate showed a lack of monotone trend when the cutoff was -2, and decreasing power when the cutoff was -10.

See Table 3.3 for the results on the α-level and Table 3.4 for the results on the power.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	$\rho = 0$	AIC	15	55	29	28	47	22	49	37	20
		BIC	15	51	24	27	42	17	48	30	14
-10	$\rho = 0$	AIC	13	26	4	25	14	1	41	5	0
		BIC	13	25	4	25	14	1	41	4	0

Table 3.3: Level of the test. Percentage of instances in which the AIC difference is greater than the cutoff. Results for sequence lengths 500, 1000, and 2000, sequence divergence rates 1%, 5%, and 10%. AIC and BIC. Transformed, 20 taxa, 1000 replications.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	$\rho = 2$	AIC	44	68	57	66	68	62	73	73	65
		BIC	44	66	52	65	65	59	72	70	63
	$\rho = 4$	AIC	57	61	54	68	61	59	69	63	60
		BIC	56	58	51	66	59	56	66	61	57
	$\rho = 8$	AIC	55	41	35	50	42	38	46	44	39
		BIC	55	37	33	48	39	36	43	41	37
-10	$\rho = 2$	AIC	41	36	17	57	33	24	60	39	30
		BIC	41	36	16	57	33	24	60	38	30
	$\rho = 4$	AIC	50	27	17	55	28	23	48	33	25
		BIC	50	27	17	55	27	22	48	32	25
	$\rho = 8$	AIC	45	12	9	32	12	11	23	16	14
		BIC	45	12	9	31	12	11	23	16	14

Table 3.4: Power of the test. Percentage of instances in which the AIC difference is greater than the cutoff. Results for sequence lengths 500, 1000, and 2000, sequence divergence rates 1%, 5%, and 10%, recombination rates 2, 4 and 8 and AIC and BIC. Transformed, 20 taxa, 1000 replications.

Discussion on the transformed scenario

Once again the performance was poor. It is possible to comment on the framework with the more appropriate α-level. With the cutoff at -10 and a high sequence divergence rate, the level was acceptable. With lower sequence divergence rates the level was too high. When the cutoff was -2, the level was always too high. Therefore, -10 appears to be the more appropriate cutoff.

As noted above, the non-monotonic behaviour of the sequence divergence rate is very concerning and because of it we are unable to hypothesize about the behaviour of this test outside the sequence divergence rate parameters tested. Increasing the sequence length also increased the level, another indicator of poor performance.

The power can increase or decrease with an increase in the recombination rate, yet another indicator of poor performance. We expected that the power of the test would increase as the recombination rate increased, and therefore the non-uniform behaviour of the power with increasing recombination rate was unexpected.

Therefore, even with the distances and splits transformed to allow for phylogenetic correlation, the information criterion approach is not suitable as a test for tree-likeness.

3.1.3 Summary of findings

The performance was poor across the range of parameters.

With the cutoff at -10 instead of -2, the α-level was a great deal closer to that of 5%. Therefore, the larger cutoff had a more acceptable performance. The BIC had a better performance than the AIC, but the difference was inconsequential. Therefore, the AIC is an acceptable choice. The transform had a lower, more acceptable α-level, but not much power. The difference is not large enough to justify its use, especially in light of its performance in Chapter 2.

Overall, the information criterion seems an inappropriate way to test for tree-likeness we now investigate why.

3.2 Investigating the AIC criterion

The idea of testing for tree-likeness using the AIC and distance based networks and trees was simple, but ineffective. We were interested in exploring the test further in an effort to understand how our very general test works and the reason(s) for its failure.

We investigated three aspects of the test. Our investigations were into the specific AIC criteria used, the model of recombination used for simulation, and measures to potentially increase the power.

The AIC formula used above came from linear regression. We were interested in whether it was appropriate in this setting.

When we investigated the AIC criteria, we looked at three aspects. The first was the cutoff. Above, we tested two cutoffs, -2 and -10, and in general, neither performed well. We were interested in two aspects of the cutoff; first, the value that gave the correct α-level for each of the simulation sets and second the power when the cutoff was optimised. This will give us an indication of whether a fixed cutoff was a feasible option for this test. The second aspect was the likelihood calculation. We applied a likelihood calculation for a network to determine whether this improved the performance. The third aspect was the estimation of σ^2 a parameter in the AIC formula.

3.2.1 Investigating the AIC cutoffs

In Section 3.1 we reported the results based on two cutoff values, -2 and -10. In this section, we investigated the potential maximum power of our test for tree-likeness. We set the cutoff at a value that gives an appropriate α-level and reported the power, given that cutoff, for each parameter set.

Methodology

We used the data simulated on a tree to determine the approximate cutoff that gives an α-level of 5% for each sequence length and sequence divergence rate combination. We reported the cutoff values and the power for each recombination rate given the cutoff.

The only scenario we investigated was the AIC criterion, and we did not transform the distances.

Results and discussion

We expected that the cutoffs would become closer to a fixed value as the sequence length increased. However, when the sequence divergence rate was low the cutoff values increased as the sequence length increased, and when the sequence divergence rate increased the cutoffs decreased as the sequence length increased. Therefore, they do not approach a common value.

Within a specified sequence length an increase in the sequence divergence rate (and consequently the average proportion of informative sites) caused the cutoff to increase and when the sequence divergence rate was 10% the cutoffs were more than -10. Potentially, this implies that the level may stabilise when the sequence divergence rate is high.

The cutoff values for the sequences which had low sequence divergence rates were very low ranging from -460 to -2360. The implication is that networks fit much better than trees when the sequence divergence rate is low. This may mean that when there are few non-constant sites, there is not enough information to determine tree-like behaviour.

The cutoff values are shown in Table 3.5, and the power results in Table 3.6.

Sequence Length	Sequence Divergence Rate	Cutoff value
500	1%	-461
	5%	-158
	10%	- 8
1000	1%	-1131
	5%	-83
	10 %	- 6
2000	1%	-2357
	5%	-11
	10%	-4

Table 3.5: A table showing the cutoffs, which according to this set of simulations, give the correct level.

Even with the level set correctly, the power was very low. Once again the 5% sequence divergence rate had the lowest power (especially with shorter alignments). Perhaps at this sequence divergence rate there is a minimal ability to discern tree-likeness.

Sequence length had a strong impact on the power of the test when the sequence divergence rate was high. The power almost doubled between the alignments that were 500 base pairs long and the alignments that were 2000 base pairs long.

It is interesting to note that often the highest power was for the recombination rate of four, our medium level of recombination. This implies that data with a larger amount of recombination appears more tree-like than data with some recombination. This is counter-intuitive. The reasons for this should be explored further, but possible explanations include that the distances are more tree-like even though the underlying data generating process is far from tree-like; or an inability to fit the distances using the circular splits.

Sequence Length	Sequence Divergence	Recom. rate		
		2	4	8
500	1%	29.6	40.0	34.8
	5%	9.2	9.6	3.0
	10%	28.8	31.8	22.4
1000	1%	31.4	35.7	20.4
	5%	11.6	8.7	2.8
	10%	44.5	43.0	29.6
2000	1%	24.9	22.2	9.1
	5%	43.1	39.3	22.6
	10%	54.9	56.3	37.0

Table 3.6: A table showing the power (as a percentage) if the α-level is set by ensuring that the cutoff gives the correct level.

3.2.2 Hadamard likelihood approach

One potential cause of the high level might be the use of the linear regression likelihood. Unfortunately few approaches to calculating likelihoods on networks exist; and furthermore, most of them require rooted networks. Should such likelihood calculations be developed, there would be scope for further investigation into using information criteria to test for tree-likeness.

One approach to phylogenetic inference which has a fully developed likelihood is the Hadamard likelihood (Hendy and Penny, 1993). This likelihood calculation uses the information in each site to calculate a likelihood.

We ran a set of simulations on ten taxa with the same simulation framework as above (sequence lengths 500, 1000, and 2000, sequence divergence rates of 1%, 5%, and 10%, and recombination rates of zero, two, four, and eight). We ran 200 replications.

Results and discussion

As the sequence length or sequence divergence rate increased, the fits of the distances were closer. The fits seemed equally good across the four recombination rates. We compared the number of splits in the model chosen by the information criterion to the number of splits displayed in the original neighbor-net network. The Hadamard framework frequently picked models with more splits than the original neighbor-net.

The test for tree-likeness always returned a tree for all recombination rates, sequence lengths, and sequence divergence rates. Therefore, the Hadamard likelihood is an inappropriate choice of method for calculating the likelihood.

The additional figures for the Hadamard likelihood framework can be found in Appendix F. Note that these figures are based on ten taxa (twenty is computationally infeasible due to the time taken to run each replicate).

3.2.3 Bulmer approach to estimating σ^2

Figure 3.1 shows the fitted distance and a set of corresponding AIC curves based on a range of σ^2 values. The AIC curve is very flat once the fits are very close and as such the estimated σ^2 can have a substantial effect on the number of parameters in the final model. Therefore one potential cause of the high level might be that σ^2 was estimated based on standard regression techniques. We therefore used the mean of the variances from the formula of Bulmer (1991) where the variance matrix is based on the shared path. This σ^2 estimator is referred to as $\hat{\sigma}^2_B$.

We ran a set of simulations on twenty taxa with the same simulation framework as above (sequence lengths 500, 1000, and 2000, sequence divergence rates of 1%, 5%, and 10% and recombination rates of zero, two, four and eight.) We ran 200 replications.

For results see Appendix G.

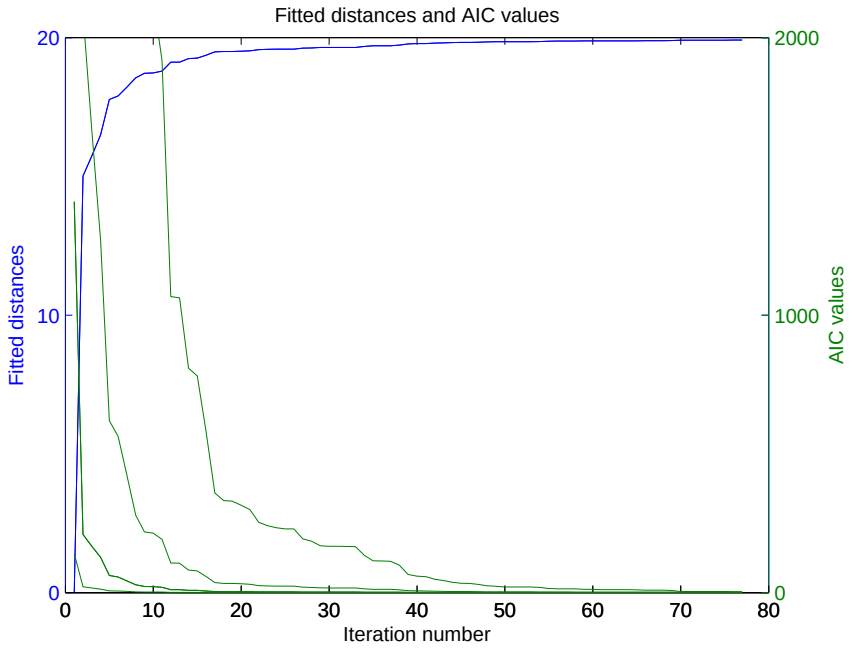

Figure 3.1: A plot of fitted distances (blue) and a range of theoretical AIC curves that vary by the choice of σ^2 (green). The LASSO algorithm increases λ at each iteration and for each iteration we calculate the NNLS coefficients for the set of variable given by the LASSO. The fitted distances is the sum of the β values.

Results and discussion

The fits of the distances were very close, especially for the lowest recombination rate. We compared the number of splits in the model chosen by the information criterion to the number of splits in the original neighbor-net network. Using $\hat{\sigma}^2_B$ trimmed out a lot of variables, such that many of the network models had fewer splits than a phylogenetic tree.

The level using $\hat{\sigma}^2_B$ was always zero; that is, a tree-like data set always returned a tree-like result. The power was very low, and most of the network based alignments were reported to be tree-like.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	ρ=2	0	0	0	0	0	0	0	2.5	3.5
		ρ=4	0	0	0	0	1	2	0	1.5	8
		ρ=8	0	0	0.5	0	0.5	2	0	6	18
-10	AIC	ρ=2	0	0	0	0	0	0	0	0	0
		ρ=4	0	0	0	0	0	0	0	0	0
		ρ=8	0	0	0	0	0	0	0	0	0

Table 3.7: Power of the test. Results for sequence lengths 500, 1000, and 2000, sequence divergence rates 1%, 5%. and 10%, recombination rates 0, 2, 4, and 8 and AIC. σ^2 calculated using the Bulmer variance matrix, untransformed, 20 taxa, 200 replicates.

3.2.4 Summary of findings about the AIC criteria

Our experiment with the AIC cutoffs showed us that our aim of using a single cutoff was unrealistic. To achieve a 5% α-level, the cutoffs varied widely, particularly across the sequence divergence rates. Therefore, one cause of ineffectiveness of this test is the fixed cutoff value. However even allowing for the AIC cutoff to be the theoretically most appropriate value, the power was low.

We carried out two investigations into aspects of the AIC formula: the likelihood calculation, and the estimator of σ^2. Our results using a Hadamard likelihood suggested that the Hadamard approach to testing for tree-likeness was ineffective. This does not imply that the linear regression likelihood is appropriate, but it *is* better than the Hadamard likelihood approach. Our investigation into using a σ^2 estimator based on Bulmer's variance shared path matrix confirmed that the choice of σ^2 estimator played an important role in the success of the reduction in the number of splits, and in the result of the test for tree-likeness. This particular σ^2 estimator removed too many splits and led to a very low power for our test. Therefore, accurately estimating σ^2 remains an open problem.

3.3 Investigating the model of recombination and the error structure

One possibility is the that the problem lies not with the test framework but with the model used to simulate data. When we first investigated our test for tree-likeness, we

used the Hudson model to generate treelike and non-treelike alignments. Here we detail four investigations into recombination and the error structure.

In the first investigation we compared the performance of the Pairwise Homoplasy Index (PHI) test with our test for tree-likeness. One potential reason for the low power of our test might be that the alignments did not have detectable signs of recombination. The PHI test (Bruen et al., 2006) is a test that indicates whether recombination is present. Therefore we used this test to check that our non-treelike alignments were non-treelike and that our treelike alignments are treelike.

In the second investigation, we simulated recombination by merging two treelike alignments together. This ensured the alignments had recombination and the results will indicate the how high the power could be.

In the third investigation, we applied our test to distances with white noise added. We wanted to investigate whether the noise structure arising from estimating distances from an alignment was leading to the poor performance of the test.

In our final investigation, we generated alignments with continuous characters. These had a different error structure from the distances with noise, but not the discrete nature of distance estimated from simulated alignments. This gave us insight into whether the discrete nature of DNA data was contributing to the poor performance of our test for tree-likeness.

3.3.1 Comparison with the PHI test

The PHI test (Bruen et al., 2006) is a test for recombination which is robust to high sequence divergence rates. It should be able to detect recombination in the alignments with non-zero recombination rates. We compared the level and power of our test to the results of conditioning on the p-value of the PHI test.

We ran a set of simulations on twenty taxa with the same simulation framework as above (sequence lengths 500, 1000, and 2000, sequence divergence rates of 1%, 5%, and 10%, and recombination rates of zero, two, four, and eight). We ran 200 replications.

Results and discussion

Cutoff	Sequence Length	500			1000			2000		
	Sequence Divergence	1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	13	55	34	26	52	25	37	39	25
-10	AIC	12	29	2	25	16	25	33	5	0

Table 3.8: Level of the test. Results for sequence lengths 500, 1000, and 2000; sequence divergence rates 1%, 5%, and 10%; recombination rate of zero and AIC. Results conditioned on the result of the PHI test, untransformed, 20 taxa, 200 replicates.

The level did not improve; in fact, under some scenarios, it was worse once we conditioned on the result of the PHI test. With the cutoff set at -2 the level ranged from 13% to 55% and with the cutoff set at -10 the level ranged from 0% to 33%. See Table 3.8 for results.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	$\rho = 2$	100	86	70	82	79	76	88	79	76
		$\rho = 4$	74	75	60	83	74	68	75	75	75
		$\rho = 8$	78	58	56	66	58	63	58	62	55
-10	AIC	$\rho = 2$	79	54	31	62	41	43	73	58	42
		$\rho = 4$	60	41	25	55	39	37	46	45	38
		$\rho = 8$	48	20	19	33	20	21	31	20	25

Table 3.9: Power of the test. Results for sequence lengths 500, 1000 and 2000, sequence divergence rates 1%, 5% and 10%, recombination rates of two, four and eight, and AIC. Results conditioned on the result of the PHI test, untransformed, 20 taxa, 200 replicates.

The power improved considerably, especially for the shorter sequences and the lower sequence divergence rates. Even with the cutoff set at -10 the power was reasonable for some scenarios. The power ranged from 19% to 100%. See Table 3.9 for results.

The increase in power suggests that some of the alignments generated with non-zero recombination did not contain a detectable recombination signal.

3.3.2 Investigating the model of recombination by merging tree-like alignments

An alternative method of generating non-tree-like sequences is to merge two tree-based alignments. The results based on simulating recombinant alignments in this manner can be compared with the results from simulation recombination using the Hudson model. We compared the fit of the distances, number of splits and test for tree-likeness. A recombination rate of zero means the full sequence was simulated on a single tree. Two tree selected at random from the distribution of all resolved binary trees will share, in expectation, approximately 0.125 splits. The distribution is approximately Poisson with parameter λ=0.125 and the approximation improves as the number of taxa is increased (Bryant and Steel, 2009).

We compared 200 replicates under a recombination rate of zero and this 'tree-plus-tree' model. We used the AIC for comparison and did not transform the distances.

Results and Discussion

The fit of the distances is similar for the 'tree-plus-tree' model compared to the Hudson method with higher non-zero recombination rates; this is seen by comparing Figure 3.2 with Figure A.1. The number of splits is often considerably less than the original neighbor-net network, as seen in Figure 3.3.

In considering the information criterion as a test for tree-likeness, we comment only on the power. With a cutoff of -2 the power was low and between 37% and 64%. This was lower than that seen under the highest recombination rates; compare with Table 3.2. The power is low when the cutoff is -10 ranging from 7% to 44%.

This confirmed our finding that, for high levels of recombination, our test for tree-likeness is ineffective.

3.3.3 Distances with Gaussian noise

We wanted to understand whether inadequate modelling of the error structure was contributing to the high level. Therefore, we investigated the performance of the test for

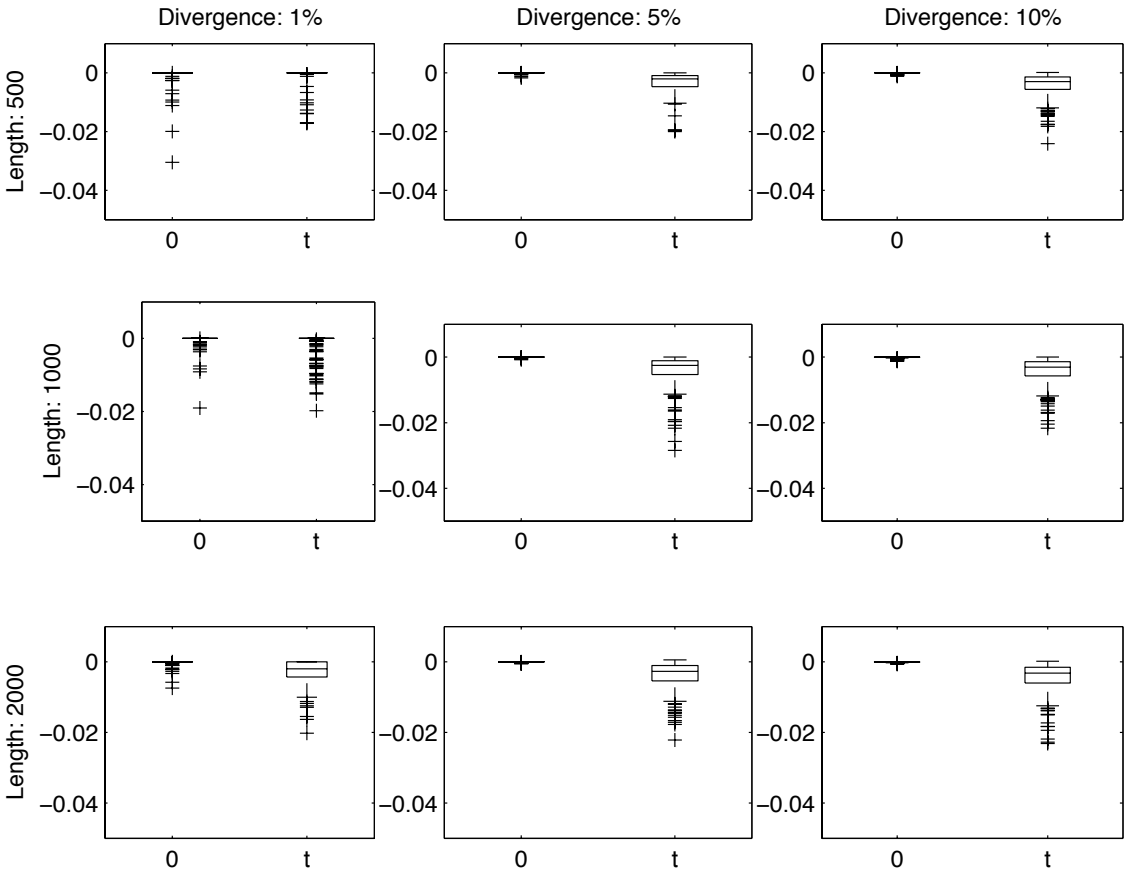

Figure 3.2: Box plots of the fit of the distances for sequence lengths 500, 1000, and 2000 and sequence divergences rate 1%, 5%, and 10%. Each figure contains fits for the recombinations rate zero, and the 'tree-plus-tree' method of recombination ('t'). Untransformed scenario, AIC criterion.

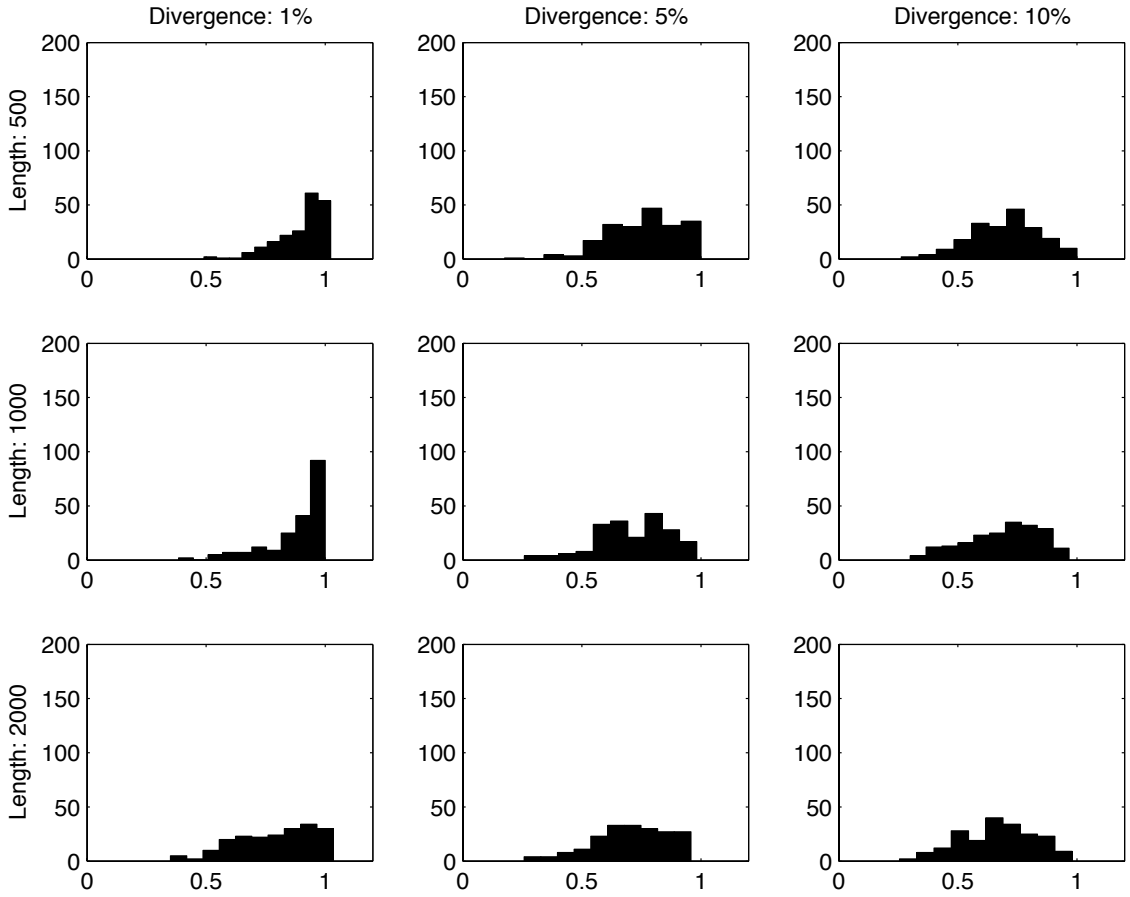

Figure 3.3: Histograms of the number of splits chosen compared to the non-negative least squares solution based on the AIC criterion. The sequence lengths are 500, 1000, and 2000 while the sequence divergences rates are 1%, 5%, and 10%. The recombination model is 'tree-plus-tree'. Untransformed scenario, AIC.

Cutoff	Sequence Length		500			1000			2000		
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	't'	64	37	42	56	41	37	48	42	42
-10	AIC	't'	44	12	12	23	7	7	16	12	13

Table 3.10: Power of the test. Results for sequence lengths 500, 1000, and 2000; sequence divergence rates 1%, 5%, and 10%, recombination rate zero and recombination model 'tree-plus-tree' ('t') and AIC. Untransformed, 20 taxa, 200 replications.

treeness in the presence of only normally distributed noise.

The neighbor-joining and neighbor-net algorithms were based on distances calculated directly from the alignments. In this investigation, we used the modelled distances plus normally distributed noise as the input distance matrix.

The procedure was

1. Calculate the neighbor-joining or neighbor-net split weights using non-negative least squares.
2. Calculate the distance matrix of modelled distances.
3. Generate a noise component for each matrix entry based on a normal distribution.
4. Add the noise to the distances and set any negative distances to zero.

The aim was to see how frequently the algorithm returned a tree given noisy tree-like distances as input, and how often it returned a network, given noisy network-based distances.

Results and Discussion

The fit of the distances was good when the noise was low, and improved as the sequence length and sequence divergence rates increased. When the noise level was high, the fits were not as close. In particular, the short sequences were not fitted well. See Figures 3.4 and 3.5.

The factors which influenced the fits were recombination rate and sequence divergence rate. With sequence data, a higher recombination rate and higher sequence divergence rate both led to poorer fit, while for the distance-plus-noise model, the high sequence

divergence rate had much closer fits and the fits were similar for all the recombination rates.

The results of the test for tree-likeness showed that a cutoff of -2 was adequate for the distances-plus-noise model as the level was 0%. While sequence length did not influence the power, sequence divergence rate did. Specifically, as the sequence divergence rate increased, so did the power. The power also increased as the recombination rate increased. These are the behaviors we expected for this data and for our discrete sequence simulation in Hudson's coalescent model.

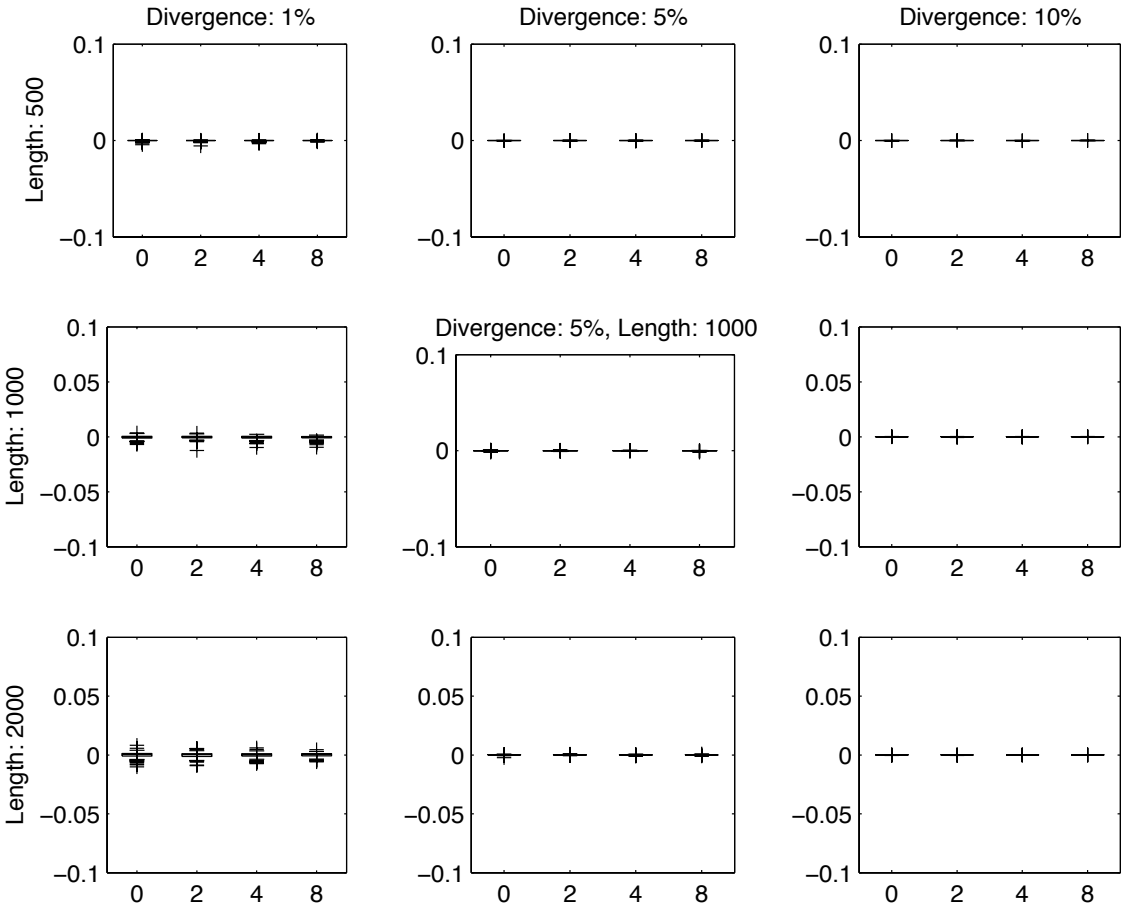

Figure 3.4: Box plots of the fit of the distances for sequence lengths 500, 1000, and 2000, and sequence divergences rate 1%, 5%, and 10%. Each figure contains fits for the recombinations rates zero, two, four, and eight. Noise was added to distances according to a normal distribution with mean zero and standard deviation 0.001. Untransformed scenario, AIC criterion.

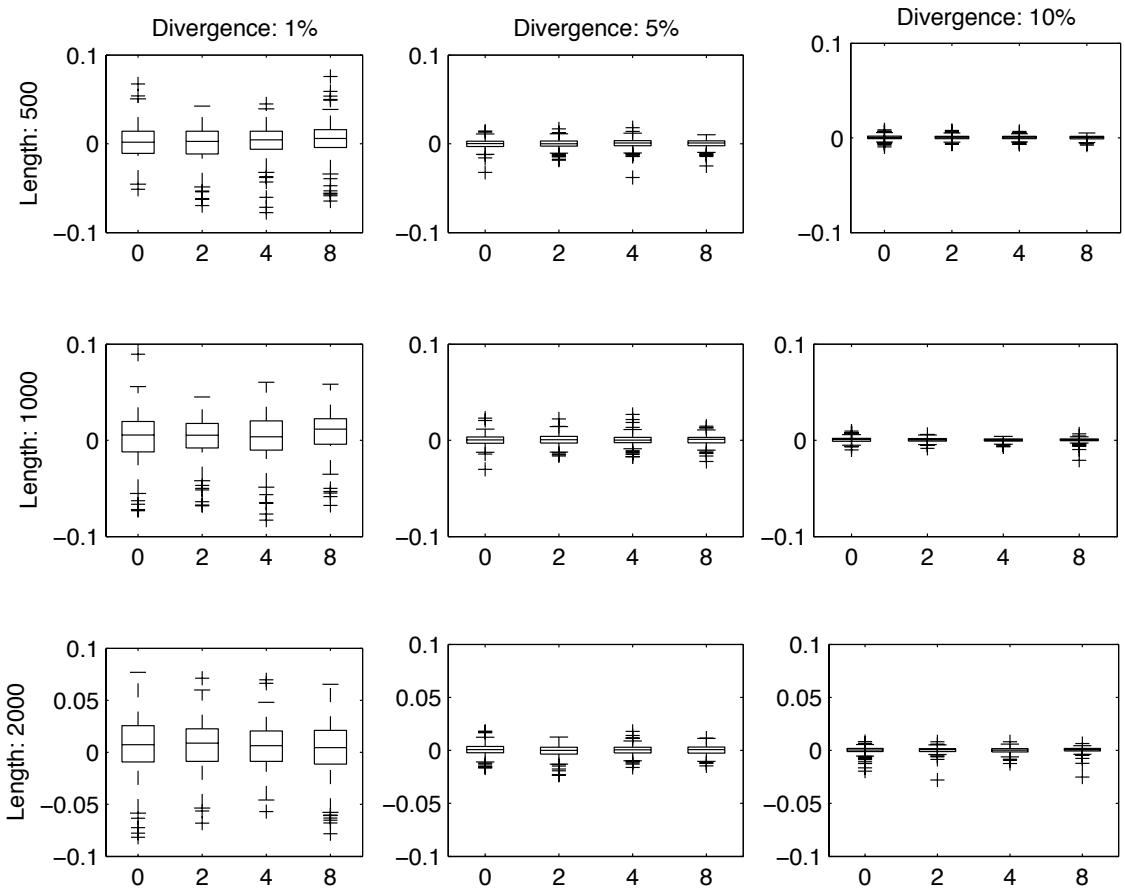

Figure 3.5: Box plots of the fit of the distances for sequence lengths 500, 1000, and 2000, and sequence divergences rate 1%, 5%, and 10%. Each figure contains fits for the recombinations rates zero, two, four, and eight. Noise was added to distances according to a normal distribution with mean zero and standard deviation 0.01. Untransformed scenario, AIC criterion.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	$\rho = 0$	0	0	0	0	0	0	0	0	0
-10	AIC	$\rho = 0$	0	0	0	0	0	0	0	0	0

Table 3.11: Level of the test. Results for sequence lengths 500, 1000, and 2000; sequence divergence rates 1%, 5%, and 10%; recombination rate (ρ) of zero, and AIC. Distances with 0.001 noise, untransformed, 20 taxa, 200 replications.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	$\rho = 2$		2	48	76	2	42	70	1	44	68
	$\rho = 4$		12	68	86	7	66	87	6	67	88
	$\rho = 8$		11	85	97	14	82	98	11	82	95
-10	$\rho = 2$		0	19	42	1	18	45	0	12	45
	$\rho = 4$		1	29	64	1	30	66	1	26	70
	$\rho = 8$		0	48	77	1	51	87	0	44	78

Table 3.12: Power of the test. Results for sequence lengths 500, 1000, and 2000; sequence divergence rates 1%, 5%, and 10%; recombination rates (ρ) 2, 4, and 8, and AIC. Distances with 0.01 noise, untransformed, 20 taxa, 200 replications.

3.3.4 The tree-likeness test on continuous characters

We investigated the performance of the test for tree-likeness on continuously generated characters based on Brownian motion. The distance measure applied to the characters was the Manhattan distance. This is the recommended distance measure for continuous characters from Felsenstein (2004). We report the results on 200 replicates.

Results and discussion

The level, with the cutoff set at -2, was very high, with all the levels reporting in between 37% and 52%. This is inappropriate for a statistical test. See Table 3.13.

The level, with the cutoff set at -10, was appropriate. All the of levels reported in betweeen 3% and 10%. While 10% is on the high side of what is ideal, it is still within the boundaries of acceptability. The level was also consistent across the sequence lengths and sequence divergence rates.

When the cutoff was -10, the power was very low, and it decreased as the recombination

rate increased. At its highest, it never reached 50%. See Table 3.14.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	ρ=0	46	37	40	48	46	47	52	49	51
-10	AIC	ρ=0	9	8	3	8	6	7	10	8	9

Table 3.13: Level of the test. Results for sequence lengths 500, 1000, and 2000; sequence divergence rates 1%, 5%, and 10%; recombination rates 0, 2, 4, and 8, and AIC. Continuous characters, untransformed, 20 taxa, 200 replicates.

Cutoff	Sequence Length		500			1000			2000	
	Sequence Divergence		1%	5%	10%	1%	5%	10%	1%	5%	10%
-2	AIC	ρ=2	66	69	71	71	70	78	74	77	73
		ρ=4	62	65	68	64	74	61	75	73	72
		ρ=8	51	57	50	52	51	52	60	47	59
-10	AIC	ρ=2	34	35	36	37	33	43	46	41	42
		ρ=4	27	28	31	28	41	30	37	43	42
		ρ=8	14	16	14	20	22	19	19	19	21

Table 3.14: Power of the test. Percentage of instances the AIC difference is greater than 10. Results for sequence lengths 500, 1000, and 2000; sequence divergence rates 1%, 5%, and 10%; recombination rates 0, 2, 4, and 8, and AIC. Continuous characters, untransformed, 20 taxa, 200 replicates.

3.3.5 Summary of findings about the recombination model and error structure

The results of the comparison with the PHI test suggested that some of our non-treelike alignments did not have a detectable recombination signal. Therefore, the power could be improved by considering only alignments with detectable signal. Therefore, one of the contributions to the ineffectiveness of this test is the type of data we used to test it. However, the α-level remained unacceptable despite conditioning on treelike alignments, so this is not the only contribution.

The results based on using the PHI test also suggested that the information within the site ordering was important. Our test was based on distances alone, and as such, could have been applied to any DNA data had it been successful. One of the reasons for its inefficacy is probably the fact that distances discard a great wealth of information.

One of the most surprising results from the initial simulations on the test was the decrease in power observed for the highest recombination rate. Our simulations on the very extreme model of recombination, the 'tree-plus-tree' method, confirmed that data sets with high recombination have low power. Therefore, another contribution to the inefficacy of this test is that tree based models might fit non-treelike distances better than network models when there is a lot of recombination.

The distances with noise added had a known noise or error structure. The noise was white noise with a specified standard deviation. The results showed consistency across sequence divergence rate and sequence length. The results based on the distances with noise added implied that our distances estimated from alignments did not have normally distributed noise. This means that our observed distances should not be assumed to be the true distances plus errors from a normal distribution. This is yet another contributor to the inefficacy of the test. This finding is at odds with Susko (2003).

The investigation which used continuous characters had very different behaviour from the original test. The level was lowest when the sequence divergence rate was 5% (rather than highest). As the sequence length increased, so did the level (and, unlike the original test, this behaviour did not depend on sequence divergence rate). The level was subject to much less variation ranging from 37% to 52% when the cutoff was -2 and from 3% to 10% when the cutoff was -10.

The continuous characters model showed more of the behaviours we expected. We expected there to be little variation in the level for a specific cutoff. The AIC values will reflect the underlying parameters (number of taxa, sequence divergence rate and sequence length) but the AIC differences should not. With continuous characters, the influence of sequence length was consistent over the range we investigated, and an increase in sequence length corresponded to a small increase in the level. This implied that the noise was increasing as sequence length increased, reducing the ability of our test to detect non-tree-likeness. This contrasted with our discrete sequences, as these had noise profiles that were not consistent. This further indicated that the noise in distance estimates from an alignment might be contributing to the ineffectiveness of this test.

If error structure was like white noise then the AIC approach might form a suitable approach to testing for tree-likeness. It seems that additional contributions to the error, especially from high recombination rates, influenced the performance of our test.

3.4 Investigating whether the power is higher for longer sequences and a higher sequence divergence rate

Our last set of investigations looked at long sequences and a higher sequence divergence rate to see if either of these increased the power.

An increase in the sequence length gave rise to a more acceptable level and more power if the sequence divergence rate was high. We wanted to investigate whether very long sequences had acceptable level. Some factors (such as the setting the cutoff such that the level is 5%) suggested that increasing the sequence divergence rate would lead to an acceptable α-level and more power, while other investigations (such as the declining power with an increase in sequence divergence rate as seen in the original experiment when the cutoff was -10) suggested that increasing the sequence divergence rate would lead to poorer performance. Therefore, we also ran the test on data sets with a sequence divergence rate of 20%.

We investigated the α-level and the power for sequences with a sequence divergence rate of 20% and with sequence lengths 500, 1000, and 2000, and recombination rates of zero, two, four, and eight.

We also investigated the α-level and the power for sequences with length 10,000 base pairs and with the sequence divergence rates 1%, 5%, and 10% to gauge the efficacy of the test with longer sequences.

We did not transform the distances and used only the AIC criterion. In both cases we report the results from 200 replications.

Sequence divergence rate: Results and Discussion

The level and power were always 0% when the sequence divergence rate was 20%. This was regardless of cutoff, sequence length, and recombination rate.

We also looked at the cutoff value to get the ideal level and the power that resulted from choosing this cutoff. The cutoff values were very small. This implied that the cutoff would not stablise with an increase in the sequence divergence rate. The power given

the ideal cutoff was good and ranged from 45% to 74%. It increased with an increase in recombination rate, which is pleasing. The longest sequences had the highest power.

Length	Cutoff value
500	-0.7106
1000	-0.7105
2000	-0.6397

Table 3.15: A table showing the cutoffs, which according to this set of simulations, give the correct level. Twenty taxa and sequence divergence of 20%.

Sequence Length	Sequence divergence	Recombination rate		
		2	4	8
500	20%	46	54	56
1000	20%	45	69	59
2000	20%	67	74	70

Table 3.16: A table showing the power if the level is set by ensuring the cutoff gives the correct level. All results are percentages based on 200 replications. Twenty taxa and sequence divergence of 20%.

Long sequences: Results and Discussion

For the lowest sequence divergence rate (1%) the level was very poor, being 65% at a cutoff of -2, and 41% at a cutoff of -10. This far exceeded the level seen when the sequence divergence rate was 5% or 10%. At a cutoff of -10 the level was 0% for sequence divergence rates 5% and 10%. This is acceptable. Thus, for the two higher sequence divergence rates, the level was too high with a -2 cutoff and too low with a -10 cutoff. This suggests that if sequence length was further increased, a cutoff of -2 would eventually be appropriate, but only for incredibly long sequences.

The power decreased as the recombination rate increased. This was concerning as we expected the power to increase. One potential reason for this is that longer sequences tend to have more site patterns, and thus some very unlikely patterns might have been present. The presence of these site patterns which did not support the tree would have created noise, and with long sequences this noise would have reduced the ability of the test to detect tree-likeness.

This simulation showed that the problems seen when the sequence divergence rate was low could not all be overcome by increasing the sequence length. However with longer sequences the sequence divergence rate of 5% had a more consistent performance, as the power was similar to that of the 10% sequence divergence rate. For the higher sequence divergence rates, we can see that eventually we would expect a cutoff of -2 to be appropriate.

Cutoff		Sequence Divergence		
		1%	5%	10 %
-2	$\rho = 0$	65	22	19
-10	$\rho = 0$	41	0	0

Table 3.17: Level of the test. Results for sequence length 10,000; sequence divergence rates 1%, 5%, and 10%; recombination rates (ρ) of 0, 2, 4, and 8. Untransformed, AIC, 20 taxa, 200 replications.

Cutoff		Sequence Divergence		
		1%	5%	10 %
-2	$\rho = 2$	87	76	80
	$\rho = 4$	81	76	76
	$\rho = 8$	64	60	63
-10	$\rho = 2$	67	54	52
	$\rho = 4$	55	52	50
	$\rho = 8$	30	23	29

Table 3.18: Power of the test. Results for sequence length 10,000; sequence divergence rates 1%, 5%, and 10%; recombination rates (ρ) of 2, 4, and 8. Untransformed, AIC, 20 taxa, 200 replications.

Summary of findings about longer sequence length and higher sequence divergence rate

Our experiment with long sequences suggested that very long sequences also have too much noise for the tree signal to be detected. As explained above, the longer sequences tended to have a great variety of site patterns and some very unlikely patterns might have been present. The presence of site patterns which did not support the tree would have created noise and reduced the ability of the test to detect tree-likeness. Increasing the sequence divergence rate did not give rise to a stable behaviour.

3.5 Discussion

A good statistical test has a low and stable α-level and high power. Here, we discuss what the investigations revealed about the level and power of the test we developed for investigating tree-likeness.

The test itself had very poor level. The level was lower for the sequence divergence rates 1% and 10%, and high when the sequence divergence rate was 5%. Increasing the sequence length caused the level to rise when the sequence divergence rate was 1% and to lower when the sequence divergence rate was 5% or 10%. The level was mostly over 20%, and ranged from 15% to 55% when the cutoff was -2, meaning that at least one in five trees reported in as networks.

The test had very low power. Ideally, the power should be high. Furthermore, we expected it to increase as the recombination rate increased; increase with sequence length; and increase with sequence divergence rate. We know that very high sequence divergence rates lead to sequences which are saturated (do not contain phylogenetic signal), and this would lead to a deterioration in the phylogenetic signal, but this is not the case with the sequence divergence rates we used.

Our detailed investigations show there was no single factor that could explain the poor performance of this test. It seems that the AIC calculation had a role. First, the sensitivity of the method used to estimate σ^2 suggests that accurately estimating σ^2 could improved the test's performance. This remains an open problem. Furthermore, noting that the error structure was not normal and that the AIC formula we used is based on normally distributed errors, we expect that the likelihood calculation also played a role. Currently, there is no better way to calculate the likelihood.

One consistent finding from our investigations is that our ideal of having a single cutoff was inappropriate. Many sets of simulations had ideal cutoffs (that is those which gave a α-level of 5%) that ranged greatly in magnitude. This was one of our test's downfalls.

While our model for recombination did not always have detectable recombination, it was fit for its purpose. It showed us interesting behaviour with alignments with high levels of recombination. These alignments were often reported as treelike by our test for tree-likeness. Our investigations have not fully resolved the issue of why this occured.

4

Partial LASSO

4.1 Background and motivation

The LASSO (Tibshirani, 1996) approach to regression became popular after the publication of Efron et al. (2004) and it is now considered standard practice to use the LASSO to get a solution to the regression equations. Throughout Chapters 2 and 3 we have used the LASSO to estimate the weights of the splits.

In this chapter, we describe a **partial** LASSO, which we developed to allow users to define a subset of variables which they believe should be in the model, and consequently, to use the model with these variables as a basis of comparison. The partial LASSO applies unconstrained optimisation to one set of variables, and the LASSO to the other subset of variables. In this manner, it is a hybrid between least squares and the LASSO.

In both the partial LASSO and full LASSO, the final model is the least squares solution on all of the variables. However, the intermediate models, between the least squares solution

on the unconstrained variables and the final model will be different from the set of models resulting from carrying out the LASSO on all of the variables.

Osborne et al. (2000a) investigated properties of the LASSO in an optimisation framework using the Karush-Kuhn-Tucker (KKT) conditions (Kuhn and Tucker, 1951). We follow this approach as we extend the LASSO to the partial LASSO. The KKT conditions are from the field of non-linear optimisation. They characterise the minimum of a function subject to inequality and equality constraints.

Our application is to use the partial LASSO to build the network from a star tree; that is, a tree with only trivial splits (splits which separate a single taxon from all of the other taxa). We believe the method has a much wider range of applications both within and outside of phylogenetics.

This chapter is laid out as follows. In the Section 4.2, we discuss the technical details of the partial LASSO. In Section 4.3, we discuss our application of the partial LASSO. We end the chapter with a discussion, in Section 4.4.

4.2 The partial LASSO

The positive LASSO algorithm solves the following problem for each choice of $\lambda \geq 0$:

> find $\boldsymbol{\beta}$ such that $||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||$ is minimised, such that $\sum_i \beta_i \leq \lambda$ and $\beta_i \geq 0$ for all i.

For the partial LASSO we define a set $\mathcal{L}$ of variables, which will be subject to the LASSO constraint. We then solve the following problem for each choice of $\lambda \geq 0$:

> find $\boldsymbol{\beta}$ such that $||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||$ is minimised, such that $\sum_{i \in \mathcal{L}} \beta_i \leq \lambda$ and $\beta_i \geq 0$ for all i.

When $\lambda = 0$ solving the partial LASSO is equivalent to solving the non-negative least squares problem for the variables not in $\mathcal{L}$, which can be done using a variety of methods (see Fletcher (2000)).

Suppose then that $\boldsymbol{\beta}$ solves the LASSO problem at $\lambda \geq 0$. Let

$$\mathcal{A}^{(\lambda)} = \{i \in \mathcal{L} : c_i = \kappa\},$$

$\mathcal{B}^{(\lambda)} = \{i \notin \mathcal{L} : c_i = 0\}$,

$\mathcal{C}^{(\lambda)} = \{i \in \mathcal{L} : c_i > \kappa\}$, and

$\mathcal{D}^{(\lambda)} = \{i \notin \mathcal{L} : c_i > 0\}$,

and, for each subset $\mathcal{V}$ of variables, let $\mathbf{X}_\mathcal{V}$ denote the matrix containing the columns of $\mathbf{X}$ relating to the variables $\mathcal{V}$. Likewise, for a vector $\mathbf{x}$ indexed by variables let $\mathbf{x}_\mathcal{V}$ denote $\mathbf{x}$ restricted to variables in $\mathcal{V}$.

Find $\mathbf{w}$ such that

$$\begin{pmatrix} \mathbf{X}_\mathcal{A}^T\mathbf{X}_\mathcal{A} & \mathbf{X}_\mathcal{A}^T\mathbf{X}_\mathcal{B} \\ \mathbf{X}_\mathcal{B}^T\mathbf{X}_\mathcal{A} & \mathbf{X}_\mathcal{B}^T\mathbf{X}_\mathcal{B} \end{pmatrix} \begin{pmatrix} \mathbf{w}_\mathcal{A} \\ \mathbf{w}_\mathcal{B} \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix}$$

with $\mathbf{w}_\mathcal{C} = 0$ and $\mathbf{w}_\mathcal{D} = 0$.

Define

$$\begin{aligned} \mathbf{c} &= \mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) \\ \mathbf{a} &= \mathbf{X}^T\mathbf{X}\mathbf{w} \\ \kappa &= \min_{i\in\mathcal{L}} c_i \\ \gamma_{\mathcal{AB}} &= \min\left\{\frac{-\beta_i}{w_i} : w_i < 0, i \in \mathcal{A}\cup\mathcal{B}\right\} \\ \gamma_\mathcal{C} &= \min\left\{\frac{c_i - \kappa}{1 - a_i} : 1_i - a_i > 0, i \in \mathcal{C}\right\} \\ \gamma_\mathcal{D} &= \min\left\{\frac{c_i}{a_i} : a_i > 0, i \in \mathcal{D}\right\} \quad \text{and} \\ \hat{\gamma} &= \min\{\gamma_{\mathcal{AB}}, \gamma_\mathcal{C}, \gamma_\mathcal{D}\}. \end{aligned}$$

We prove below that for all γ such that $0 \le \gamma \le \hat{\gamma}$, $\boldsymbol{\beta} + \gamma\mathbf{w}$ is optimal solution of the partial LASSO problem with LASSO constraint $\lambda + \gamma\sum_{i\in\mathcal{L}} w_i$.

The algorithm advances to $\lambda + \hat{\gamma}\mathbf{1}^T\mathbf{w}$, updates $\boldsymbol{\beta} \to \boldsymbol{\beta} + \hat{\gamma}\mathbf{w}$ and records this value, before continuing to the next iteration.

Theorem 1 *For all γ such that $0 \le \gamma \le \hat{\gamma}, \boldsymbol{\beta}^{(\lambda)} = \boldsymbol{\beta} + \gamma\mathbf{w}$ minimises $||\mathbf{X}\boldsymbol{\beta}^\gamma - \mathbf{y}||$ such that $\sum_{i\in\mathcal{L}} \beta_i^\gamma = \lambda + \sum_{i\in\mathcal{L}} \gamma w_i$.*

Proof

For each λ, the partial LASSO problem is a convex quadratic programming problem, so $\boldsymbol{\beta}$ is the optimum if and if only there are multipliers η, κ that satisfy the KKT conditions:

$$\begin{aligned} c_i - \eta_i - \kappa &= 0 \qquad \forall\ i \in \mathcal{L} \qquad (4.1)\\ c_i - \kappa &= 0 \qquad \forall\ i \notin \mathcal{L} \qquad (4.2)\\ \sum_{i\in\mathcal{L}} \beta_i &= \lambda \\ \beta_i &\geq 0 \\ \eta_i &\geq 0 \\ \beta_i \eta_i &= 0 \qquad \forall i. \end{aligned}$$

Recall $\mathbf{c} = \mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$, so that first condition (that given in Equations (4.4) and (4.5)) is the gradient of the Lagragian (Kuhn and Tucker, 1951; Fletcher, 2000).

Suppose that $0 \geq \gamma \geq \hat{\gamma}$. Define

$$\begin{aligned} \kappa^{(\gamma)} &= \kappa + \gamma \\ \eta_{\mathcal{A}}^{(\gamma)} &= 0 \\ \eta_{\mathcal{B}}^{(\gamma)} &= 0 \\ \eta_{\mathcal{C}}^{(\gamma)} &= c_i - \kappa \\ \eta_{\mathcal{D}}^{(\gamma)} &= c_i - \gamma \end{aligned}$$

We want to show that

$$\boldsymbol{\beta}^{\gamma} = \boldsymbol{\beta} + \gamma\mathbf{w} \qquad (4.3)$$

satisfies the KKT conditions.

$$\begin{aligned}
\mathbf{c}_i^\gamma - \eta_i^\gamma - \kappa^\gamma &= 0 \qquad \forall\ i \in \mathcal{L} && (4.4)\\
\mathbf{c}_i^\gamma - \kappa^\gamma &= 0 \qquad \forall\ i \notin \mathcal{L} && (4.5)\\
\sum_{i\in\mathcal{L}} \beta_i^\gamma &= \lambda^\gamma && (4.6)\\
\beta_i^\gamma &\geq 0 && (4.7)\\
\eta_i^\gamma &\geq 0 && (4.8)\\
\beta_i^\gamma \eta_i^\gamma &= 0 \qquad \forall i. && (4.9)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{c}^\gamma &= \mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}^\gamma - \mathbf{y})\\
\lambda^\gamma &= \lambda + \hat{\gamma}\sum_{i\in\mathcal{L}} w_i.
\end{aligned}$$

Note (4.6) is satisfied trivially, (4.7) is satisfied by the constraint that $\gamma \leq \gamma_{\mathcal{AB}}$, (4.8) is satisfied by constraint that $\gamma \leq \min\{\gamma_{\mathcal{C}}, \gamma_{\mathcal{D}}\}$. Condition (4.9) is satisfied as for sets $\mathcal{A}$ and $\mathcal{B}$ as we have $\eta_i = 0$ and for sets $\mathcal{C}$ and $\mathcal{D}$ as we have $\beta_i = 0$.

To show (4.4) and (4.5), we consider four cases for the index i.

For set $\{i \in \mathcal{A}\}$ from Equation (4.4) we have

$$\begin{aligned}
[\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}^\gamma - \mathbf{y})]_i - \hat{\kappa}^\gamma &= \{\mathbf{X}^T[\mathbf{X}(\boldsymbol{\beta} + \gamma\mathbf{w}) - \mathbf{y}]\}_i - \kappa - \gamma a_i\\
&= [\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})]_i + \gamma[\mathbf{X}^T\mathbf{X}\mathbf{w}]_i - \kappa - \gamma a_i\\
&= c_i + \gamma a_i - \kappa - \gamma a_i\\
&= c_i - \kappa\\
&= 0.
\end{aligned}$$

For set $\{i \in \mathcal{B}\}$ from Equation (4.5) we have

$$\begin{aligned}
[\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}^\gamma - \mathbf{y})]_i &= [\mathbf{X}^T(\mathbf{X}(\boldsymbol{\beta} + \gamma\mathbf{w}) - \mathbf{y})]_i \\
&= [\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}-)\mathbf{y}]_i + \gamma[\mathbf{X}^T\mathbf{X}\mathbf{w}]_i \\
&= [\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})]_i \\
&= c_i \\
&= 0.
\end{aligned}$$

For set $\{i \in \mathcal{C}\}$ from Equation (4.4) we have

$$\begin{aligned}
[\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}^\gamma - \mathbf{y})]_i - \eta_{\mathcal{C}}^\gamma - \hat{\kappa} &= [\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) + \gamma\mathbf{X}^T\mathbf{X}\mathbf{w}]_i - (c_i - \kappa) - (\kappa + \gamma a_i) \\
&= c_i + \gamma a_i - (c_i - \kappa) - (\kappa + \gamma a_i) \\
&= 0.
\end{aligned}$$

For set $\{i \in \mathcal{D}\}$ from Equation (4.5) we have

$$\begin{aligned}
[\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}^\gamma - \mathbf{y})]_i - \eta_{\mathcal{D}}^\gamma &= \{\mathbf{X}^T[\mathbf{X}(\boldsymbol{\beta} + \gamma\mathbf{w}) - \mathbf{y}]\}_i - (c_i + \gamma) \\
&= c_i + \gamma[\mathbf{X}^T\mathbf{X}\mathbf{w}]_i - c_i - \gamma \\
&= 0,
\end{aligned}$$

as required □

The algorithm for solving a set of equations using the partial LASSO is similar to the partial LASSO algorithm.

1. Let $\boldsymbol{\beta}$ be the minimum of $||\mathbf{X}\boldsymbol{\beta} - \mathbf{y}||$ such that $\boldsymbol{\beta} \geq 0$, $\beta_i = 0, \ \forall i \in \mathcal{L}$.
2. Calculate $\mathbf{c}$ as $\mathbf{c} = \mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$.
3. Set $\kappa = \min_{i \in \mathcal{L}} c_i$
4. Define the four sets

   $$\mathcal{A} = \{i \in \mathcal{L} : c_i = \kappa\}$$

$$\mathcal{B} = \{i \notin \mathcal{L} : c_i = 0\}$$

$$\mathcal{C} = \{i \in \mathcal{L} : c_i > \kappa\}$$

$$\mathcal{D} = \{i \notin \mathcal{L} : c_i > 0\}$$

5. Let $\tilde{\mathbf{w}}$ solve $(\mathbf{X}_{\mathcal{AB}}^T \mathbf{X}_{\mathcal{AB}})\tilde{\mathbf{w}} = \begin{pmatrix} \mathbf{1}_{\mathcal{A}} \\ 0_{\mathcal{B}} \end{pmatrix}$.
6. Extend $\tilde{\mathbf{w}}$ to $\mathbf{w}$ by $\mathbf{w}_{\mathcal{AB}} = \tilde{\mathbf{w}}_{\mathcal{AB}}, \mathbf{w}_{\mathcal{CD}} = 0$.
7. Calculate $\mathbf{a}$ as $\mathbf{a} = \mathbf{X}^T\mathbf{X}\mathbf{w}$.
8. Calculate γ on each set.

$$\gamma_{\mathcal{AB}} = \min\{-\beta_i/w_i : w_i < 0, i \in \mathcal{A} \cup \mathcal{B}\}$$

$$\gamma_{\mathcal{C}} = \min\{\tfrac{c_i - \kappa}{1 - a_i} : 1 - a_i > 0, i \in \mathcal{C}\}$$

$$\gamma_{\mathcal{D}} = \min\{\tfrac{c_i}{a_i} : a_i > 0, i \in \mathcal{D}\}.$$

9. Let $\gamma = \min\{\gamma_{\mathcal{AB}}, \gamma_{\mathcal{C}}, \gamma_{\mathcal{D}}\}$.
10. Update $\boldsymbol{\beta}, \kappa$ and c using

$$\boldsymbol{\beta}^{(\lambda)} \leftarrow \boldsymbol{\beta} + \gamma\mathbf{w}$$

$$\kappa^{(\lambda)} \leftarrow \kappa + \gamma$$

$$c^{(\lambda)} \leftarrow c + \gamma a.$$

11. If $\kappa = 0$ then stop otherwise go to step 4.

Once $\kappa = 0$ the NNLS solution has been obtained.

The algorithm will have a finite number of steps. As λ increases the sum of the residuals must decrease and at some particular values of λ variables are added or removed. This guarantees $\sum_{i \in \mathcal{L}}$ ¿0 and that it does not get smaller as $\boldsymbol{\beta}$ increases.

In our experience to date the algorithm always converges to the NNLS solution.

4.3 Application

4.3.1 Partial LASSO with NNLS hybrid applied to neighbor-net

In this section, we apply the partial LASSO to neighbor-net in the same manner in which we applied the LASSO in Chapter 2.

We used the same two measures to assess the performance of the partial LASSO. The measure of the performance of the fits is

$$\text{difference in fit} = \sum_{ij} \frac{d_{ij} - \hat{d}_{ij}}{d_{ij}} \tag{4.10}$$

where d is the pairwise distances and $\hat{d}$ is the modelled pairwise distances. Ideally, this difference should be very small.

The measure we used to study the reduction in the number of splits is a comparison of the number of splits chosen under the partial LASSO framework with the number of splits chosen by neighbor-net as estimated by non-negative least squares.

We once again used an NNLS-hybrid; that is, we used the partial LASSO algorithm to choose the splits, but the $\boldsymbol{\beta}$ coefficients came from fitting an NNLS model. In the next section we do not use the NNLS hybrid.

The simulation study follows that of Wiuf et al. (2001). Alignments were generated according to the coalescent model of Hudson (1983). In all cases, the number of taxa used was 20. The sequence lengths used were 500, 1000, and 2000 base pairs. The recombination parameter values used were zero, two, four, and eight. Higher recombination parameters should give rise to less tree-like data. The expected sequence divergence rates were one, five, and ten percent site differences. All sets of simulations have 200 replications.

The fit of the distances was very close regardless of sequence length, sequence divergence rate, and recombination rate. The fits were, on average, closer than those of the LASSO. With the LASSO, the fits were not close for the higher recombination rates; but with

the partial LASSO, we see that the fits were close regardless of the recombination rate. The bias is different, with distances more likely to be over-fitted with the partial LASSO, while the distances were more likely to be under-fitted with the LASSO.

Often the partial LASSO network is the original neighbor-net network. The proportion of instances of this decreased as the sequence divergence rate increased; remained unchanged as the sequence length increased and increased as the recombination rate increased. This behaviour contrasts with the LASSO, where the proportion increased as the sequence divergence rate increased, and decreased as the recombination rate increased. While the LASSO trimmed splits from the alignments with high recombination rates, the partial LASSO does not seem to be able to do this.

See Appendices H and I for plots of the fits of the distances and number of splits compared with the original neighbor-net network.

4.3.2 Partial LASSO without the NNLS hybrid applied to neighbor-net

We carried out the same experiment as above, but did not use the NNLS hybrid. In this experiment, we used only one σ^2 estimator, $\hat{\sigma}^2_N$ based on the final partial LASSO model. We used the same two measures of performance: fit of distances, and a comparison of the number of splits.

The general observation is that the fit of the distances was very close regardless of recombination rate, sequence divergence rate, and sequence length.

The number of splits chosen relative to the original neighbor-net network increased as the recombination rate increased, and remained unchanged as the sequence length and sequence divergence rate increased. Compared with the partial LASSO-NNLS hybrid, there are, on average, fewer splits in the partial LASSO with the $\hat{\sigma}^2_N$ estimator.

See Appendix J.

4.4 Discussion

The partial LASSO extends the LASSO approach. The partial LASSO allows users to define a set of variables to be the initial model; the LASSO regression approach is then applied to a further set of regressors.

The theorem shows us exactly how to move from the initial model to the LASSO solutions by varying λ. We also gave the algorithm we used in our implementation of the partial LASSO. This gives the exact vector to travel along at each step, and how to calculate how far to go in that direction.

The partial LASSO has potential to be widely applicable, as any time there is a prior belief that a certain variable or group of variables should be in the model, our partial LASSO could allow users to put these variables in the model first.

Our application was neighbor-net. We noticed, while working with the LASSO approach to neighbor-net, that trivial splits were often added to the model in the later stages. Therefore, for the partial LASSO we let the trivial splits be our user-defined group of variables so that they would be in the model first.

We ran two sets of simulation studies: one in which we used a partial LASSO-NNLS hybrid, and one in which we used only the partial LASSO. With the NNLS hybrid, we found that in the majority of cases the partial LASSO network was the same as the neighbor-net network. Without the NNLS hybrid the estimator of $\hat{\sigma}^2_N$ was smaller than either of $\hat{\sigma}^2_T$ or $\hat{\sigma}^2_N$ estimated under the partial LASSO-NNLS hybrid. As a result, the models contain fewer splits.

Therefore, using the partial LASSO approach with σ estimated using $\hat{\sigma}^2_N$ (a smaller estimate of σ^2 than one under the NNLS hybrid) allows us to place the trivial splits in the model first, and results in networks that, on average, have fewer splits than the original neighbor-net network. This further shows us how estimating σ^2 strongly influences the results.

5

Visualising heterogeneity in a set of trees

In this chapter we describe a method for visualising heterogeneity in sets of trees.

As we discussed in the Introduction, networks can visualise conflict in trees and they can represent multiple trees. In this chapter, we focus on networks which visualise the key features of a set of trees. Our application was to use a splits-based approach and the LASSO (Tibshirani, 1996) to visualise the set of trees from an MCMC run.

Consensus tree methods take a set of trees and represent them as a single tree. Majority-rule consensus trees (Margush and McMorris, 1981) contain subtrees which appear in at least 50% of the input set of trees. Strict consensus trees (Rohlf, 1982) contain subtrees which appear in all of the input trees. Therefore, when there is conflict, the tree becomes unresolved; that is, several taxa meet at an ancestral node, rather than just two taxa as in resolved trees.

Consensus networks (Holland and Moulton, 2003) are a splits based extension of consensus trees. These networks display all the splits that are in at least the specified proportion of the input trees. These networks may be multidimensional, and Holland and Moulton (2003) introduced a greedy algorithm which chooses a splits system to visualise the set of splits.

We aimed to provide a network-style representation of the important features of the set of input trees. By expressing the statistically significant information in set of input trees as a network, we were able to see which parts of the tree were fully resolved and which parts had uncertainty. This generalises Holland and Moulton (2003) by developing a framework for systematically choosing an appropriate set of splits to represent the information in the input set of trees.

We applied our method to the set of trees which arise from Bayesian analysis with Markov Chain Monte Carlo (MCMC) sampling. Bayesian analysis has had a great impact on estimating phylogenetic trees (Huelsenbeck and Ronquist, 2001; Drummond and Rambaut, 2007). An MCMC run produces an ordered set of trees, and one of the challenges of Bayesian phylogenetics is how to interpret and summarise this output. Typically, users report the most frequently-observed tree topology, but this discards a great wealth of information. Here, we investigated using a statistically-informed technique for representing these large sets of trees.

5.1 Method

As in Chapters 2, 3, and 4, we used regression as the statistical framework for the development of these consensus networks. The setup was as follows.

Each tree was represented as a vector indexed by splits. Each input tree was converted into a set of splits with associated branch lengths or split weights. Let m be the total number of splits in the set of input trees and let N be the number of trees. Tree i was encoded as a vector of length m with branch lengths as entries, the vector is denoted $\mathbf{y}^{(i)}$. If a split did not appear in the tree, the entry was zero.

We assumed that each tree vector $\mathbf{y}^{(i)}$ had an approximately normal distribution with unknown mean μ and covariance Σ. We used networks to represent the mean vector

μ, which is estimated using $\overline{y}$ (where $\overline{y}$ is the observed mean of all the $\mathbf{y}^{(i)}$). We used the LASSO framework to make sure only well-supported splits were displayed. This was equivalent to minimizing the least squares residuals

$$\sum_{i=1}^{N}(\mathbf{y}^{(i)} - \boldsymbol{\beta})^T\Sigma^{-1}(\mathbf{y}^{(i)} - \boldsymbol{\beta}) \tag{5.1}$$

in a regression analysis with untransformed design matrix equal to the identity. Σ was the covariance matrix, and $\boldsymbol{\beta}$ was the vector of regression coefficients or split weights. The elements of the predictive vector were not independent of each other; however, we estimated the covariance matrix from the input trees. We expected the sample covariance matrix to be positive definite, and that it would accurately reflect the correlation structure. However, it was possible that the matrix may not be well-conditioned. We stabilised the covariance matrix by adding the mean of the variances to the diagonal and scaling, as recommended by Schäfer and Strimmer (2005).

Regression was carried out on the transformed means and splits. As in Chapter 2, the positively-constrained LASSO algorithm was used, providing a suite of models.

Let $\mathbf{R}$ be upper triangular factor in the Cholesky decomposition of the inverse of Σ. The LASSO solution in this scenario for a given λ is the value $\boldsymbol{\beta}$ minimising

$$S(\boldsymbol{\beta}) = ||\mathbf{R}^T\mathbf{y} - \mathbf{R}^T\mathbf{I}\hat{\boldsymbol{\beta}}|| \tag{5.2}$$

subject to the LASSO constraints

$$\beta_i \geq 0, \tag{5.3}$$

and

$$\sum_{j=1}^{m}\beta_j < \lambda, \tag{5.4}$$

for all j.

We used the positive LASSO algorithm of Efron et al. (2004) to get sets of candidate splits and we re-estimated the split weights using non-negative least squares. The final model was chosen by the AIC criterion.

We applied this method to two data sets.

5.2 Case studies

5.2.1 Case study one: Tiger moths

Ratcliffe and Nydam (2008) published a study on tiger moths. They sequenced one mitochondrial gene (*cytochrome oxidase I*, COI) and two nuclear genes (*elongation factor 1a*, EF1a, and *wingless*) from 26 closely-related moths and the out-group species *Lymantria Dispar*. The study used the 50% majority consensus phylogram to define phylogenetic clusters. These clusters and several variables (the types of clicks the moths are capable of making, their colouring, and the extent to which the species is nocturnal) were used to perform a comparative analysis to investigate the evolution of signaling.

Following Ratcliffe and Nydam (2008) we ran a Bayesian analysis using MrBayes 3.1.2 (Huelsenbeck and Ronquist, 2001). This involved using a GTR + I + G model to COI and a SYM + I + G model to EF1a and wingless. We ran the analysis for 10 million generations, sampling every 1,000 generations. We carried out two runs with a 20% burn in. We applied our consensus network method to the resulting trees and produced the network given in Figure 5.1. The AIC and BIC criteria chose the same model.

The 50% majority consensus phylogram produced in Ratcliffe and Nydam (2008) had low support for the branches separating the four clades seen at the bottom of Figure 5.1. This uncertainty was reflected in the box-like structures seen in this area. This implies a lack of resolution in the tree at this position, and possibly a rapid expansion (that is, where an ancestral species has quickly diverged into several species). The consensus network of Holland and Moulton (2003) had a great deal of reticulation at this location representing the uncertainty (Figure 5.2).

A Lento plot shows the support and conflict for a split. The conflict is "the sum of all other splits that contradict the partitioning of taxa in the first split" (Lento et al., 1995). The Lento plot (Figure 5.3) shows that the set of splits strongly overlaps, and that the conflict is greater in the consensus network of Holland and Moulton (2003), as we can see from the much larger reticulations.

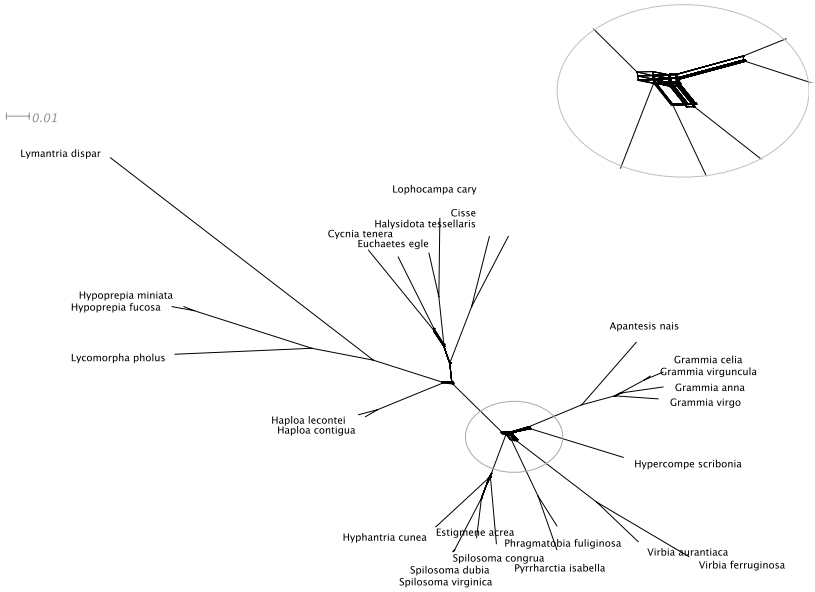

Figure 5.1: LASSO consensus network for Tiger moths based on the data of Ratcliffe and Nydam (2008) using our consensus network method.

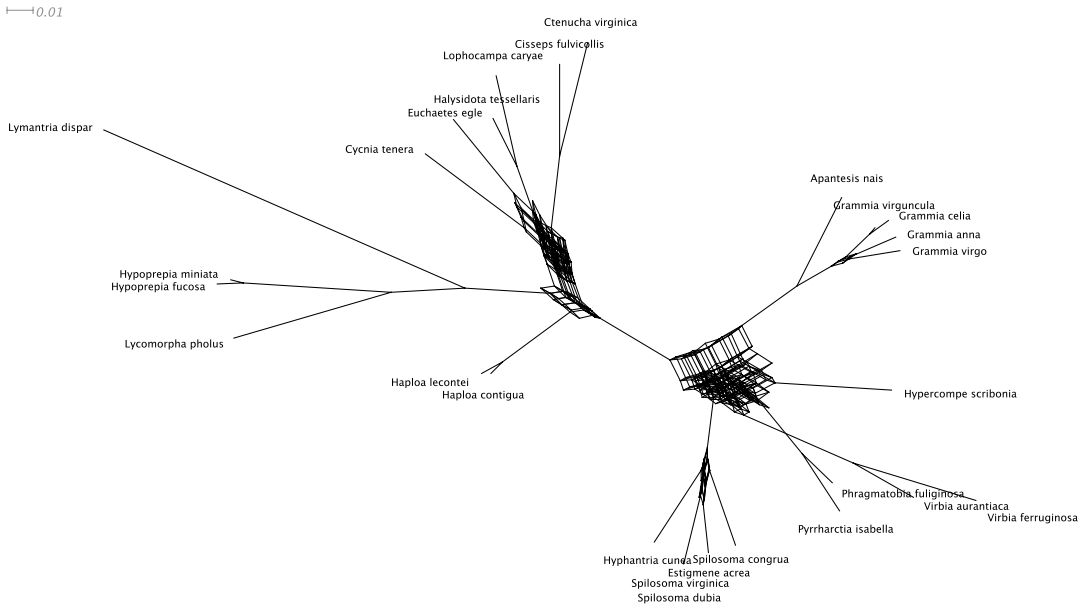

Figure 5.2: Consensus network for Tiger moths based on the data of Ratcliffe and Nydam (2008) using the consensus network method of Holland and Moulton (2003), mean edge weights, threshold of zero.

Figure 5.3: Lento plot of the two consensus networks showing the split weights and the conflict against these split weights.

5.2.2 Case study two: Human Mitochondria

The second data set has 53 taxa (Ingman et al., 2000). The data set contained full mitochondrial sequences, as opposed to just the control region. Ingman et al. (2000) found support for the "out of Africa" hypothesis; that is, the hypothesis that humans originated in Africa. Vigilant et al. (1991) carried out the first DNA based study into the "out of Africa" hypothesis, estimating the age of the most recent common ancestor to be about 170,000 years ago, plus or minus 50,000 years. Their method was based on the assumption that the data is tree-like; later analysis using network methods showed that the network was cluttered and non-treelike, mostly likely a result of fast-evolving sites cluttering the tree-like signal (Bryant and Moulton, 2004).

With 53 taxa there are considerably more potential splits than in the previous example. The total number of splits observed in the set of MCMC trees was 560. If we allowed the consensus network to be applied to all of these splits, then the AIC criteria chose 324 splits while the BIC criteria chose 312 splits. Splitstree is not capable of drawing the corresponding network.

The summary information in Figure 5.4 shows that beyond iteration 100 very little improvement was made to the model. A further inspection of the size of the coefficients confirmed this. Therefore, we choose a model with 120 splits, which is displayed in Figure 5.5.

This result suggested that AIC and BIC criteria are not optimal for model selection in this context. We need a criterion with a higher penalty for including additional splits.

If we subset the taxa, we see that for the 20 randomly-selected taxa, the consensus network is quite treelike (see Figure 5.6). The few areas of non-treelikeness are so small that they are barely distinguishable.

5.3 Discussion

This style of consensus method has potential to provide a quick visual summary of the MCMC results. With the Tiger moths example, we saw that areas with low posterior support, as seen in the majority consensus phylogram showed up as small reticulations in

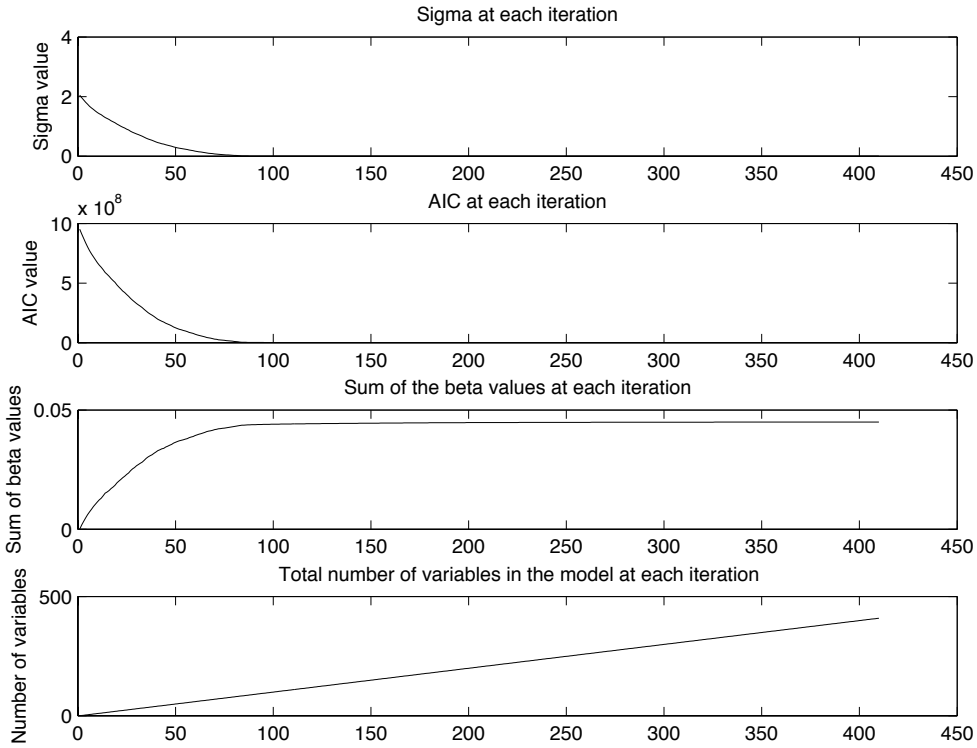

Figure 5.4: Summary information for the Consensus network for Human mitochondria based on the data of Ingman et al. (2000).

the network.

There is scope for a further investigation into alternative model selection tools. One popular model selection tool is cross-validation. A potential implementation would involve taking a sample of the sites and re-running the entire procedure; this is computationally not feasible at this stage.

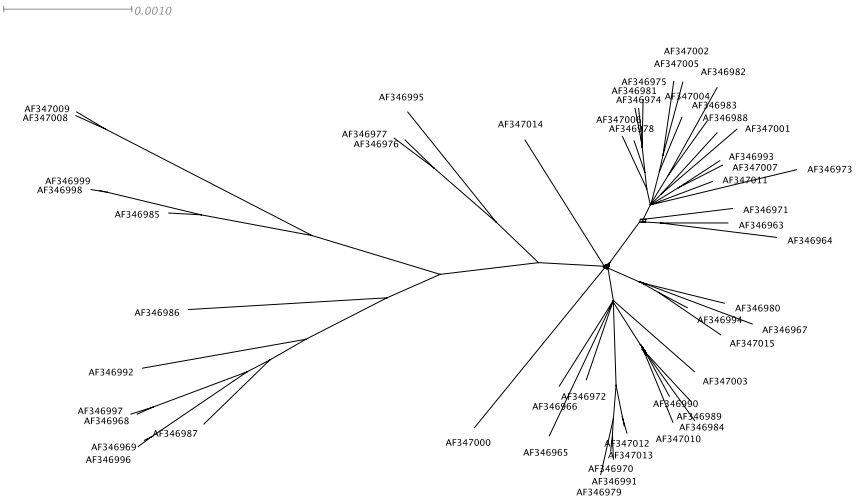

Figure 5.5: Consensus network for Human mitochondria based on the data of Ingman et al. (2000). Network is based on the first 120 splits chosen by the LASSO algorithm.

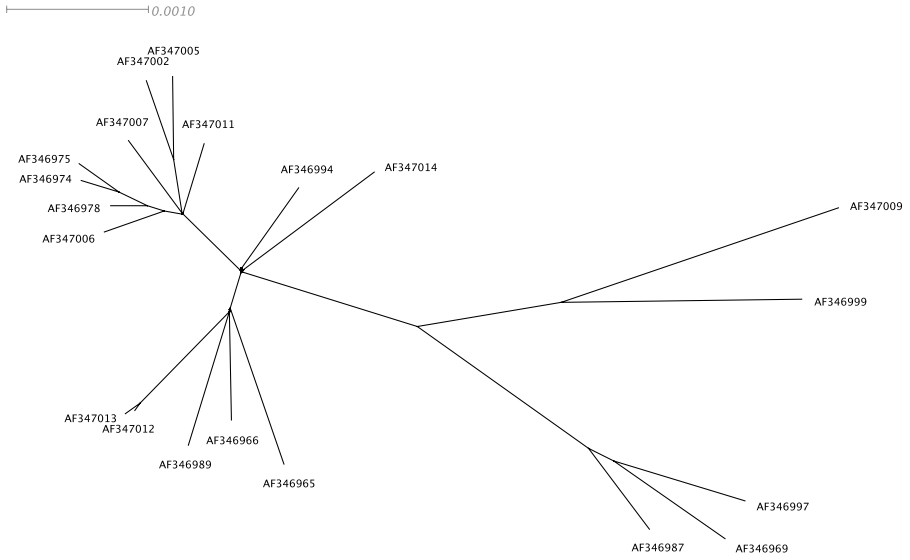

Figure 5.6: Consensus network for Human mitochondria based on the data of Ingman et al. (2000) for a randomly selected subset of 20 taxa from the 53 of Ingman et al. (2000).

6

Confidence sets on trees

6.1 Background and motivation

In this chapter we further investigate incongruence by developing a confidence set method for a set of genes.

With the rise of modern sequencing technology, it is increasingly common for phylogenetic data sets to be comprised of multiple genes from a single sample. One of the first approaches for phylogenetic analysis of this type of data was to concatenate the genes and form a phylogenetic tree from the concatenated alignment; see Baldauf (1999) and many others. However, there is a growing body of knowledge about how genes evolved under different pressures within a genome, and an understanding that different genes may have different evolutionary histories. Therefore, it is relevant to ask: which genes evolved along the same topology? Only genes arising from a single topology should be concatenated for a single phylogenetic analysis.

Several authors have proposed tests to examine whether several genes have evolved on a single topology. Farris et al. (1995) developed the incongruence length difference method, which compared the length of the parsimony-based tree on the combined data with the combined length of all the trees for each gene. Bapteste et al. (2008) developed a method called progressive reconstruction analysis. This is a split-based approach which applies an algorithmic procedure and results in a heat map from which similar genes can be observed. Leigh et al. (2008) developed CONCATERPILLAR. The method uses hierarchical clustering and likelihood ratio tests to identify congruent genes. The method combines pairs of genes and compares the likelihood of the combined data with each of the underlying genes. The pair with the smallest likelihood ratio is then treated as a single gene and compared with the other genes (or gene groupings). A non-parametric bootstrap is used to determine significance levels and the procedure ends when the smallest likelihood ratio test exceeds the α threshold. Additionally, Goldman et al. (2000) reviewed the inappropriate, but popular, use of the Kishino-Hasegawa test (Kishino and Hasegawa, 1989) (which was originally designed as a method to compute confidence intervals on posterior probabilities resulting from a Bayesian analysis) as a test for creating a confidence interval of topologies.

Here, we developed a method based on combining the p-values associated with single genes to obtain a confidence set of topologies. An empty confidence set corresponds to a rejection of the null hypothesis; that is we reject the hypothesis that the genes have evolved on the same topology.

6.1.1 Hypothesis testing for phylogenies

A hypothesis, is a statement about a population parameter, tested using a test statistic. A hypothesis test specifies the range of values under which the null hypothesis is not rejected, and the range of values under which the alternative hypothesis is accepted. Some hypothesis tests generate a p-value (that is the probability that a result at least as extreme as that in the data was generated under the null hypothesis). The p-value is the strength of evidence against the null hypothesis. In classical statistics, the p-value is compared with the α threshold and when the p-value is less than α the null hypothesis is rejected. As α gets smaller the test becomes more conservative in favour of the null hypothesis. Despite philosophical objections to the hypothesis testing approach we use it

for pragmatic reasons.

Let T_i be the true but unknown topology for gene i. Let T be a potential or candidate topology. To test the null hypothesis that a gene evolved on a particular candidate topology, we test

$$H_{(i)}^{T} : T_i = T. \tag{6.1}$$

Constructing the set of candidate trees

For very small taxon sets it is possible to use all possible topologies as the set of candidate trees, but in general the set of all possible topologies is too large. Therefore, we need a reliable method of generating a set of candidate trees. This set of trees should contain the most likely true topology from each gene, and contain all the trees which might remain in the confidence set. The set should be generous in size to increase the chances of the true tree being in the candidate set.

The procedure we developed for creating a set of candidate trees was not overly computationally intensive. We first used RAxML (Stamatakis, 2006) to obtain a maximum likelihood tree for each gene. RAxML is efficient at estimating the likelihoods; these trees are included in the set of candidate tress. The maximum likelihood trees were used as input for the Most-Parsimonious Reconstruction (MPR) supertree method (Bremer, 1990; Ragan, 1992). Using PAUP* (Swofford, 2000), we added to the candidate set all other trees with maximum parsimony scores that were equal to or less than those already in the candidate set. If the candidate set was too small, then we also added the trees which were up to $k + 1$ steps away from the maximum parsimony supertree where k is the maximum of the number of steps away a tree currently in the candidate set is from the supertree. This was repeated until the candidate set was at least as big as the desired size; usually, 100 trees. In future we need to examine the construction of this candidate to ensure that

To summarise, the procedure we used to construct the set of candidate tree is:

1. Estimate the maximum likelihood tree for each gene using RAxML.
2. Create a binary character matrix from gene tree bifurcations.
3. Use the character matrix to find the maximum parsimony supertree using PAUP*.

4. Use PAUP* to find all the trees within one step, two steps, etc until the set of trees contains all of the maximum likelihood tree and is sufficiently large.

Computing p-values for each tree, T

There are several methods that compute p-values for a single tree given one gene, including the Kishino-Hasegama (KH) test (Kishino and Hasegawa, 1989), the Shimodaira-Hasegama (SH) test (Shimodaira and Hasegawa, 1999), the Approximately Unbiased (AU) test (Shimodaira, 2002), and minBP of Susko (2006).

Below, we show how we used the single distribution nonparametric bootstrap (SDNB) method of Shi et al. (2005) to get a p-value for each topology and each gene. The main advantages of this method are that the coverage is small and it is computationally efficient.

The test statistics for the SDNB method are log likelihood differences; that is, $\delta_1 = l_{ML} - l_1, \ldots, \delta_m = l_{ML} - l_m$ where ML is the maximum likelihood tree on the alignment and m is the number of topologies to test. Instead of generating a bootstrap replicate to construct a distribution for δ_i of each topology, the same bootstrap replicate was used to test all m hypotheses. The procedure was:

1. Generate nonparametric bootstrap replicate data sets by sampling the sites of the alignment.

2. Estimate the maximum likelihood trees for the original alignment and each of the bootstrap replicates. We used RELL bootstrapping (Kishino et al., 1990) rather than re-estimating the parameters of the maximum likelihood tree for each replicate.

3. Approximate the distribution of δ_i by differencing the log likelihood scores of the maximum likelihood tree from the original alignment and the maximum likelihood tree of the replicate.

4. Compare the sample δ_i values to the distribution. The p-value is the proportion of bootstrap replicates with δ values higher than the sample δ_i.

The use of a single set of bootstrap replicates ensures that this procedure is computationally efficient.

6.2 Constructing confidence sets on multiple genes

For several genes, we have the null hypothesis

$$H_{eq} : T_1 = T_2 = \ldots = T_k, \tag{6.2}$$

that is all the genes evolved on the same topology.

For a specific candidate topology, we have

$$H_{eq}^{T} : T_1 = T_2 = \ldots = T_k = T \tag{6.3}$$

so that

$$H_{eq}^{T} = \cap_{i=1}^{k} H_{(i)}^{T}. \tag{6.4}$$

We obtained a confidence set of topologies by including all the topologies that do not reject the hypotheses in Equation (6.4).

We used composite p-values to test the hypotheses in Equation (6.4). Composite p-values combine p-values, and here we combined p-values from each gene to get a p-value for each candidate topology across all genes. There is a body of literature on how to find a composite p-value from a range of p-values; see Loughin (2004) and the references therein for overviews of the field.

The method we used to calculate the composite p-values for each topology was the Normal method or Stouffer's method. Stouffer's method was published in Stouffer et al. (1949) as an obscure footnote; see Whitlock (2005) for other applications of the method within evolutionary biology. Stouffer's method falls into the category of quantile combination approaches (Loughin, 2004). An advantage of this method is that the p-values have equal emphasis or weighting, and as such, a single gene will not determine the acceptance or rejection of the null hypothesis. Stouffer's method assumes that the p-values come from a one-sided hypothesis test and that they are uniformly distributed on [0,1] under the null hypothesis.

Stouffer's method has four steps, namely:

1. Convert each p-value to a Z-score (that is, $\Phi^{-1}(p\text{-value}) \sim N(0,1)$ under the null

hypothesis).

2. Sum the Z-scores.
3. Divide the sum by the square root of the number of p-values combined (this standardises the sum) giving a composite Z-score.
4. Convert the composite Z-score to a p-value.

The formula for the composite Z-score is

$$Z_T(X_1, \ldots, X_k) = \frac{\sum_{i=1}^{k} Z_T(X_i)}{\sqrt{k}}, \tag{6.5}$$

where X_i is the data from each of the units being combined.

We combined the p-values from each gene and each topology to get a p-value for each topology.

So, combining our knowledge from studies on p-values of single genes and the Z-score method, we had the following procedure:

1. Construct a set of candidate trees.
2. For each candidate tree T, and each gene X_i, compute the p-values $p_T(X_i)$.
3. For each candidate tree T, and each gene X_i, compute $Z_T(X_i) = \Phi^{-1}(p_T(X_i))$.
4. For each candidate tree T, compute $p_T(X_1, \ldots, X_k) = \Phi\left(\frac{\sum_{i=1}^{k} Z_T(X_i)}{\sqrt{k}}\right)$.
5. Form the confidence set containing all trees T for which $p_T(X_1, \ldots, X_k) \geq \alpha$. If there are no such trees, reject the null hypothesis H_{eq}, that all genes evolved on the same tree.

6.3 Simulation study

We used simulations to investigate the efficacy of the composite Z scores method. We were interested in the level (type I error) of the test; that is, the performance of the method when there was a single topology. We were also interested in the attributes of the confidence set when the data were generated on different topologies.

6.3.1 Method

We describe three sets of simulations used to test the method. The first simulation used two genes generated on the same tree, and the second simulation used two genes generated on the same topology with different branch lengths. Both of these simulations tested the level of the test. The third simulation used two genes generated on two different trees.

In the first simulation, the topology was generated on ten taxa according to a Yule distribution (Yule, 1925). The branch lengths were chosen from Gamma distributions with shape parameters 2 (internal edges) and 5 (external edges), and scale parameter 10. The amino acids were simulated according a WAG model (Whelan and Goldman, 2001), using Seq-Gen (Rambaut and Grassly, 1997). The rates across sites were modelled by a Gamma distribution.

In the second set of simulations we generated trees with different branch lengths. These edge lengths were then modified before simulating the second gene by multiplying the length used for the simulation of the first gene (t_1) by $\exp(u\theta)$ where u is uniformly distributed between -1 and 1, and θ is a parameter describing the extent to which edge lengths differed between genes. We used $\theta = 0.1, 0.5, 2$, and 5.

In the final set of simulations we needed to generate trees with slightly different topologies. A series of subtree prune and regraft (SPR) operations was used to rearrange a starting topology to produce a new gene topology. The number of SPR operations between trees was two, four, or eight.

6.3.2 Results

Same topology, same branch lengths

In these sets, we varied the number of genes. If the method was statistically valid then the true topology would have been in the confidence set 95% of the time and the set of candidate trees would get smaller as the number of genes increased.

As the number of genes increased, the number of trees in the confidence set decreased. The level showed us that in over 98% of cases the true topology was within the confidence

# of genes	# of Reps	Mean # distinct trees 95% conf.	Level set size
1	100	47.9	0.99
2	100	11.3	0.98
5	100	6.0	0.99
10	100	3.8	1.00

Table 6.1: Result on the confidence set when varying the number of genes, one topology, fixed branch lengths. Level is the frequency with which the true topology is in the confidence set.

set. Therefore, the test was a bit conservative.

Same topology, varying branch lengths

In these sets, we varied the branch lengths. We first created a topology with branch lengths and simulated the first alignment. The second alignment was built on the same topology, but with adjusted branch lengths.

θ parameter	# of Reps	Mean # distinct trees 95% conf.	Level set size
0.0	100	11.3	0.98
0.1	100	15.2	0.99
0.5	100	16.5	0.99
2	100	13.4	0.98
5	100	18.1	0.76

Table 6.2: Results on the confidence set when varying the branch lengths, one topology, varying branch lengths. Level is the frequency with which the true topology is in the confidence set.

As the variation in the branch lengths between topologies increased, the mean number of trees in the confidence set was stable, ranging between 11 and 18. The number of times the true tree was in the confidence set declined dramatically when the tree branch lengths were very different in the second tree. Once the θ parameter was five, only 76% of the confidence sets contained the true tree. This might have been because the second tree had branch lengths which make phylogenetic analysis difficult (that is some branches may have been very short, making resolving the tree difficult).

Varying topologies

In these sets we varied the true gene trees. We would expect that, in the majority of replicates, the confidence set would not have any trees, since the two genes have evolved

on different topologies.

SPR distance	# of Reps	# true topologies in conf. set 0	1	2	# non-empty confidence set with no true trees
2	100	83	15	2	4
4	100	96	3	1	0
8	100	94	6	0	2

Table 6.3: Results on the confidence sets with two topologies.

The proportion of empty confidence sets increased as the SPR distance increased. When the SPR distance was two, 83% of replicates had an empty confidence set, while when the SPR distance was higher (four or eight), over 90% of replicates had an empty confidence set.

6.3.3 Discussion of simulation results

There are three important attributes of our method: when the genes are based on one topology, that the confidence set is not empty; the true topology is often in that set; and when the true gene trees are different, the confidence set is usually empty.

The size of the confidence set should be small when we have good evidence in favour of a single common topology. This was most clearly demonstrated as we increased the number of genes while keeping the topology and branch lengths unchanged. Under these conditions as the number of genes was increased the size of the confidence set decreased, but remained non-empty.

The size of the confidence set remained fairly constant as the branch lengths were modified, showing that the method is robust to different genes having differing evolutionary rates. However, once the branch lengths became very different, the true topology was sometimes not in the confidence set at all. The method is not robust to considerable changes in the branch lengths.

In our simulations with two topologies the sets were often empty, and as the expected differences between the topologies grew we saw an increase in the proportion of empty sets. This was pleasing. We would expect that if we used further topologies that were different, then the confidence set would almost always be empty.

6.4 Case study: Tiger Moths

Ratcliffe and Nydam (2008) published a study on tiger moths using phylogenetic clusters and several variables (the types of clicks the moths are capable of making, their colouring, and the extent to which the species is nocturnal) to investigate the evolution of signaling. They sequenced one mitochondrial gene (*cytochrome oxidase I*, COI) and two nuclear genes (*elongation factor 1a*, EF1a, and *wingless*) from 26 closely related moths and the out-group species *Lymantria Dispar*.

We applied our congruence test to their data set and the confidence set was empty.

An empty confidence set implies that the genes did not evolve on the same topology. This is particularly interesting when we compare this result with that of Section 5.2.1. There, we found that the data was relatively tree-like and that the amount of conflict was small. This result suggests that, while the conflict is small, it is significant.

6.5 Discussion

In this chapter, we developed a test for incongruence. We developed an efficient method of finding a confidence set of topologies that contains the true topologies most of the time; furthermore, as the evidence increased (with an increased number of genes), the size of the confidence set decreased. When there was incongruence, the sets were often empty.

Stouffel's method provides a way to combine p-values based on a single gene and single topology. Using this method we can test for congruence by testing whether it is plausible that all the genes come from the same topology, and our results show that the method is reliable and robust.

7

Recombination breakpoint detection

7.1 Background and motivation

The final two chapters of this thesis look at recombination. In this chapter, we discuss our development of a method for testing for the location of a recombination event.

Recombination results in one strand of DNA having inherited material from more than one parent or source. The process has been extensively studied, and recombination is one of the main reasons an assumption of tree-like descent may not hold. Here, we focus on intragenic recombination; that is, recombination that occurs within genes rather than between genes. Biologists are interested in methods that test for the presence or absence of recombination, estimate the rate of recombination, or estimate the segment(s) of DNA involved in a recombination event.

In time series analysis, the term **structural break** refers to the boundary of a stationary process. The interpretation is that the data generating process in the time before the

structural break, and the data-generating process in the time after the structural break, are different. Most standard time series analyses first check for structural breaks.

The recombination detection methods we introduce here apply structural break detection methods from time series analysis to the analysis of genetic recombination. We consider the site positions as an ordering analogous to time points in a time series, and we apply two structural break methods from time series analysis to look for recombination events. We introduce these two methods below.

Atheoretical regression trees for structural break detection

The Atheoretical Regression Trees (ART) technique was introduced by Cappelli et al. (2008) and is based on Fisher's method of optimisation (Fisher, 1958). The method calculates the deviance score for every allowable partition. The deviance score of a structural break is given by

$$SS(h) = [SS(h_l) + SS(h_r)], \tag{7.1}$$

where $SS(h)$ is the sum of squares associated with a structural break at position h, $SS(h_l)$ is the sum of squares difference from the mean of the left hand segment, and $SS(h_r)$ is the sum of squares difference from the mean of the right-hand side.

The algorithm creates nested partitions of segments with the same mean. The resulting tree of nested partitions is then trimmed using a pruning algorithm.

Bai and Perron structural break detection

The breakpoint detection method of Bai and Perron (1998, 2003), hereafter referred to as BP, relies on the calculation of an upper triangular matrix of sum of squared residuals. The matrix is indexed by date, or in our case, site number.

The entries in this matrix are calculated subject to three conditions: there is a prespecified minimum distance between two breaks, h; if the series has m breaks then $m-1$ breaks must be able to fit inside the largest partition; and a new segment cannot start before observation h.

The method uses dynamic programming to calculate the optimal partitions given a prespecified number of breakpoints.

7.2 Existing recombination breakpoint analysis methods

In this section we review methods for detecting the boundaries of recombination events (breakpoints). We present the techniques in two classes: sliding window methods, and full alignment modelling methods.

7.2.1 Recombination detection using a sliding window

Sliding window methods examine contiguous subsets of sites for particular attributes that indicate recombination. There are two types of sliding window methods. The first set of methods looks for evidence that the data-generating process on the sites in the left-hand half of the window is different from the data-generating process on sites in the right-hand half of the window. The second set of methods looks for evidence that the subset of sites within the window has different properties from the entire alignment.

The first set of methods is based on the assumption that summary statistics can indicate when the topology for the sites from the first half of the window is different from the topology for the sites from the second half of the window. As the window moves along the alignment across a breakpoint, the signal will grow until an equal number of sites within the window come from two different topologies, and then fade. The peaks are usually tested for significance using permutation testing or Monte Carlo simulations.

Methods in this class include those of McGuire et al. (1997), who compared topologies on the left and right using distance based methods and sum of squares differences; Husmeier and Wright (2001b), who compared topologies on the left and right using maximum likelihood score differences; and Husmeier et al. (2005), who compared the set of topologies on the left with the set of topologies on the right using likelihood and Robinson-Foulds distances (Robinson and Foulds, 1981).

The method of Smith (1992) compared the proportion of variable sites before and after the proposed break point and the significance of the difference is assessed by Monte Carlo simulations or a permutation test. The key difference is that the number of sites on either side of the proposed breakpoint does not need to be the same, and therefore two sliders

move along the alignment creating the left hand and right hand sides of windows with varying width.

The second set of sliding window methods looks for signs that the subset of sites has different properties from the entire alignment.

The first subgroup of these methods are based on the assumptions that most of the alignment comes from one topology, and that this topology is the one inferred by the phylogenetic method of choice. The methods then compare the topology on a small block of sites to the global alignment in order to identify the regions where the phylogenetic signal is different.

Hein (1993) found the most parsimonious tree on the global alignment and compared this to local trees built using maximum parsimony. Grassly and Holmes (1997) found the maximum likelihood tree on the global alignment and compared the likelihood score on this tree to the score on the maximum likelihood tree built on the sites within the sliding window. Archibald and Roger (2002) used a likelihood ratio test which compared the maximum likelihood tree on the global alignment to the maximum likelihood tree on the sites within the sliding window.

Two more recent methods were based on triplets; that is, sets of three sequences. Martin and Rybicki (2000) compared informative sites in user-defined triplets. Using a sliding window, they looked for regions where there was a swap in the two closest sequences, and compared it to the probability of observing the change by chance. Hao (2010) modified the method to compare consensus sequences rather than all triplets of sequences.

Other methods look at changes in topology as the sliding window is moved along the alignment without reference to the global alignment. These methods will infer recombination whenever there is a switch in topology.

Gibbs et al. (2000) counted several types of informative sites on sets of three taxa and a random sequence. By permuting the fourth taxa they were able to calculate standard normal Z-scores for the counts. Z-scores with a magnitude of three or more indicated recombination candidates.

Boni et al. (2007) considered three sequences at a time with the explicit requirement that the first two be ancestral to the third. Using the informative sites, they created a binary sequence which denoted which of the two ancestral sequences was closer. Using

similarities between this sequence and a random walk, they developed an exact expression for the probability of observing that pattern, given the probability of observing a state and allowing for one or two breakpoints. As an exact method it is very fast.

Bruen and Poss (2007) looked for changes in the p-value from the PHI test (Bruen et al., 2006). Small p-values indicated recombination, and therefore the sliding window allowed them to find the edges of the recombination event.

Lemey et al. (2009) developed recombination detection by quartet scanning. There are three possible unrooted topologies on four taxa and for each set of four taxa the relative supports for the three topologies can be calculated by statistical geometry. Using a sliding window approach, the changes in support can be detected and tested for statistical significance.

With all sliding window methods there is a trade off between a large window size, which may give rise to a stable behaviour; and a small window size, which will show the boundary as a more extreme value of the test statistic.

7.2.2 Recombination detection using a full alignment model

In this section we discuss full alignment models. These models also recognise that recombination creates a mosaic of origins in the alignment, and fit a model that includes the recombination events. In this way, some of the methods are able to estimate other evolutionary parameters simultaneously. We review four schools of approaches.

McGuire et al. (2000) focused on finding changes in topologies using a hidden Markov model. Using four taxa, they developed a model that tolerated small segments of sites which had low likelihood scores but gave rise to a switch in topologies when the sites were sufficiently different. Using a Bayesian approach and likelihood calculations, McGuire et al. (2000) calculated posterior probabilities for the breakpoints. Husmeier and Wright (2001a) extended the work by optimising the joint topologies and recombination rate rather than fixing them in advance.

Suchard et al. (2003) extended standard phylogenetic Bayesian analysis to sample and estimate the topologies and the location of the breakpoints alongside standard evolutionary parameters. Later, Minn et al. (2005) further extended the method and separated out

the estimation of the locations of substitution parameters changes and topology changes. Fang et al. (2007) produced an implementation that is very efficient and quick to run, but can only handle up to eight taxa (Martins et al., 2008).

The Genetic Algorithm Recombination Detection method or GARD (Pond et al., 2006) is actually two methods. The first GARD method screens for a single breakpoint by inferring a neighbor-joining tree for the global alignment, with the parameters estimated by maximum likelihood, then for each possible breakpoint (there is a potential breakpoint between each variable site) neighbor-joining trees are estimated on either side and the AIC calculated based on the original maximum likelihood parameters. Recombination is inferred when the AIC of the global alignment is greater than the AIC of the alignment with a breakpoint. The second GARD method screens for multiple breakpoints by fixing the number of breakpoints and iterating the procedure. The method uses a genetic algorithm to create new individuals who are recombinants based on the potential breakpoints and existing individuals. Tools from model averaging and the AIC values are used to infer the breakpoints.

The methods of Etherington et al. (2005) and Maydt and Lengauer (2006) form recombinant sequences from other sequences in the alignment. The method of Etherington et al. (2005) relied on three parameters: two distance thresholds specifying the genetic distance by which a sequence must change in order to be considered a recombinant; and the maximum number of sequences able to contribute to a recombination strand. Maydt and Lengauer (2006) took each sequence in turn and estimated the cost of forming it from the other available sequences. This cost incorporated the relative cost of explaining the sequence by mutation and recombination. The sequences with low cost given recombination are the output candidate recombinants. Both methods returned the suggested position of the recombination event and identify the taxa involved. They also implicitly make the assumption that if a population sample contains recombinants, then it contains the parental sequences as well.

Modelling the full alignment gives a set of breakpoints that takes into account the rest of the alignment, and sometimes other evolutionary parameters.

7.2.3 Desirable properties of recombination detection methods

Modern DNA alignments can be very long, up to entire chromosomes. Therefore, recombination detection methods must be able to search for recombination events on this scale. Many of the sliding window methods could be used on these long sequences as they scale linearly with sequence length. Many of the full alignment methods are not scalable in this way. Therefore, there is a need for full alignment methods with can accommodate long alignments.

Furthermore, there is a need for methods which are powerful even in the presence of high mutation rates. GARD (Pond et al., 2006) is a successful method but it performs best when the mutation rate is low. Therefore, there is a need for recombination breakpoint detection methods which are robust.

7.3 Input series for recombination breakpoint detection

Our strategy is to derive an analogue of a time series from the alignment and then apply the two recombination breakpoint detection methods: ART, and Bai and Perron (BP).

We used two input series for our recombination breakpoint detection: the influence function, and a distance based splits approach. We describe each in turn, below.

7.3.1 The influence function series

Bar-Hen et al. (2008) introduced the influence function to phylogenetics. The method quantifies the influence of each site on the tree likelihood calculation. They define the influence function as

$$IF_{S,F_n}(\mathbf{X}_h) = (n-1)(l_T(\theta_T|\mathbf{X}) - l_{T^{(h)}}(\theta_{T^{(h)}}|\mathbf{X}^{(h)})) \tag{7.2}$$

where $\mathbf{X}$ is the full alignment, $\mathbf{X}^{(h)}$ is the alignment with the h^{th} position removed, $l(\theta_T|\mathbf{x})$

is the tree likelihood, θ is the maximum likelihood parameters, and n is the number of sites in the alignment. The influence function measures influence as the difference in likelihoods when a single site is removed. Large negative influence values imply that the site does not support the maximum likelihood tree. They used this method to rank sites and create more robust topologies by removing a few outlier sites with large negative values.

The influence function has the potential to be used as an input series for breakpoint detection methods. It has large negative values where the maximum likelihood tree does not fit well (and removing the sites would improve the fit), and we look for clusters of these large negative values. We expect the mean of this series to contain information on recombination.

This is the procedure used to study breakpoint detection using the influence function. For each alignment:

1. Calculate the maximum likelihood tree on the full alignment.
2. Calculate the maximum likelihood tree for alignments with a single site removed.
3. Calculate the influence function.
4. Apply a breakpoint detection method to the resulting influence function series.

The program PhyML (Guindon and Gascuel, 2003) was used to estimate the maximum likelihood trees and compute likelihood scores.

7.3.2 Distance and splits based series

Our second method was very similar to that of the influence function-based approach. We used a series based on comparing a neighbor-joining tree built on the whole alignment with the neighbor-joining trees built on the alignments with one site removed.

Trees have $2n - 3$ branches or splits where n is the number of taxa. For each split in the neighbor-joining tree we took the difference between the distances fitted to the whole alignment, and the distances fitted with one site removed, and got $2n - 3$ series.

We took the mean of each of these series. The series with the largest mean was denoted as

the base series; that is, the series to which other series of differences will be added.

In order to determine how the other series were added to the base series we first looked at the correlation matrix. We added the other series to the base series, but first weighted the differences by the correlation coefficient between the base series and the series in question. Therefore each element in the input series was a weighted sum of the $2n-3$ differences where the weights were the correlation coefficients, that is

$$SD(X_h) = \sum_{i=1}^{2n-3} w_i \text{diff}_i^h \tag{7.3}$$

where SD is the distance and splits based series, w_i is the weighting from the correlation matrix and diff^h is the vector of differences in branch lengths at site h. Adding the series together gives a series with a higher variance than a single set of differences alone, hopefully increasing the effect of a change in the phylogenetic signal on the series SD.

When there is no recombination SD should be, on average, zero. In the presence of recombination it may be positive or negative and significantly different from zero.

This procedure created a single series which was used as input for the breakpoint detection methods ART and BP.

We expected this series to change significantly under a new evolutionary regime. This simple change would be picked up by a breakpoint detection method if the change was large enough.

We used the following procedure to study breakpoint detection using the branch or split weight approach. For each alignment:

1. Calculate the neighbor-joining tree on the full alignment.
2. Calculate the neighbor-joining tree for all alignments with a single site removed.
3. Calculate the $2n-3$ series of differences in the estimated branch weights.
4. Find the series with the largest mean difference; this is the base series.
5. Calculate the correlation matrix of the differences.

6. Calculate the input series by adding the weighted differences to the base series, where the weights are the correlation coefficients.

7. Apply breakpoint detection methods to the resulting series.

The methods in this chapter were implemented in R (R Development Core Team, 2009).

7.4 Data

We used two main sources of data to test our methods. The first source was the data sets from Pond et al. (2006) here referred to as A, B, C and D. They are described below in detail. The second set of data came from merging two tree-based alignments simulated using the framework of Hudson (1983). This allowed us to explore a greater range of sequence divergence rates (often simplified to divergence rates). These data sets are referred to as E and F, and they are also described below in detail.

Description of the six data sets

A Pond et al. (2006) simulated ten data sets, each with a specified number of breakpoints and divergence rate. The breakpoints were at different places in each of the 100 replicates. The divergence rates were 5% and 25%, and the number of breakpoints were zero, one, two, four, and eight. Each alignment had eight taxa.

B The second set from Pond et al. (2006) was the neutral scenario, which was based on merging two alignments, one 400 codons long and the other 100 codons long. There were 32 taxa in each replicate. Pond et al. (2006) notes that this scenario is designed to mimic a hotspot in an area with a high mutation rate.

C Scenario one data sets from Pond et al. (2006) had eight taxa and two recombination points. The alignments were 2,500 base pairs long, and the segment from the 1000th to 1400th base pair was based on a tree with a single taxon moved to a different location. The data was generated based on the HKY85 model and had a divergence rate of 1%.

D Scenario two data sets from Pond et al. (2006) had eight taxa and three recombination points. The alignments were 2,500 base pairs long. The segment from base

pair 600 to the base pair 1100 had a three-taxa clade moved, and the segment from 1100th base pair to the 1900th base pair had a two-taxa clade moved.

E We also tested our method on alignments simulated on trees using Hudson (1983). We generated a set of alignments where the first part of the alignment was simulated on one tree and merged with sites generated on another tree. This method is referred to as the 'tree-plus-tree' method. The two alignments were of equal length. We used the divergence rates of 1%, 5%, and 10%, and sequence lengths of 1000 and 2000. All data sets had ten taxa.

F We generated alignments following the same protocol as E, except the alignments were not of equal length. Instead, 80% of sites were from one alignment and the remaining 20% from another alignment. For the merged alignments where one is four times longer than the other, we used divergence rates of 1%, 5%, 10%, 20%, and 50% and a sequence length of 1000.

7.5 Results

A breakpoint was considered to be detected if the breakpoint detection method returned a breakpoint within 50 base pairs of a true breakpoint. This cutoff was arbitrary, as neither method provided a confidence interval to guide our choice. While we did not carry out any sensitivity testing this would have assisted us in determining if this 50 base pair limit is acceptable.

7.5.1 Results for the influence function based approach

The data sets without breakpoints were used to estimate the level of the test. The false positive rate as measured by data sets A (the Coalescent data sets) was very low. The BP detection method with a 5% divergence rate had the highest false positive rate, with four of the 100 data sets returning at least one breakpoint; ART had two of the 100 data sets return a false positive. When the divergence was 25% there were three and two false positive detections for BP and ART respectively. This level of false discovery was reasonable.

The remainder of the data sets from A had one, two, four, or eight breakpoints. With a single true recombination event and a low divergence rate (that is a 5% divergence rate) the true breakpoint was never detected by either ART and BP. However, seven and 12 of the data sets reported a falsely detected recombination event for ART and BP respectively. For the higher divergence rate (that is the 25% divergence rate), the true breakpoint was detected two and three times by ART and BP respectively, and there were fewer false positives.

When there was more than one breakpoint, the methods detected more than a single breakpoint only once. The frequency with which one of the breakpoints was detected was very low, and at its highest, a correct breakpoint was located in only 11 of the 100 data sets.

Data set	Seq. Length	# True breakpoints	Div. level	Reps	Method	Breakpoints correctly identified			Breakpoints incorrectly identified		
						0	1	2 or more	0	1	2 or more
A	3000	One	5%	100	ART	100	0	-	93	7	0
					BP	100	0	-	89	7	4
			25%	100	ART	98	2	-	96	4	0
					BP	97	3	-	93	4	3
		Two	5%	100	ART	99	1	0	97	3	0
					BP	99	1	0	95	3	2
			25%	100	ART	96	4	0	100	0	0
					BP	96	4	0	99	0	1
		Four	5%	100	ART	96	4	0	96	4	0
					BP	96	4	0	92	3	5
			25%	100	ART	90	10	0	97	3	0
					BP	89	11	0	95	4	1
		Eight	5%	100	ART	97	3	0	95	4	1
					BP	95	5	0	87	9	4
			25%	100	ART	93	7	0	88	12	0
					BP	91	8	1	86	13	1
B	1500	One		100	ART	1	99	-	92	7	1
					BP	1	99	0	54	26	20
C	2400	Two		100	ART	100	0	0	100	0	0
					BP	100	0	0	91	1	10
D	2400	Three		100	ART	100	0	0	100	0	0
					BP	97	3	0	92	3	5
E	1000	One	1%	200	ART	97.5	2.5	-	91	8.5	0.5
					BP	99	1	-	85.5	4.5	10
			5%	200	ART	87.5	12.5	-	82.5	16	1.5
					BP	91	9	-	90	7.5	2.5
			10%	200	ART	92	18	-	86	12.5	1.5
					BP	77.5	22.5	-	92	6	2
	2000	One	1%	200	ART	99.5	1	-	98.5	1.5	0
					BP	99	1	-	77	5	18
			5%	200	ART	89	11	-	94.5	5.5	0
					BP	86.5	13.5	-	82.5	9	8.5
			10%	200	ART	80.5	19.5	-	88.5	11.5	0
					BP	78	22	-	81	14.5	4.5
F	1000	One	1%	200	ART	95	5	-	80	16.5	3.5
					BP	94.5	5.5	-	70	25	5
			5%	200	ART	30	70	-	69.5	26	4.5
					BP	29	71	-	71.5	24	4.5
			10%	200	ART	46	54	-	66	28.5	5.5
					BP	37.5	62.5	-	74.5	17	9.5
			20%	200	ART	62.5	37.5	-	72	23	5
					BP	57.5	42.5	-	81	16	3
			50%	200	ART	82.5	17.5	-	74.5	20.5	5
					BP	77.5	22.5	-	78	17.5	4.5

Table 7.1: Using ART and BP and the influence function to detect recombination breakpoints

The influence function breakpoint detection method worked very well on the data sets B. It reported the correct breakpoint in nearly all of the data sets. The main difference

between ART and BP was the level of spurious breaks reported. BP reported a great deal more spurious breaks.

The influence function-based method performed very poorly on the data sets C, and did not report the correct breakpoint once. The BP method reported some spurious breakpoints.

The method also performed very poorly on data from data sets D. The BP method reported some suprious breaks, but neither method detected any of the true breakpoints.

For data sets E there was a single breakpoint, located in the middle. As the divergence rate increased, the rate of detection increased. At the highest divergence rate (10%), the correct breakpoint was located in about 20% of datasets. As the divergence rate increased, we also noted an increase in false positive detection; the rate was nearly the same as the rate of accurate detection. This was true of both ART and BP detection methods. Sequence length appears not to influence performance.

For data sets F, 80% of the alignment was based on one tree and 20% on another. When the divergence rate was 10%, the true breakpoint was detected in about 40% percent of the data sets. Once the divergence rate was 50%, about 80% of the data sets reported the true breakpoint. The rate of false detection was relatively small, and in about 15% of cases an additional breakpoint was falsely reported (when the sequence divergence is 50%). The performance of ART and BP methods was similar, with ART generally detecting a few more breakpoints and reporting fewer false positives.

7.5.2 Results for the distance based splits approach

We applied only the ART method of breakpoint detection to the data sets A, C, and D. BP is much more computationally intensive, and in the first set of investigations based on the influence function the ART procedure slightly outperformed the BP procedure.

The distance based splits approach was applied to data sets A. The set with no recombination events could be used to investigate the level. Of the 100 data sets with a divergence rate of 25%, only one had a single false detection, and all of the data sets with a divergence rate of 5% did not have any false detections. Therefore, the level appears to be very low

under these parameters.

Data set	Seq. Length	# True breakpoints	Div. level	Reps	Method	Breakpoints correctly identified			Breakpoints incorrectly identified		
						0	1	2 or more	0	1	2 or more
A	3000	One	5%	100	ART	98	2	-	98	2	0
			25%	100	ART	95	5	-	99	1	0
		Two	5%	100	ART	98	2	0	96	3	1
			25%	100	ART	89	9	2	99	1	0
		Four	5%	100	ART	96	4	0	92	7	1
			25%	100	ART	86	12	2	97	3	0
		Eight	5%	100	ART	84	16	0	91	8	1
			25%	100	ART	79	21	0	93	7	0
B	1500	One		100	ART	84	16	0	89	11	0
					BP	82	18	0	87	11	2
C	2400	Two		100	ART	100	0	0	98	2	0
D	2400	Three		100	ART	100	0	-	99	1	0
E	1000	One	1%	200	ART	94.5	5.5	-	72.5	14	13.5
					BP	93.5	6.5	-	55.5	12.5	32
			5%	200	ART	77	23	-	74	19	7
					BP	78	22	-	69.5	16	14.5
			10%	200	ART	62.5	37.5	-	76	18	6
					BP	68.5	31.5	-	72	14	14
			20%	200	ART	45	55	-	83.5	14.5	2
					BP	51.5	49.5	-	76	17	7
			50%	200	ART	40.5	59.5	-	81.5	17	1.5
					BP	44	56	-	87.5	11	1.5
	2000	One	1%	200	ART	95.5	4.5	-	81	6	3
					BP	87.5	12.5	-	50	12.5	37.5
			5%	200	ART	72.5	27.5	-	86	12	2
					BP	71.5	28.5	-	57	13.5	29.5
			10%	200	ART	68	32	-	85	14.5	0.5
					BP	67	33	-	56.5	16.5	27
			20%	200	ART	56	44	-	92.5	7.5	0
					BP	55.5	45.5	-	81	12.5	6.5
			50%	200	ART	45	55	-	89	11	0
					BP	45.5	55.5	-	78.5	16.5	5
F	1000	One	1%	200	ART	95.5	4.5	-	80	13.5	6.5
					BP	89.5	10.5	-	63.5	13.5	23
			5%	200	ART	81.5	18.5	-	76	18.5	5.5
					BP	83	17	-	25	12.5	12.5
			10%	200	ART	68	32	-	75	18.5	7
					BP	71	29	-	71	17	12
			20%	200	ART	54.5	45.5	-	76.5	25	8.5
					BP	60.5	39.5	-	64	23.5	12.5
			50%	200	ART	53	47	-	83	22.5	4.5
					BP	58	42	-	81.5	12.5	6
	2000	One	1%	200	ART	93.5	6.5	-	87	9	4
					BP	88	12	-	37.5	21.5	41
			5%	200	ART	81	19	-	85	11.5	3.5
					BP	75.5	24.5	-	51	14	35
			10%	200	ART	64	36	-	86	11.5	2.5
					BP	62. 5	37.5	-	69.5	13	17.5
			20%	200	ART	62	38	-	91.5	8	0.5
					BP	60	40	-	74	10	16
			50%	200	ART	52.5	47.5	-	86.5	12.5	1
					BP	47.5	52.5	-	74.5	15	10.5

Table 7.2: Using ART and BP and the distance-based splits approach to detect recombination breakpoints

Studying the results reported from using coalescent-based simulated data with two, four, and eight breakpoints, we found that more breaks were reported when the divergence rate was higher; that is, 25% instead of 5%. However, even for the higher divergence rate, the proportion of correctly reported breakpoints was very low. The data set which reported the most correct detections was the data set with eight breakpoints, and it reported one break correctly in 21 of 100 of the data sets and did not report any of the remaining seven breakpoints. Therefore, the detection rate is very low.

The splits based detection method did not report many of the breakpoints for data sets

B. We used both ART and BP and they reported the correct breakpoint 16 and 18 times respectively. Both method reported at least one false positive in ten percent of the replicates.

The splits based detection method did not report any of the true breakpoints for data sets C and D. This method did report one or two false positives.

When two series of equal lengths were merged (data sets E) then in over 40% of data sets the true breakpoint was detected provided the divergence rate was 20% or 50%.

When one series was four times longer than the other (data sets F), we expected a much higher rate of detection, but the performance of the method was mediocre, reaching only 52% of data sets reporting the true break.

In general, BP found a similar number of breakpoints to ART; however, it often had a much higher false positive rate.

7.6 Discussion

Comparing the influence function-based approach with the split-based approach, we see that neither method consistently outperformed the other.

The performance of our methods compared with GARD on data sets A, B, C and D

Both of our methods only infrequently reported any breakpoints when the data sets were built using a coalescent model and a fixed number of recombination events (data sets A). In the majority of the instance that a breakpoint was reported correctly, only one out of up to eight breakpoints was reported (with one exception). Our method did not perform nearly as well as GARD, though GARD also had considerable difficulty finding multiple breakpoints. The authors note that the recombination signal is "quickly saturated for small alignments (8 sequences)". This implies it would be very difficult for any method to consistently detect all the breakpoints.

The data sets B (neutral scenario) was designed to replicate hotspots and as such it had a high mutation rate. Our influence function method performed very well, with this type of

data finding the one true breakpoint in nearly all of the 100 replicates. Our distance-based splits approach seldom reported a breakpoint correctly.

For scenarios one and two (data sets C and D), both our methods seldom reported a breakpoint, while GARD performed very well on these two data sets.

Pond et al. (2006) found that GARD reported a much higher false positive rate for data sets B than either of data sets C or D. Our false positive rate was very low, while our detection rate was very high when using the influence function-based approach for data sets B. Therefore, it seems our method outperformed GARD under these settings. This suggests that using the influence function to detect recombination may be most useful for hotspots.

The performance of our methods on data sets E and F

Merging two trees is an extreme way of simulating recombination, and this does not accurately model likely site patterns seen in recombinant alignments. Typically, only a small number of taxa are influenced by the recombination event; perhaps only a single taxon. Therefore, we expect that using the 'tree-plus-tree' method for simulating and testing a recombination breakpoint method would overstate the power of the method.

Using the influence function, the correct breakpoint was only detected a few times when the merged alignments were of equal length. The performance of the method improved significantly when one of the regimes dominated the series, as seen by the improved performance of the alignments with 80% of sites coming from one tree and the remaining 20% of sites coming from the second tree.

The distance-based splits approach performed better than the influence function approach when the merged alignments were of equal lengths, but was worse than the influence function approach when the alignments had 80% of sites coming from one tree and 20% of sites coming from another.

General observations

The influence function as input into the breakpoint detection method was based on a reasonable assumption. The influence function gives rise to a large negative value when removing the site results in an improved fit. These large negative values therefore indicated which sites did not fit, and it was a natural progression to consider detecting evolutionary

regimes by testing for groupings of large negative influence function values.

The common feature of most of the data sets from Pond et al. (2006) was the low divergence rate. When the divergence rate was low, the proportion of non-constant sites was smaller and the subset of sites which did not fit on the maximum likelihood tree was even smaller still. This reduced the chance of observing a group of large negative influence values, and consequently, the chance of detecting such a group. Our method looked for statistically significant groupings in these influence function values, and therefore required a reasonable amount of information or informative sites. Therefore, our method would be expected to improve in accuracy as the divergence rate increased, and we see this when comparing the results on a 5% and 25% divergence rate for the coalescent-based simulations. Furthermore, the rate of detection was very high for the neutral scenario, which had a high mutation rate.

If the alignment was composed of two trees, each contributing an equal number of sites, then the maximum likelihood tree on the whole alignment would reflect this mixture and many of the sites might not fit on this tree. Therefore, the poor performance observed when merging two alignments of equal length is likely to have resulted from the fact that on both sides of the break we observed large negative influence function values, and consequently the breakpoint detection method would not observe any differences in the mean of the influence function on either side of the simulated breakpoint.

The improved performance of the method on the alignment with 80% of sites coming from one tree and 20% of sites coming from the second tree was a result of the maximum likelihood tree being less influenced by the 20% of the alignment. Therefore, there was an increased chance of some site patterns producing large negative influence function values within the smaller regime; furthermore, there was an increased chance of these large values clustering in the segment of the alignment simulated under a different tree. This confirmed that our influence function-based series and a recombination breakpoint detection method showed potential in detecting hotspots.

The disadvantage of the influence-based method was that it only indicated which sites did not fit well, and lacked an ability to discern the extent to which they did not fit, or the influence the site had on the topology. However this was only a problem if the regimes were of equal lengths; when one regime dominated the series, our influence-based approach performed very well.

The splits-based approach performed better than the influence function for the coalescent data from Pond et al. (2006) and data sets E. It suggested that the method was more sensitive than the influence function-based approach. The only method we used to mix the branch lengths into a single series was weighting by the correlation matrix of the differences. We chose to weight by correlations as we felt this would exaggerate the differences. This was not enough to get the level of sensitivity required.

Neighbor-joining is one of several tree construction methods that could be used to construct a topology based series in this way. Neighbor-joining is a distance-based method, and as such, it is very efficient. When the input alignment was 1000 base pairs long the number of neighbor-joining trees estimated was 1001; one for the whole alignment and one for each site removed. Therefore, the tree topology estimation must be fast and neighbor-joining was a good choice. It seems unlikely that using an alternative tree construction method would improve the performance.

ART generally outperformed the method of Bai and Perron (1998, 2003). It reported far fewer spurious breaks, especially for alignments with highly-diverged sequences where the performance of this method is of the greatest interest.

The procedure of BP was presented originally as a method for finding the optimal breakpoint locations given the maximum number of breakpoints. The R implementation in the `strucchange` package (Zeileis et al., 2002) has a parameter which specifies the minimum distance between two breakpoints. In all the simulations we specified the minimum distance to be 25 base pairs. We did not carry out any investigations into the optimal specified minimum distance. Bai and Perron (1998) suggested to users that they run the method several times with different numbers of maximum breakpoints, as the method can be sensitive to this parameter, so additional investigations into this parameter would be useful. The BP procedure had a tendency to report higher levels of false positive breakpoints. This may potentially be reduced if the minimum distance between two breakpoints is increased.

Extensive simulations using the ART procedure found ART to be relatively insensitive to the tuning parameter which determined the strength of pruning in the atheoretical regression trees (Rea, 2008). Therefore, ART is likely to have been performing optimally and would not benefit from further refinement.

ART offers considerable benefits for longer alignments. BP becomes infeasible for alignments more than just a few thousand base pairs long, while ART has no such restriction. The ART procedure, upon detecting a breakpoint, then considers the two alignment segments separately, and as such it is very efficient, even for long alignments. It is worthwhile noting that Graham et al. (2005) detected breakpoints by splitting the alignment upon detection of a breakpoint and re-searching the fragments. This has similarities with the ART approach.

One factor that had no detectable effect on the chances of correctly detecting a true breakpoint was sequence length. The simulations based on merging two alignments had sequence lengths of 1000 and 2000 base pairs. From the performance of these two sets we were unable to predict the efficacy of the method on longer or shorter alignments. This would require further investigation.

However, if our method is to be applied to hotspot detection, it needs to be able to handle millions of base pairs, not just a few thousand. Clearly ART is the only choice. However, we have not yet tested its capabilities with very long alignments.

One factor not yet investigated is the number of taxa. The simulations based on the Pond et al. (2006) data had a varying number of taxa but these data sets all had low divergence rates, giving rise to very few correctly-reported breaks. All of the simulations carried out on alignments created by merging two alignments had ten taxa. Therefore, it would be interesting to run a further set of simulations with more taxa to provide a comparison.

The approach taken has focussed on two input time series for the breakpoint detection methods namely the influence function approach and the distances-based splits approach. These procedures could potentially be more widely applicable, as an appropriate input series is any series which has the property that different evolutionary regimes give rise to different mean levels in the series.

8

Investigating recombination using incompatibility

8.1 Background and motivation

In this chapter we once again look at recombination. Recombination creates a mosaic of origins, and therefore sites on different sides of a breakpoint will have been generated on different trees. This attribute of the site patterns has been used to develop a range of recombination tests. We used it to develop a method for detecting recombination breakpoints.

Our work relies on the concept of refined pairwise incompatibility outlined in Penny and Hendy (1986). The incompatibility score is calculated on two sites, i and j, and is given by

$$s_{i,j} = l(\chi_i, \chi_j) - (|\chi_i| - 1) - (|\chi_j| - 1), \tag{8.1}$$

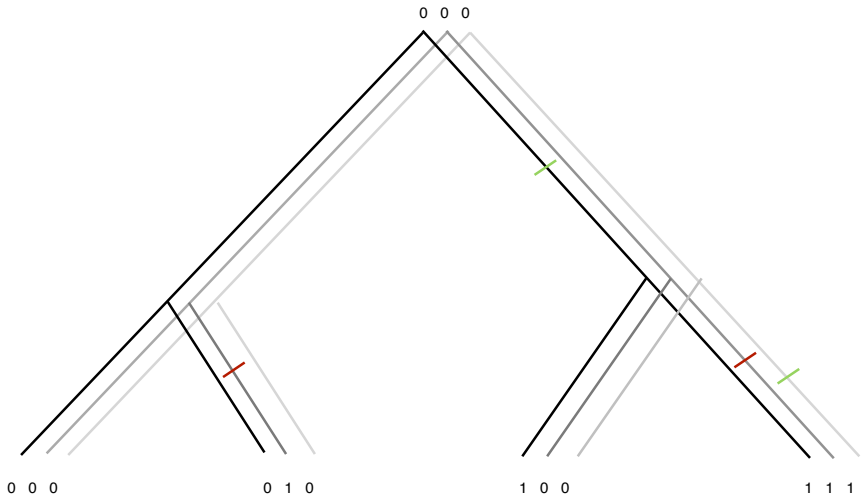

Figure 8.1: Three sites on the same topology. Red and green bars indicate mutations from a zero to a one. The middle site has a convergent evolution event, as a one has evolved independently twice at this site. This type of event can lead to incompatibility when paired with another site such as the first site; but not necessarily, as we can see by comparing the second and third sites.

where $|\chi_i|$ is the number of observed states at site i, and $l(\chi_i, \chi_j)$ is the maximum parsimony score for the two sites (the maximum parsimony score is minimum number of mutations required for both characters to evolve on the same tree). We say that two sites are compatible if $s_{i,j} = 0$, and incompatible if $s_{i,j} > 0$. The score indicates the number of additional mutations required to fit both sites onto a single tree (see Figure 8.1).

On some occasions we use the raw score $s_{i,j}$, on other we follow Camin and Sokal (1965) and Le Quesne (1969) and just use an indicator variables for where $s_{i,j}$ is zero (compatible sites) or non-zero (incompatible sites).

The assumption that incompatibility levels are higher across breakpoints led to a variety of methods for investigating recombination. Jakobsen and Easteal (1996) introduced the incompatibility matrix. The rows and columns are indexed by sites, and the squares where the pairwise incompatibility score is non-zero are darkened. Jakobsen and Easteal

(1996) developed a method which tests for the presence of recombination by comparing the observed 'neighbor similarity score'; that is, the fraction of adjacent squares of the incompatibility matrix with the same score, with the expected score based on Monte Carlo simulations. The PHI statistic (Bruen et al., 2006) is a non-parameteric test for recombination. The test is based on the idea that closer sites are more likely to be compatible than distant sites, in the presence of recombination.

Salemi et al. (2008) published an algorithm for investigating recombination using the PHI test and neighbor-net, but did not provide a formal justification for the method. The reason that this method is valid is that the distances are independent of the order in which the sites occur in the alignment, while the PHI test relies on the order. Therefore, the subgroups identified by a distance-based method (either a tree-based or network-based method) are also independent of the ordering. The combination of these two independent sources of information and the user's intuition may result in finding the source of recombination with just a few simple tests.

We developed a recombination breakpoint detection method which returned a ranked set of potential breakpoints. These candiates must be checked before a recombination breakpoint is confirmed.

We devised a small simulation-based study to investigate whether compatibility held information on the number of recombinations. Consider two sites in an alignment. If there is a breakpoint between the two, this increases the likelihood of incompatibility, since the two sites have evolved on different trees. We wanted to investigate whether multiple breakpoints, or recombinations, would lead to a higher level of incompatibility; and, conversely, whether the incompatibility score reflected the number of breakpoints between two sites.

As Figure 8.2 shows, there was an increase in the amount of incompatibility as the number of recombinations increased and as the sequence divergence rate increased.

Based on the fact that incompatibility increased across a breakpoint, we developed a linear modelling approach to finding the location of the breakpoints.

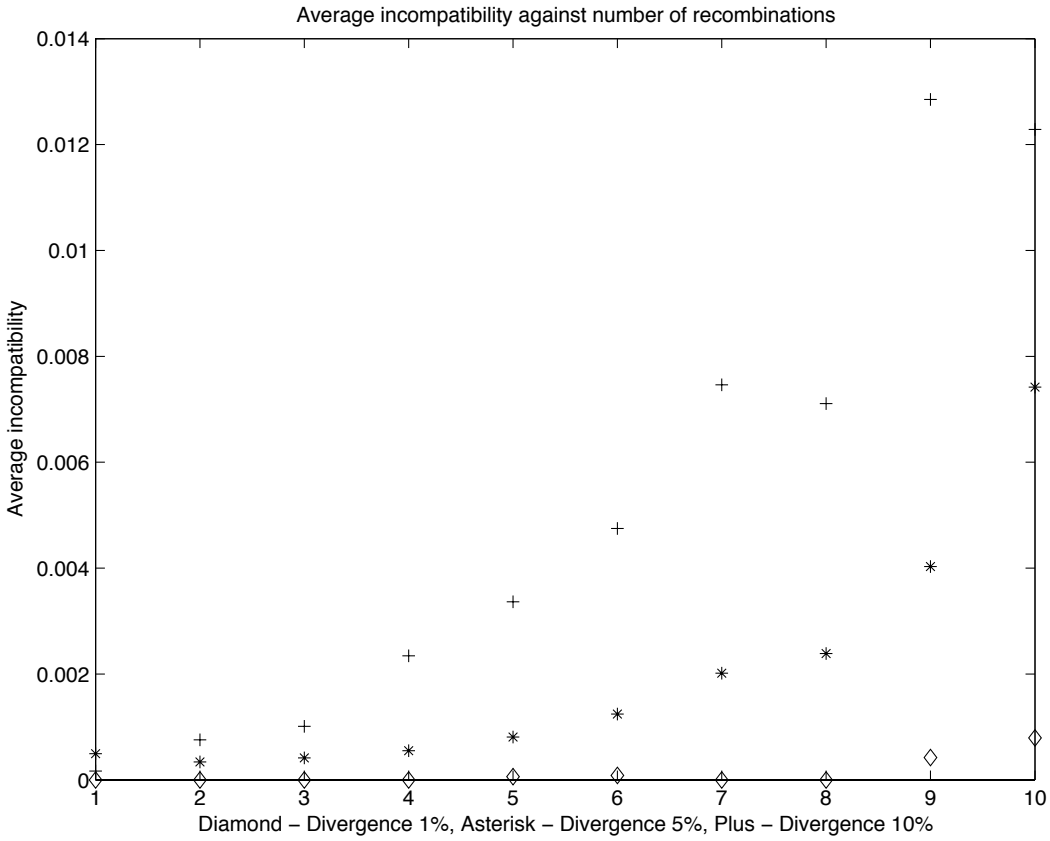

Figure 8.2: Number of recombinations with average incompatibility over a range of parameters. Data generated on alignments of length two base pairs with the ancestral recombination graph (Hudson, 1983) determining the number of recombinations between the two sites.

8.2 Using incompatibility to detect recombination breakpoints

To date, incompatibility has been primarily used as a measure to detect recombination. In this section, we report how we used it to estimate the number and locations of recombination events.

8.2.1 Breakpoint recombination model

The model we developed was based on the principle that the level of incompatibility increased linearly as the number of breakpoints separating the two sites increased. It

modelled the vector of pairwise incompatibility scores using variables which indicated which pairs were expected to have an increased level of compatibility given a specified breakpoint.

The incompatibility matrix contained the incompatibility score for every pair of sites, and the rows and columns of the matrix were indexed by sites. The matrix contained the incompatibility patterns we wanted to explain, and it could be represented as a vector using the pair of sites for the indexing.

If there was one breakpoint we expected a higher level of incompatibility when comparing pairs of sites separated by a breakpoint than with those pairs which were within a single regime. The small simulation study above showed that the average incompatibility level increased when the sites were separated by additional breakpoints (Figure 8.2).

Each breakpoint induced a split of the set of sites. We could therefore apply split-based methods and models from Chapter 2; except that the 'taxa' were the sites. The splits matrix, $\mathbf{X}$, had a column for each split, and rows were indexed, in this case, by pairs of sites. The entries $\mathbf{X}_{ij;k}$ were given by

$$\mathbf{X}_{ij;k} = \begin{cases} 1 \text{ if } i \text{ and } j \text{ are on opposite sides of the breakpoint} \\ 0 \text{ if } i \text{ and } j \text{ are on the same side of the breakpoint.} \end{cases} \tag{8.2}$$

We used the set of splits that represented the $n-1$ potential breakpoints where n is the number of informative sites. Therefore, a column of the splits matrix contained ones when the two sites of the row index were on either side of the breakpoint, and zeros when the two sites were on the same side.

We used a linear regression framework to estimate the number of recombination breakpoints and their locations. Specifically,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon \tag{8.3}$$

where $\mathbf{y}$ is the vector of incompatibility scores indexed by pairs of sites and $\boldsymbol{\beta}$ is the vector of fitted coefficients.

We noted that the splits induced by the breakpoints were all contained within the splits of a phylogenetic tree (with taxa equal to the set of sites) therefore OLS estimates can

be computed in order O(N^2) time (Bryant and Waddell, 1997). However, here, we used simple linear regression.

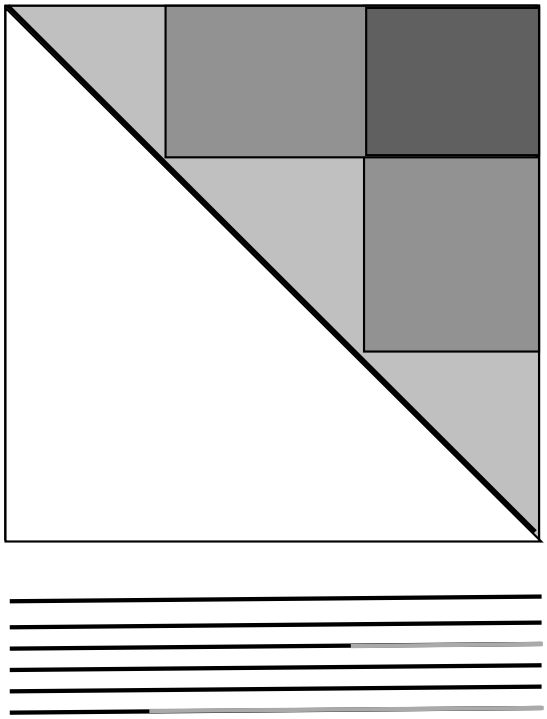

Figure 8.3: The zones of incompatibility for the alignment below. The pale grey part of the DNA strands represent recombinant strands. The palest grey represents the part of the incompatibility matrix in which the two sites are not separated by a breakpoint, the mid-tone grey represents the part of the incompatibility matrix in which the two sites are separated by a single breakpoint, and the darkest grey represents the part of the incompatibility matrix in which the two sites are separated by two breakpoints.

8.2.2 Simulation study

The simulation procedure used to test this model was based on merging two trees together. The procedure was as follows:

1. Simulate two alignments, each on a different tree, and merge them together.

2. Calculate the incompatibility score for every pair of sites.
3. Estimate the set of potential breakpoints based on the locations of the variable sites.
4. Estimate the pairs which are expected to have an increased rate of incompatibility for each breakpoint, and form $\mathbf{X}$.
5. Fit an ordinary least squares model.

We ran this simulation under four scenarios; that is with sequence lengths 400 base pairs and 800 base pairs and sequence divergence rates of 10% and 20%. We used ten taxa. The computational requirements of the current implementation meant that running this procedure for alignments longer than 800 base pairs was infeasible and running this for alignments of 800 base pairs took considerable time and memory. This is unsurprising given that the predictive vector contains all the information from the upper triangle of the incompatibility matrix and the dimensions of the problem grow quickly as the sequence length increases. However, there is potential for avoiding many of these computational difficulties by taking advantage of the structure of $\mathbf{X}$.

8.2.3 Results

		Break number of the correct breakpoint reported			
Sequence length	Sequence divergence	First	Second	Third	Fourth or more
400	10%	77	14	4	5
	20%	91	7	1	1
800	10%	76	14	6	4
	20%	90	10	0	0

Table 8.1: The percentage of times the correct breakpoint (at site 200 for alignments 400 base pairs long and at site 400 for alignments 800 base pairs long) was returned at that break number or `glmpath` ranking. The correct breakpoint was considered to be one within 20 base pairs of the true breakpoint.

The result of each replicate was a ranked list of potential breakpoints based on the order in which the R command `glmpath` (Park and Hastie, 2007) included them in the model. The command `glmpath` fits a regression model with an L1 penalty. We currently do not have a method for determining an appropriate cutoff.

The true breakpoint is in the middle (that is, at site 200 when the alignment is 400 base

pairs long, and at site 400 when the alignment is 800 base pairs long). Therefore, the frequency with which the true breakpoint is reported as the first, second, third, fourth, or later position.

The sequence divergence rate had the strongest influence on the results. When the divergence was low (10%), just over three quarters of the time the true breakpoint was the first reported breakpoint and at least 90% of the time the true breakpoint was one of the first two reported breaks. When the sequence divergence rate was high (20%), then at least 90% of the time the true breakpoint was the first reported breakpoint.

8.2.4 Case studies

We applied the method to the two shortest sequences that Posada (2002) found to have recombination: Petunia RNase and Neisseria ArgF. These plots show the locations of the breakpoints with respect to site position. The number of breakpoints displayed was chosen by the BIC criterion. These plots show that an objective measure of how many breakpoints there are is essential to the success of the method.

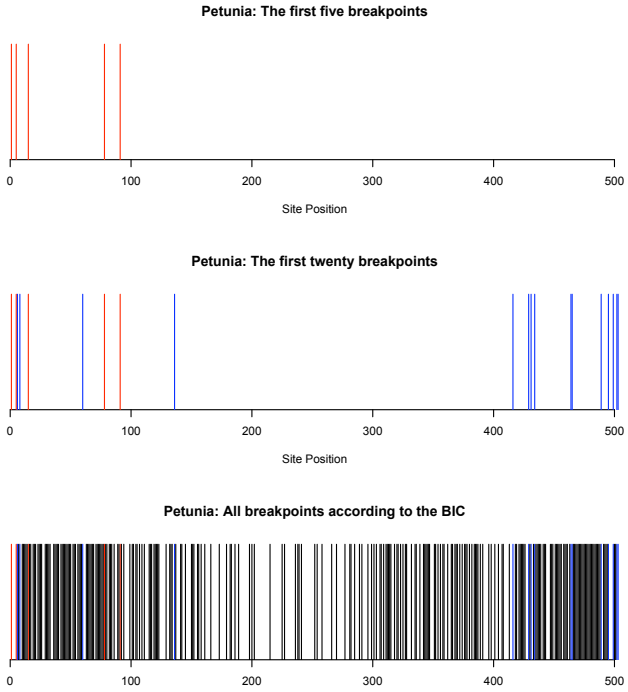

Figure 8.4: Breakpoints reported by our incompatibility breakpoint detection method.

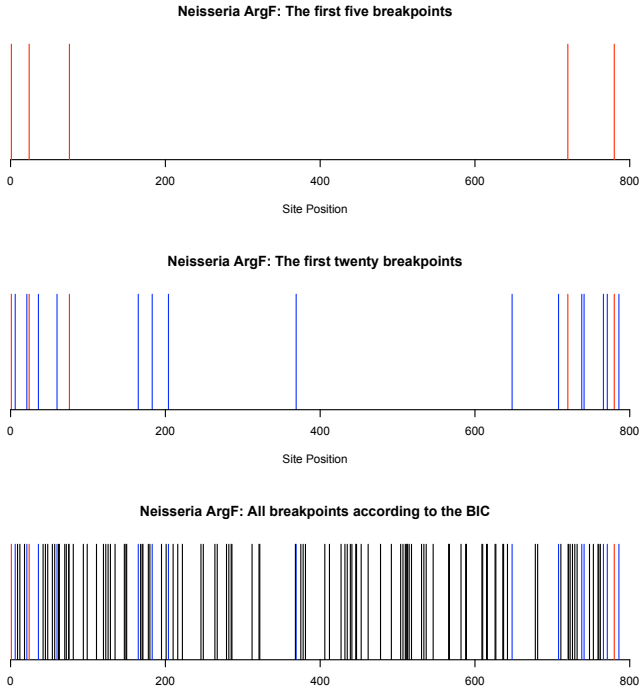

Figure 8.5: Breakpoints reported by our incompatibility breakpoint detection method.

8.2.5 Discussion

These results suggested that incompatibility, which has already been proven valuable in detecting recombination, can also provide information that will allow us to detect recombination breakpoints.

In the majority of cases, the correct breakpoint was the first or second reported breakpoint. When the sequence divergence rate was high (that is, 20%) at least 90% of the time the true breakpoint was the first reported breakpoint.

Our margin of error was 20 base pairs. Therefore when the alignment was 400 base pairs long and the true breakpoint at site 200, then if the first breakpoint was between site 180 and site 220 we said that the model reported the correct breakpoint first. If we were to narrow or widen this gap, the results would change.

Simulating sequences on two different trees and merging them exaggerated the signs of recombination by leading to high levels of incompatibility for pairs across the breakpoint. Therefore, the results of this simulation may be better than we can hope for in practice.

Our method assumed that the contributions from each breakpoint were linear and additive. These results suggested that this was a good approximation of the truth.

Unfortunately, the accurate estimation of the number of breakpoints remains an unresolved problem. The AIC and BIC pick an excessive number of breakpoints, and are consequently not suitable. These criteria pick models which incorporate the splits explaining the smaller changes because of the mutation rate, as well as the split which determined the breakpoint. Potential future work could include conditioning on the number of breakpoints or using a reversible jump MCMC to infer possible breakpoints.

The current implementation of this method is time-consuming for small sequences. This was a direct result of the dimensions of $\mathbf{y}$ and $\mathbf{X}$ being quadratic in the number of sites. Given this initial demonstration that the method has considerable potential, future work includes speeding up the calculations.

Further investigations could include a wider range of simulation conditions including running the method with the breakpoint not in the center, expanding the range of sequence divergence rates tested, and expanding the range of the number of taxa investigated.

9

Discussion

This thesis has looked at several sources of heterogeneity in phylogenetics from a statistical perspective. In the first half of the thesis we investigated visualising heterogeneity by using networks; we used split networks and consensus networks. We have looked at recombination, and in particular two ways of detecting the boundaries of a recombination event. We have also looked at interspecific heterogeneity caused by different genes evolving on different topologies. Here, we discuss the main outcomes of this work.

The first part of our work on networks centered on directly applying a statistical technique to network construction. Estimating a phylogenetic tree or network involves estimating discrete (topology) and continuous (branch length) components. Viewing splits as explanatory variables, as we did, allowed us to directly apply statistical tools to estimating the branch lengths. We applied the framework to neighbor-net (Bryant and Moulton, 2004), and as a consequence we were able to reduce some of the clutter in the networks.

One philosophical question we briefly discussed was what is the null model when considering a network; is it a network or a tree? There are arguments in favour of each approach, but the issue is far from settled. This is an issue worth further consideration.

Our chapter on testing for tree-likeness aimed to develop a test that would have distinguished between between alignments with some form of heterogeneity and alignments without heterogeneity. The main outcome of this work was that testing for tree-likeness using our information criteria approach is not possible. It was a novel approach in that we tried to develop a test for tree-likeness that did not use site ordering information.

Further work in this field could take the form of a search for a method that works; however, at this stage it would also be valuable to review in detail the approaches to testing for treeness currently in the literature. This could be useful as the bias against publishing negative results may have resulted in some of this research in this area not having been published.

The partial LASSO algorithm provided a novel extension of the LASSO approach to regression. This approach allows users to define a group of variables which form the initial model; and we applied the method to neighbor-net, and our user-defined set was the set of trivial splits.

In future, we could apply the procedure with the user-defined set as the splits from a tree. We could then use our method to see which split(s) are optimal to add to the tree. An F-test or comparison of AIC would allow us to investigate tree-likeness in a slightly different way. The method is also likely to have a wide range of applications in statistics more generally.

Our final network chapter was on consensus networks. We extended the splits-based approach of Holland and Moulton (2003) by applying the LASSO algorithm to choosing the set of statistically supported splits. This is a novel approach to building consensus networks, as previous methods chose the splits by their frequency in the input trees. This method has the potential to outperform the frequency-based approaches when several infrequent splits show heterogeneity that would otherwise be missed or when an infrequent but very different topology is within the set of input trees. The method is useful in its current state.

In our work on heterogeneity caused by different genes evolving on different topologies,

we developed a method for finding a confidence set of topologies for a set of genes. Our method was novel in that it took well-established methods for assessing the p-value of a single topology for a single gene and used them to get p-values for those topologies on multiple genes. As data sets with DNA from multiple genes are becoming readily available we believe this method is very timely. The method is useful in its present form.

In future, we will investigate whether other methods of evaluating p-values on a single gene and topology perform well in this framework. We will also carry out some further case study investigations.

The first part of our work on recombination developed a full alignment method to detect recombination event boundaries, which we termed "breakpoints". Our method was novel in that it applied tools from time series analysis to detecting recombination. By exploiting the fact that the ordering of the sites is important, we were able to see that time series tools could be applied to information collected based on the site ordering. Our approach was to construct two different sequences (the influence function and the distance based splits approach) that we hypothesized to contain information on recombination as a change in the mean. We then applied two structural break detection methods (Cappelli et al., 2008; Bai and Perron, 1998, 2003) to the series to locate changes in the mean.

The most promising outcome of this work is that the influence function (Bar-Hen et al., 2008) has potential as a hotspot detection method. The influence function and breakpoint detection method together locate recombination best when the proportion of the alignment that is recombined is small. It works best in sequences with a lot of informative sites.

In future, this work could be expanded, first by testing other series for their potential to detect recombination. Second, the idea that site orderings contain a great deal of information on phenomena like recombination may allow us to borrow other tools from time series analysis.

The second part of our work on recombination focussed on using incompatibility to learn more about recombination. Incompatibility had previously been used as a basis for a few tests looking for the presence or absence of recombination. We developed a breakpoint detection method. Our work was novel in taking a first principles approach to modelling incompatibility based on how the incompatibility pattern changed in the presence of

recombination. Our method often accurately returned the correct breakpoint as the first or second potential breakpoint for short alignments.

In future, we will improve the computational efficiency of this method and will also investigate whether it is possible to systematically determine how many recombination breakpoints there are. Our current method returns a ranked list, but we have no way of determining how many breakpoints are significant.

All these approaches to heterogeneity were from a statistical perspective. We have made several original contributions to the field of phylogenetics by approaching these problems with a statistical mindset.

Bibliography

Abby, S. S., E. Tannier, M. Gouy, and V. Daubin (2010). Detecting lateral gene transfers by statistical reconciliation of phylogenetic forests. *BMC Bioinformatics 11*(324).

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

Andresson, J. (2005). Lateral gene transfer in eukaryotes. *Cell Mol Life Sci 62*(11), 1182–1197.

Archibald, J. M. and A. J. Roger (2002). Gene conversion and the evolution of euryarchaeal chaperonins: A maximum likelihood-based method for detecting conflicting phylogenetic signals. *J. Mol. Evol. 55*(2), 232–245.

Bai, J. and P. Perron (1998). Estimating and testing linear models with multiple structural changes. *Econometrica 66*(1), 47–78.

Bai, J. and P. Perron (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics 18*, 1–22.

Baldauf, S. L. (1999). A search for the origins of animals and fungi: Comparing and combining molecular data. *The American Naturalist 154*(178-188).

Bandelt, H.-J. and A. W. Dress (1992). Split decomposition: A new and useful approach to phylogenetic analysis of distance data. *Mol. Phylogenet. Evol. 1*(3), 242–252.

Bandelt, H.-J., P. Forster, B. C. Sykes, and M. B. Richards (1995). Mitochondrial portraits of human popluations using median networks. *Genetics 141*, 743–753.

Bapteste, E., E. Susko, J. Leigh, I. Ruiz-Trillo, J. Bucknam, and W. Doolittle (2008).

Alternative methods for concatenation of core genes indicate a lack of resolution in deep nodes of the prokaryotic phylogenyprokaryotic phylogenyack of resolution in deep nodes of the prokaryotic phylogeny. *Mol. Biol. Evol. 25*(1), 83–91.

Bar-Hen, A., M. Mariadassou, M.-A. Poursat, and P. Vandenkoornhuyse (2008). Influence function for robust phylogenetic reconstructions. *Mol. Biol. Evol. 25*(5), 869–873.

Baroni, M., C. Semple, and M. Steel (2004). A framework for representing reticulate evolution. *Annals of Combinatorics 8*, 381–408.

Becq, J., C. Churlaud, and P. Deschavanne (2010). A benchmark of parametric methods of horizontal transfers detection. *PLoS One 5*(4).

Birin, H., Z. Gal-Or, I. Elias, and T. Tuller (2008). Inferring horizontal transfers in the presence of rearrangements by the minimum evolution criterion. *Bioinformatics 24*(6), 826–832.

Boc, A., H. Philippe, and V. Makarenkov (2010). Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Syst. Biol. 59*(2), 195–211.

Boni, M. F., D. Posada, and M. W. Feldman (2007). An exact nonparametric method for inferring mosiac structure in sequence triplets. *Genetics 176*, 1035–1047.

Bremer, K. (1990). Combinable component consensus. *Cladistics 6*(4), 369–372.

Brown, C. J., E. C. Garner, A. K. Dunker, and P. Joyce (2001). The power to detect recombination using the coalescent. *Mol. Biol. Evol. 18*(7), 1421–1424.

Bruen, T. C., H. Philippe, and D. Bryant (2006). A simple and robust statistical test for detecting the presense of recombination. *Genetics 172*(2665-2681).

Bruen, T. C. and M. Poss (2007). Recombination in feline immunodeficiency virus genomes from naturally. *Virology 364*, 362–370.

Bryant, D. (2005). Extending tree models to split networks. In *Algebraic statistics for computationaly biology*. Cambridge University Press.

Bryant, D. and V. Moulton (2004). Neighbor-net: An agglomerative method for the construction of phylogenetic networks. *Mol. Biol. Evol. 21*(2), 255–265.

Bryant, D. and M. Steel (2009). Computing the distribution of a tree metric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 6*(3), 420–426.

Bryant, D. and P. Waddell (1997). Rapid evaluation of least squares and minimum evolution criteria on phylogenetic trees. *Mol. Biol. Evol. 15*(10), 1346–1359.

Bulmer, M. (1991). Use of the method of generalized least squares in reconstructing phylogenies from sequence data. *Mol. Biol. Evol. 8*(6), 868–883.

Burnham, K. and D. R. Anderson (1998). *Model selection and inference: a practical information-theoretic approach*. Springer-Verlag.

Burnham, K. and D. R. Anderson (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological methods and research 33*(2), 261–304.

Cai, J. J., D. K. Smith, X. Xia, and K. Y. Yuen (2005). MBEToolbox: a matlab toolbox for sequence data analysis in molecular biology and evolution. *BMC Bioinformatics 6*(64).

Camin, J. H. and R. R. Sokal (1965). A method for deducing branching sequences in phylogeny. *Evolution 19*(3), 311–326.

Cappelli, C., R. N. Penny, W. S. Rea, and M. Reale (2008). Detecting multiple mean breaks at unknown points in official time series. *Mathematics and Computers in Simulation 78*(2-3), 33511356.

Casola, C. and M. W. Hahn (2009). Gene conversion among paralogs results in moderate false detection of positive selection using likelihood methods. *J. Mol. Evol. 68*, 679–687.

Cavalli-Sforza, L. L. and A. W. F. Edwards (1967). Phylogenetic analysis: Models and estimation procedures. *Evolution 32*, 550–570.

Chan, C. X., R. G. Beiko, and M. A. Ragan (2006). Detecting recombination in evolving nucleotide sequences. *BMC Bioinformatics 7*(412).

Chen, Z.-Z. and L. Wang (2010). HybirdNET: a tool for constructing hybridization networks. Advance Access published, Bioinformatics.

Cohen, O. and T. Pupko (2010). Inference and characterization of horizontally transferred gene families using stochastic mapping. *Mol. Biol. Evol. 27*(3), 703–713.

Cotton, J. A. and M. Wilkinson (2008). Supertrees join the maintream of phylogenetics. *Trends in Ecology and Evolution 24*(1), 1–3.

Crawford, D. C., T. Bhangale, N. Li, G. Hellenthal, M. J. Reider, D. A. Nickerson, and M. Stephens (2004). Evidence for substantial fine-scale variation in recombination rates across the human genome. *Nature Genetics 36*(7), 700–706.

Doolittle, R. F., D. F. Feng, K. L. Anderson, and M. R. Alberro (1990). A naturally occuring horizontal gene transfer from a eukaryote to a prokaryote. *J. Mol. Evol. 31*, 383–388.

Drummond, A. and A. Rambaut (2007). BEAST: bayesian evoluionary analysis by sampling trees. *BMC Evolutionary Biology 7*, 214.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *Ann. Statist. 32*(2), 407–499. With discussion, and a rejoinder by the authors.

Etherington, G. J., J. Dicks, and I. N. Roberts (2005). Recombination Analysis Tool (RAT): a program for the high-throughput detection of recombination. *Bioinformatics 12*(3), 278–281.

Fang, F., J. Ding, V. N. Minn, M. A. Suchard, and K. S. Dorman (2007). cBrother: relaxing parental tree assumptions for bayesian recombination detection. *Bioinformatics 23*(4), 507–508.

Farris, J. S., M. Kallersjo, A. G. Kluge, and C. Bult (1995). Constructing a significance test for incongruence. *Syst. Biol. 44*(4), 570–572.

Fearnhead, P., R. M. Harding, J. A. Schneider, S. Myers, and P. Donnelly (2004). Application of coalescent methods to reveal fine-scale rate variation and recombination hotspots. *Genetics 167*(2067-2081).

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates.

Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American Statistical Association 53*(284), 789–798.

Fletcher, R. (2000). *Practical Methods of Optimization* (2nd ed.). Wiley.

Gascuel, O. (1997). *Concerning the NJ algorithm and its unweighted version, UNJ*. American Mathematical Society, Providence, R.I.

Gibbs, M. J., J. S. Armstrong, and A. J. Gibbs (2000). Sister-scanning: a Monte Carlo procedure for assessing signals in recombinant sequences. *Bioinformatics 16*(7), 573–582.

Goldman, N. (1993). Statistical tests of models of DNA substituion. *J. Mol. Evol. 36*(182-198).

Goldman, N., J. P. Anderson, and A. G. Rodrigo (2000). Likelihood-based tests of topologies in phylogenetics. *Syst. Biol. 49*(4), 652–670.

Graham, J., B. McNeney, and F. Seillier-Moiseiwitsch (2005). Stepwise detection of recombination breakpoints in sequence alignments. *Bioinformatics 21*(5), 589–595.

Grassly, N. C. and E. C. Holmes (1997). A likelihood method for the detection of selection and recombination using nucleotide sequences. *Mol. Biol. Evol. 14*(3), 239–247.

Grünewald, S., K. Forslund, A. Dress, and V. Moulton (2007). QNet: an agglomerative method for the construction of phylogenetics networks from weighted quartets. *Mol. Biol. Evol. 24*(2), 532–538.

Guindon, S. and O. Gascuel (2003). A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol. 52*(5), 696–704.

Hao, W. (2010). Orgconv: detection of gene conversion using consensus sequences and its application in plant mitochondrial and chloroplast homologs. *BMC Bioinformatics 11*(114).

Hein, J. (1993). A heuristic method to reconstruct the history of sequences subject to recombination. *J. Mol. Evol. 36*, 396–405.

Hendy, M. and D. Penny (1993). Spectral analysis of phylogenetic data. *Journal of Classification 10*, 5–24.

Hey, J. (2004). What's so hot about recombination hotspots? *PLoS Biology 2*(6), 730–733.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67.

Holland, B. and V. Moulton (2003). Consensus networks: A method for visualising

incompatibilities in collections of trees. In G. Benson and R. Page (Eds.), *Algorithms in Bioinformatics*, Volume 2812, pp. 165–176. Springer Berlin.

Holland, B. R., S. Benthin, P. Lockhart, V. Moulton, and K. T. Huber (2008). Using supernetworks to distinguish hybirdization from lineage-sorting. *BMC Evolutionary Biology 8*.

Holland, B. R., K. T. Huber, A. Dress, and V. Moulton (2002). δ plots: A tool for analzying phylogenetic distance data. *Mol. Biol. Evol. 19*(12), 2051–2059.

Hudson, R. R. (1983). Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology 23*, 183–201.

Huelsenbeck, J. P. and F. Ronquist (2001). MRBAYES: Bayesian inference of phylogeny. *Bioinformatics 17*, 754–755.

Husmeier, D. and F. Wright (2001a). Detection of recombination in DNA multiple alignments with hidden markov models. *Journal of Computational Biology 8*(4), 401–427.

Husmeier, D. and F. Wright (2001b). Probabilistic divergence measures for detecting interspecies recombination. *Bioinformatics 1*(1-8).

Husmeier, D., F. Wright, and I. Milne (2005). Detecting interspecific recombination with a pruned probablistic divergence measure. *Bioinformatics 21*(9), 1797–1896.

Huson, D. and D. Bryant (2006). Application of phylogenetic networks in evolutionary studies. *Mol. Biol. Evol. 23*(2), 254–267.

Huson, D. H., R. Rupp, and C. Scornavacca (2010). *Phylogenetic networks*. Cambridge.

Ingman, M., H. Kaessmann, S. Pääbo, and U. Gyllensten (2000). Mitochondrial genome variation and the origin of modern humans. *Nature 408*, 708–713.

Jakobsen, I. B. and S. Easteal (1996). A program for calculating and displaying compatibility matrices as an aid in determing reticular evolution in molecular sequences. *CABIOS 12*(4), 291–295.

Joly, S., P. A. McLenachan, and P. Lockhart (2009). A statistical approach for distinguishing hybridization and incomplete lineage sorting. *The American Naturalist 174*(2), 54–70.

Jukes, T. and C. Cantor (1969). *Evolution of protein molecules*. New York: Academic Press.

Keeling, P. and J. Palmer (2008). Horizontal gene transfer in eukaryotic evolution. *Nat Rev Genet 9*(8), 605–618.

Kimura, M. and T. Ohta (1972). On the stochastic model for estimation of mutational distance between homologous proteins. *J. Mol. Evol. 2*, 87–90.

Kishino, H. and M. Hasegawa (1989). Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea. *J. Mol. Evol. 29*, 170–179.

Kishino, H., T. Miyata, and M. Hasegawa (1990). Maximum likelihood inference of proteion phylogeney and the origins of chloroplats. *J. Mol. Evol. 30*(2), 151–160.

Kuhn, H. W. and A. W. Tucker (1951). Nonlinear programming. In J. Neyman (Ed.), *Processding of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492. University of California Press, Berkeley, California.

Le Quesne, W. J. (1969). A method of selection of characters in numerical taxaonomy. *Systematic Zoology 18*(2), 201–205.

Leigh, J. W., E. Susko, M. Baumgartner, and A. J. Roger (2008). Testing congruence in phylogenomic analysis. *Syst. Biol. 57*(1), 104–115.

Lemey, P., M. Lott, D. P. Martin, and V. Moulton (2009). Identifying recombinants in human and primate immunodeficiency virus sequence alignments using quartet scanning. *BMC Bioinformatics 10*(126).

Lento, G. M., R. E. Hickson, G. K. Chambers, and D. Penny (1995). Use of spectral analysis to test hypotheses on the origin of pinnipeds. *Mol. Biol. Evol. 12*(1), 28–52.

Li, N. and M. Stephens (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics 165*, 2213–2233.

Loughin, T. M. (2004). A systematics comparision of methods for combining p-values from independent tests. *Computational Statistics and Data Analysis 47*(3), 467–485.

Lyons-Weiler, J., G. A. Hoelzer, and R. J. Tausch (1996). Relative Apparent Synapomor-

phy Analysis (RASA) I: The statistical measurement of phylogenetic signal. *Mol. Biol. Evol. 13*(6), 749–757.

Maddison, W. P. (1997). Gene trees in species trees. *Syst. Biol. 46*(3), 523–536.

Makarenkov, V. and P. Legendre (2004). From a phylogenetic tree to a reticulated network. *Journal of Computational Biology 11*(1), 195–212.

Margush, T. and F. R. McMorris (1981). Consensus n-Trees. *Bulletin of Mathematical Biology 43*(2), 239–244.

Martin, D. and E. Rybicki (2000). RDP: detection of recombination amongst aligned sequences. *Bioinformatics Applications Note 16*(6), 562–563.

Martins, L. d. O., È. Leal, and H. Kishino (2008). Phylogenetic detection of recombination with a bayesian prior on the distance between trees. *PLoS One 3*(7).

Maydt, J. and T. Lengauer (2006). Recco: recombination analysis using cost optimization. *Bioinformatics 22*(9), 1064–1071.

McGuire, G., F. Wright, and M. J. Prentice (1997). A graphical method for detecting recombination in phylogenetic data sets a graphical method for detecting recombination in phylogenetic data sets. *Mol. Biol. Evol. 14*(11), 1125–1131.

McGuire, G., F. Wright, and M. J. Prentice (2000). A bayesian model for detecting past recombination events in dna multiple alignments. *Journal of Computational Biology 7*(1/2), 159–170.

McVean, G. A. T., S. R. Myers, S. Hunt, P. Deloukas, D. R. Bentely, and P. Donnelly (2004). The fine-scale structure of recombination rate variation in the human genome. *Science 304*, 581–584.

Minn, V. N., K. S. Dorman, F. Fang, and M. A. Suchard (2005). Dual multiple change-point model leads to more accurate recombination detection. *Bioinformatics 21*(13), 3034–3042.

Myers, S. R. and R. C. Griffiths (2003). Bounds on the minimum number of recombination events in a sample history. *Genetics 163*, 375–394.

Nakleh, L., T. Warnow, C. R. Linder, and K. St. John (2005). Reconstructing reticulate

evolution in species - theory and practice. *Journal of Computational Biology 12*(6), 796–811.

Nei, M. and L. Jin (1989). Variances of the average numbers of nucleotide substitutions within and between populations. *Mol. Biol. Evol. 6*(3), 290–300.

Osborne, M. R., B. Presnell, and B. A. Turlach (2000a). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis 20*, 389–404.

Osborne, M. R., B. Presnell, and B. A. Turlach (2000b). On the lasso and its dual. *Journal of Computational and Graphical Statistics 9*(2), 319–337.

Padhukasahasram, B., J. D. Wall, P. Marjoram, and M. Nordborg (2006). Estimating recombination rates from single-nucleotide polymorphisms using summary statistics. *Genetics 174*, 1517–1528.

Park, M. Y. and T. Hastie (2007). *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*. R package version 0.94.

Penny, D. and M. Hendy (1986). Estimating the reliability of evolutionary trees. *Mol. Biol. Evol. 3*(5), 403–417.

Penny, D., M. Steel, P. Waddell, and M. Hendy (1995). Improved analysis of human mtDNA sequences support a recent African origin for Homo sapiens. *Mol. Biol. Evol. 12*(5), 863–882.

Pond, S. L. K., D. Posada, M. B. Gravenor, C. H. Woelk, and S. D. Frost (2006). GARD: a genetic algorithm for recombination detection. *Bioinformatics Applications Note 22*(24), 3096–3098.

Posada, D. (2002). Evaluation of methods for detecting recombination from DNA sequences: Emperical data. *Mol. Biol. Evol. 19*(5), 708–717.

Posada, D. and K. A. Crandall (2001). Evaluation of methods for detecting recombination from DNA sequences: Computer simulations. *PNAS 98*(24), 13757–13762.

Puigbò, P., Y. I. Wolf, and E. V. Koonin (2010). The tree and nets components of prokaryote evolution. *Genome Biol Evol 2*, 745–756.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Ragan, M. A. (1992). Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics And Evolution 1*(1), 53–58.

Rambaut, A. and N. C. Grassly (1997). Seq-Gen: an application for the Monte Carlo simulation of the DNA sequence of DNA sequence evolution along phylogenetic trees. *CABIOS 13*(3), 235–238.

Rao, C. R. (1970). Estimation of heteroscedastic variances in linear models. *Journal of the American Statistical Association 65*(329), 161–172.

Ratcliffe, J. M. and M. L. Nydam (2008). Multimodal warnings signals for a multiple predator world. *Nature 455*(4), 96–99.

Rea, W. (2008). *The Application of Atheoretical Regression Rrees to Problems in Time Series Analysis*. Ph. D. thesis, University of Canterbury, New Zealand.

Robinson, D. R. and L. R. Foulds (1981). Comparison of phylogenetic trees. *Mathematical Biosciences 53*(1-2), 131–147.

Rohlf, J. F. (1982). Consensus indices for comparing classifications. *Mathematical Biosciences 59*, 131–144.

Rzhetsky, A. and M. Nei (1992). A simple method for estimating and testing minimum-evolution trees. *Mol. Biol. Evol. 9*(5), 945–967.

Saitou, N. and M. Nei (1987). The neighbor-joining method: a new method for reconstruction of phylogenetic trees. *Mol. Biol. Evol. 4*(4), 406–425.

Salemi, M., R. R. Gray, and M. M. Goodenow (2008). An exploratory algorithm to identify intra-host recombinant viral sequences. *Mol. Phylogenet. Evol. 49*(2), 618–628.

Sanderson, M., M. M. McMahon, and M. Steel (2010). Phylogenomics with incomplete taxon coverge: the limits to inference. *BMC Evolutionary Biology 10*(155).

Schäfer, J. and K. Strimmer (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statstical Applications in Genetics and Molecular Biology 4*(1).

Schierup, M. H. and J. Hein (2000). Consequences of recombination on traditional phylogenetic analysis. *Genetics 156*, 879–891.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Semple, C. and M. Steel (2003). *Phylogenetics*. Oxford University Press.

Shi, X., H. Gu, E. Susko, and C. Field (2005). The comparision of the confidence regions in phylogeny. *Mol. Biol. Evol. 22*(11), 2285–2296.

Shimodaira, H. (2002). An approximately unbiased test of phylogenetic tree selection. *Syst. Biol. 51*(3), 492–508.

Shimodaira, H. and M. Hasegawa (1999). Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Mol. Biol. Evol. 16*(8), 1114–1116.

Smith, J. M. (1992). Analyzing the mosaic structure of genes. *J. Mol. Evol. 34*, 126–129.

Soltis, P. S. and D. E. Soltis (2009). The role of hybridization in plant speciation. *Annual Review of Plant Biology 60*, 561–588.

Song, Y. S., Z. Ding, D. Gusfield, C. H. Langley, and Y. Wu (2007). Algorithms to distingusih the role of gene-conversion from single-crossover recombination in the derivation of snp sequences in population. *J. Comput. Biol. 14*(10), 1273–1286.

Stamatakis, A. (2006). RAxML: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics Applications Note 22*(21), 2688–2690.

Stouffer, S., E. Suchman, L. DeVinnery, S. Star, and W. R (1949). *The American Soldier, Volume I: Adjustment during Army Life*. Princeton University Press.

Strimmer, K. and A. von Haeseler (1997). Likelihood-mapping: A simple method to visualize phylogenetic content of a sequence alignment. *PNAS 94*(13), 6815–6819.

Stuart, A. and J. K. Ord (1987). *Kendall's advanced theory of statistics* (5th ed.), Volume 1. Griffen.

Stumpf, M. P. H. and G. A. T. McVean (2003). Estimating recombination rates from population-genetic data. *Nature Reviews, Genetics 4*(959-968).

Suchard, M. A., R. E. Weiss, K. S. Dorman, and J. Sinsheimer (2003). Inferring spa-

tial phylogenetic variation along nucleotide sequences: A mulitiple changepoint model. *Journal of the American Statistical Association 98*(462), 427–437.

Susko, E. (2003). Confidence regions and hypothesis tests for topologies using gneralized least squares. *Mol. Biol. Evol. 20*(6), 862–868.

Susko, E. (2006). Using minimum bootstrap support for splits to construct confidence regions for trees. *Evolutionary Bioinformatics Online 2*, 129–143.

Swofford, D. L. (2000). *PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4.* Sunderland, Massachusetts.: Sinauer Associates.

Than, C., D. Ruths, H. Innan, and L. Nakhleh (2007). Confounding factors in HGT detection: Statistical error, coalescent effects and multiple solutions. *Journal of Computational Biology 14*(4), 517–535.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Vach, W. (1989). *Least squares approximation of additive trees*, pp. 230–238. Springer-Verlag.

Vigilant, L., M. Stoneking, H. Harpending, K. Hawkes, and A. C. Wilson (1991). African populations and the evolution of human mitochondrial DNA. *Science 253*(5027), 1503–1508.

Wall, J. D. and J. K. Pritchard (2003). Assessing the performance of the haplotype block model of linkage disequilibrium. *Ann. J. Hum. Genet. 73*, 502–515.

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

Whitlock, M. C. (2005). Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. *J. Evol. Biol 18*(5), 1368–1373.

Wilson, E. B. (1927). Probable inference, the law of succession and statistical inference. *Journal of the American Statistical Association 22*(158), 209–212.

Wiuf, C., T. Christensen, and J. Hein (2001). A simulation study of the reliability of recombination detection methods. *Mol. Biol. Evol. 18*(10), 1929–1939.

Wiuf, C. and J. Hein (2000). The coalescent with gene conversion. *Genetics 155*, 451–462.

Yule, G. (1925). A mathematical theorey of evoluation, based on the conclusions of Dr. J. C. Willis. *Phyilos. Trans. Roy. Soc. London Ser. B 213*(402-410), 21–87.

Zeileis, A., F. Leisch, K. Hornik, and C. Kleiber (2002). strucchange: An R package for testing for structural change in linear regression models. *Journal of Statistical Software 7*(2), 1–38.

A

Figures based on NNLS-LASSO and $\hat{\sigma}_T^2$

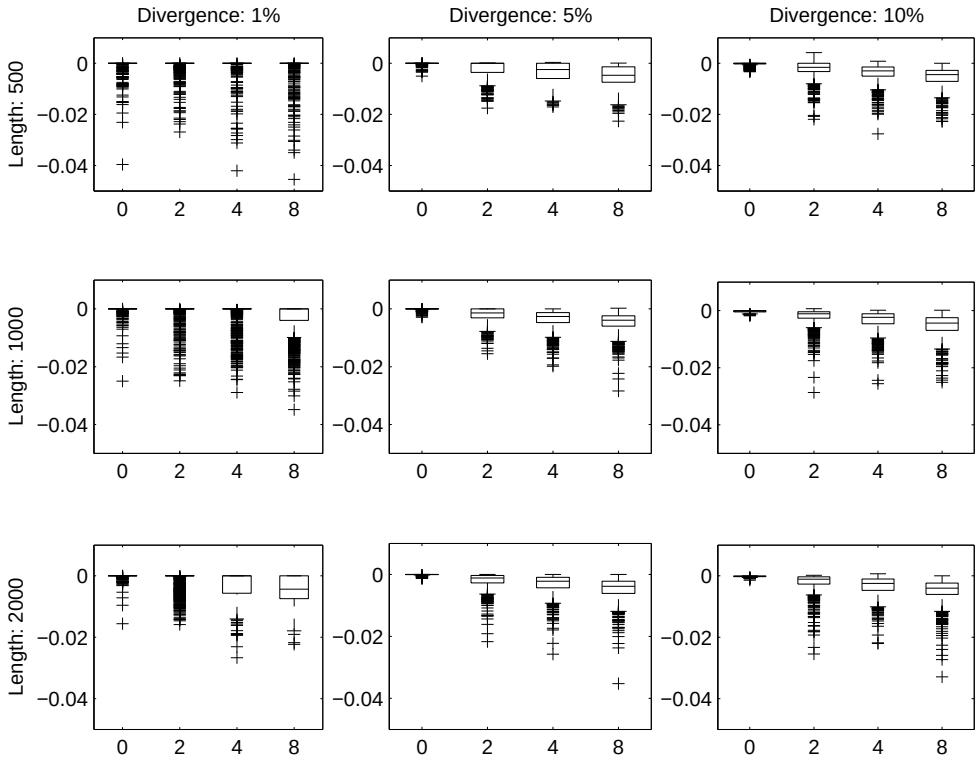

Figure A.1: Fit of the distances, bias. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

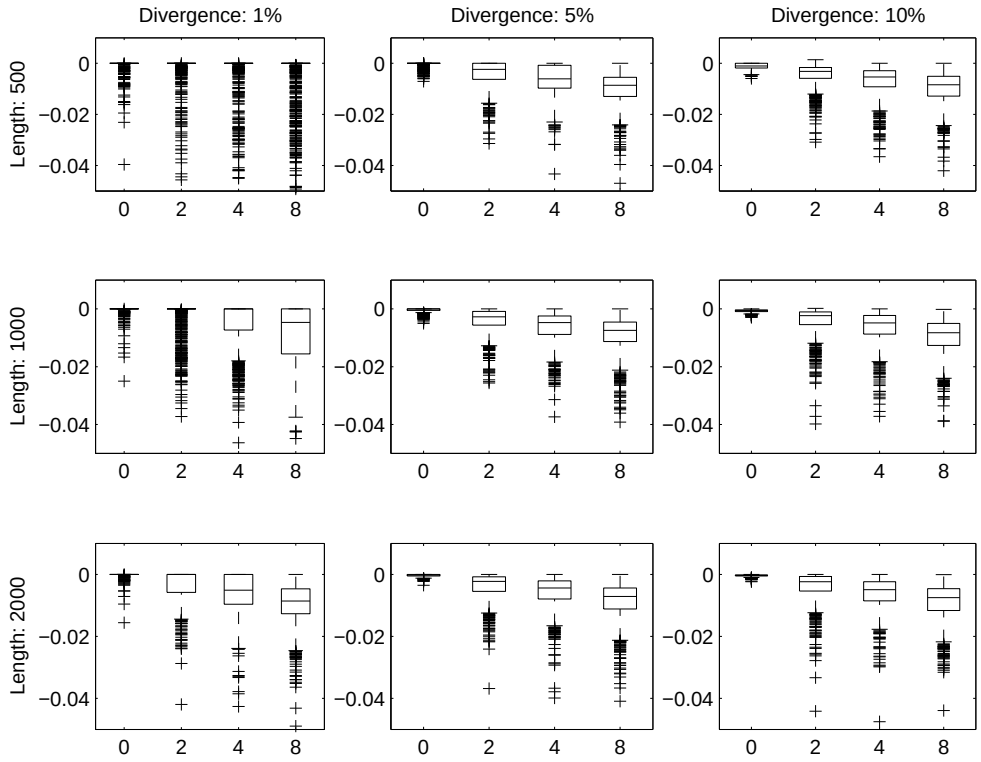

Figure A.2: Fit of the distances, bias. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

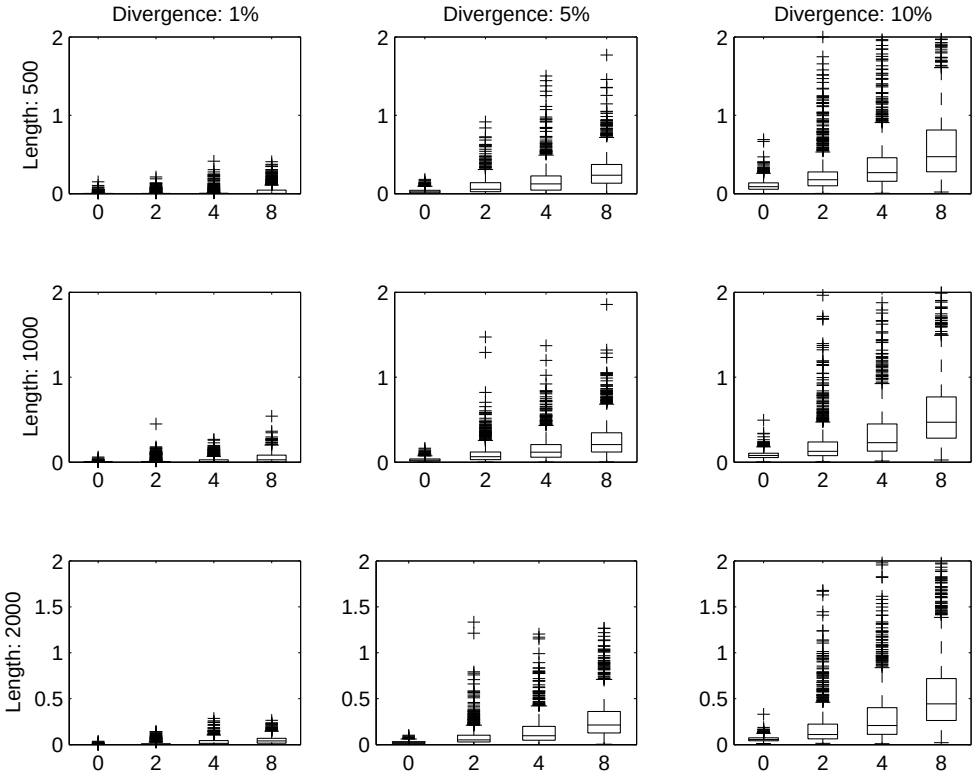

Figure A.3: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

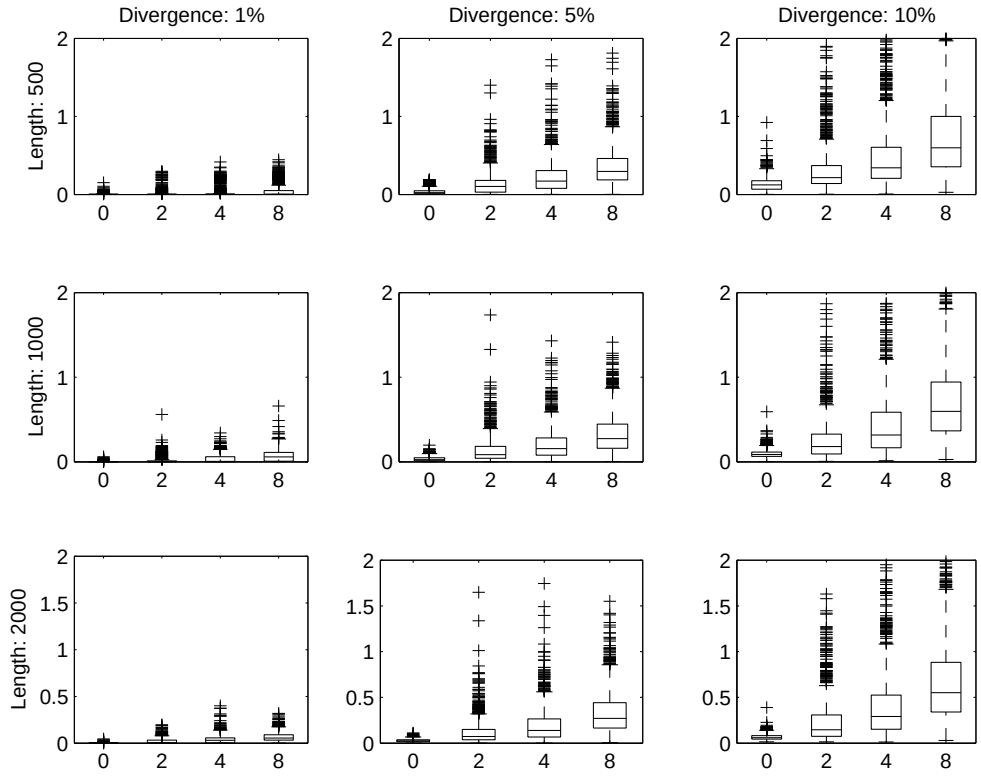

Figure A.4: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

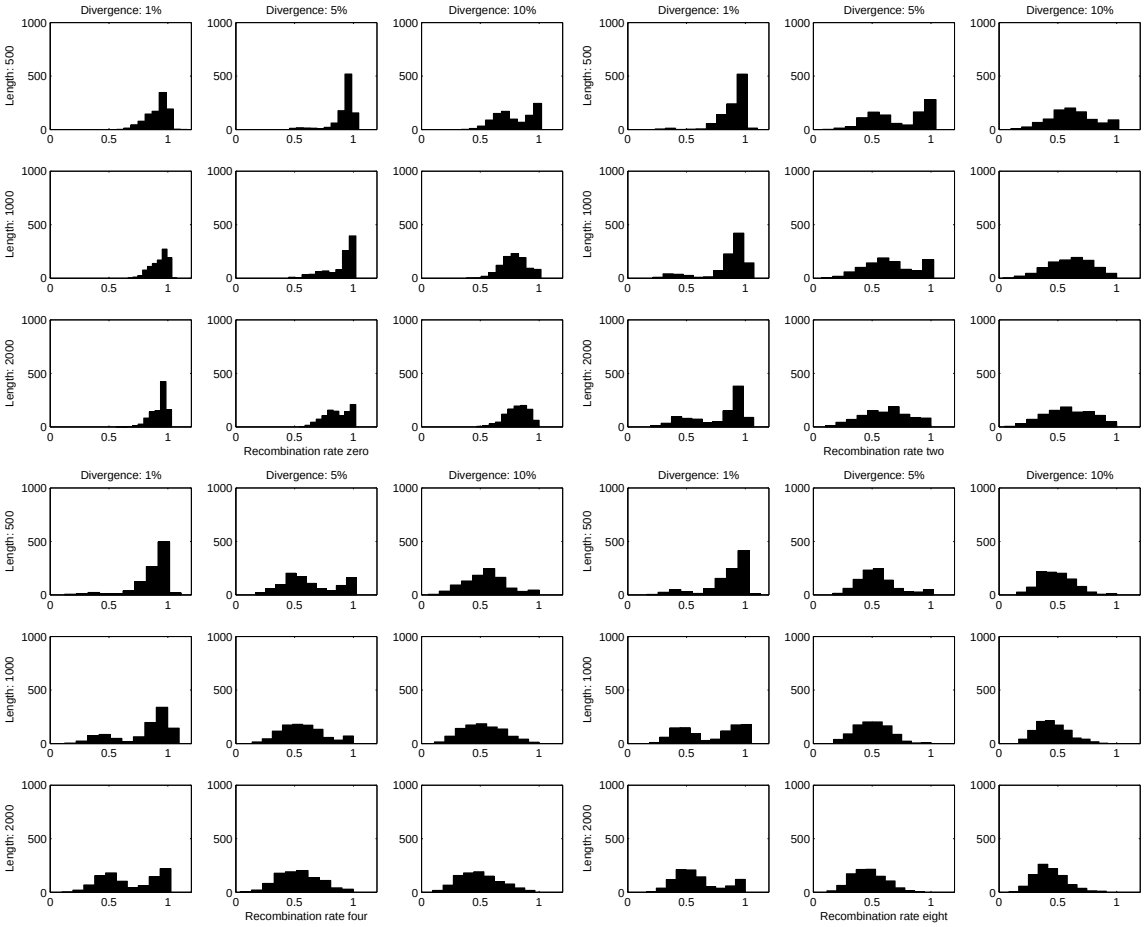

Figure A.5: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 1000 replications.

B

Figures based on NNLS-LASSO and $\hat{\sigma}_N^2$

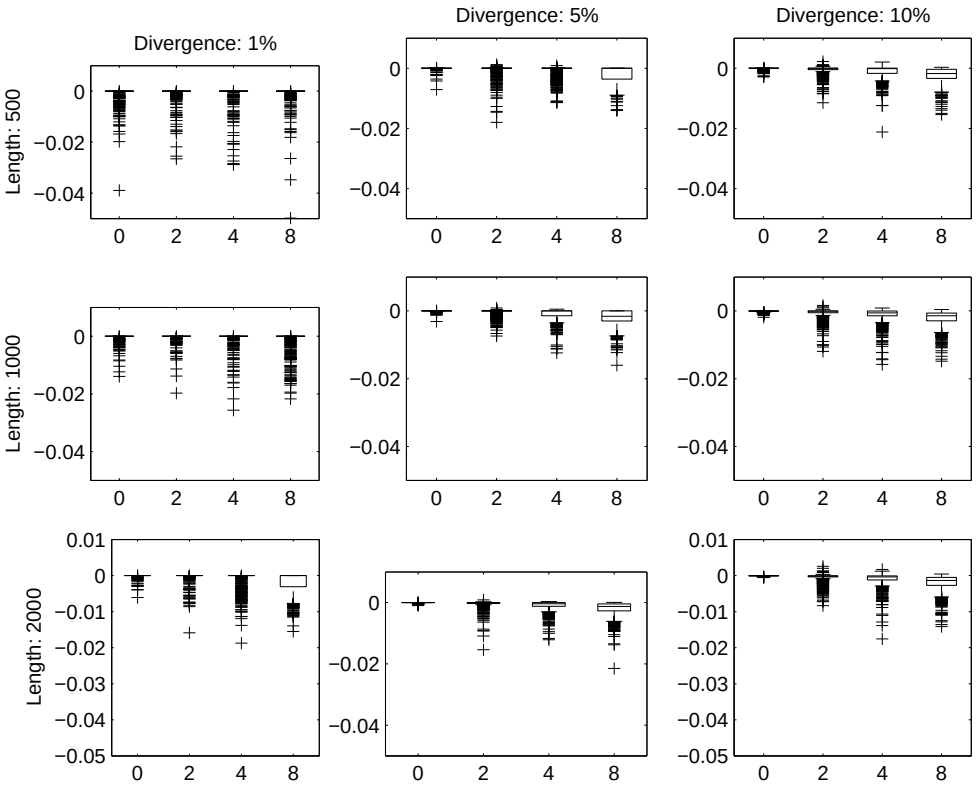

Figure B.1: Fit of the distances, bias. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

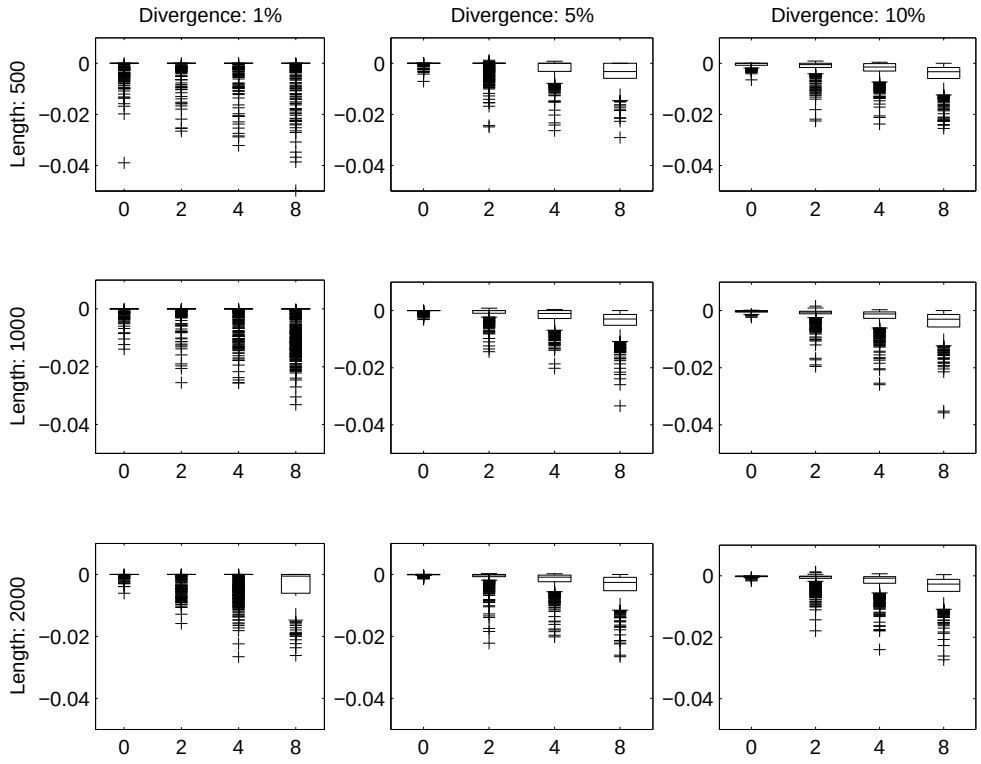

Figure B.2: Fit of the distances, bias. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

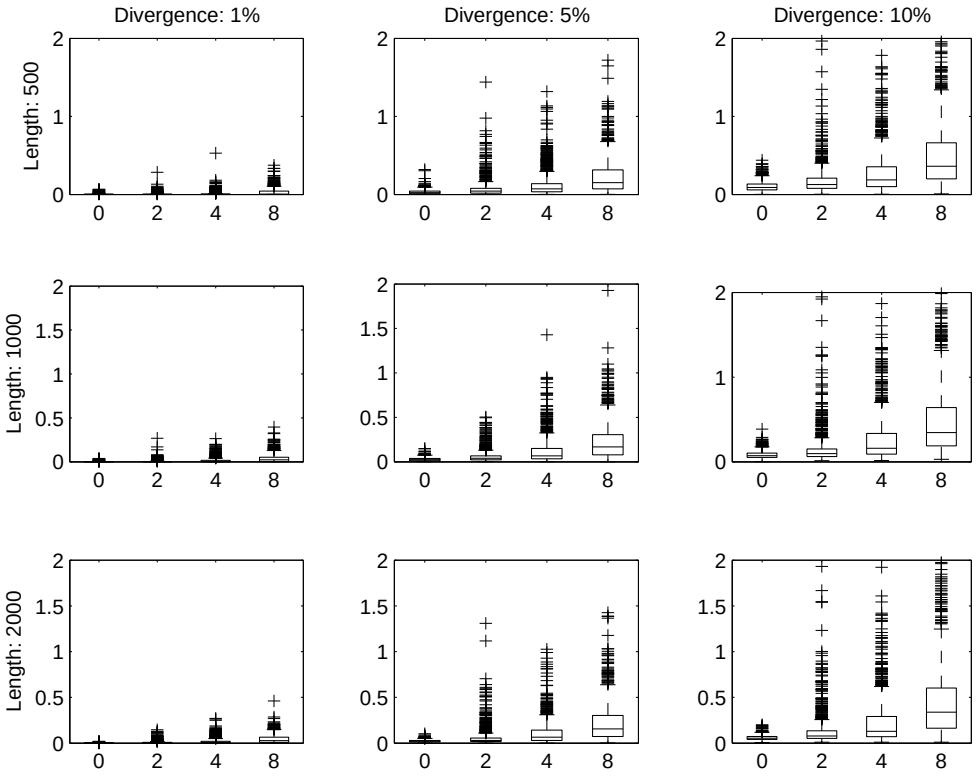

Figure B.3: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

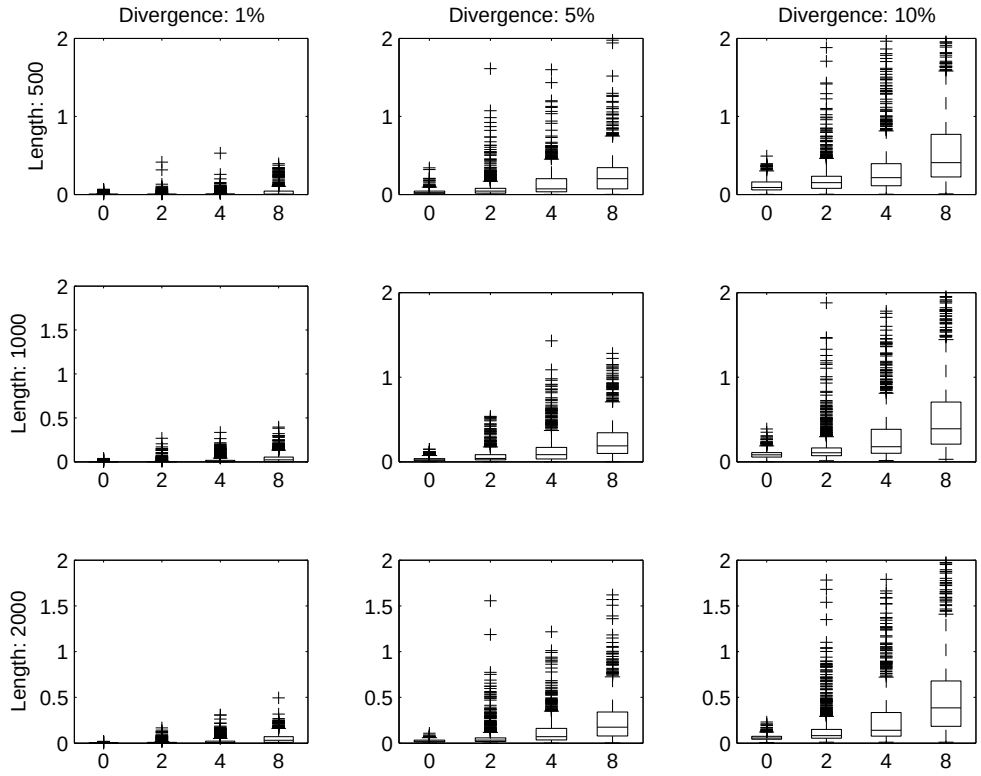

Figure B.4: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

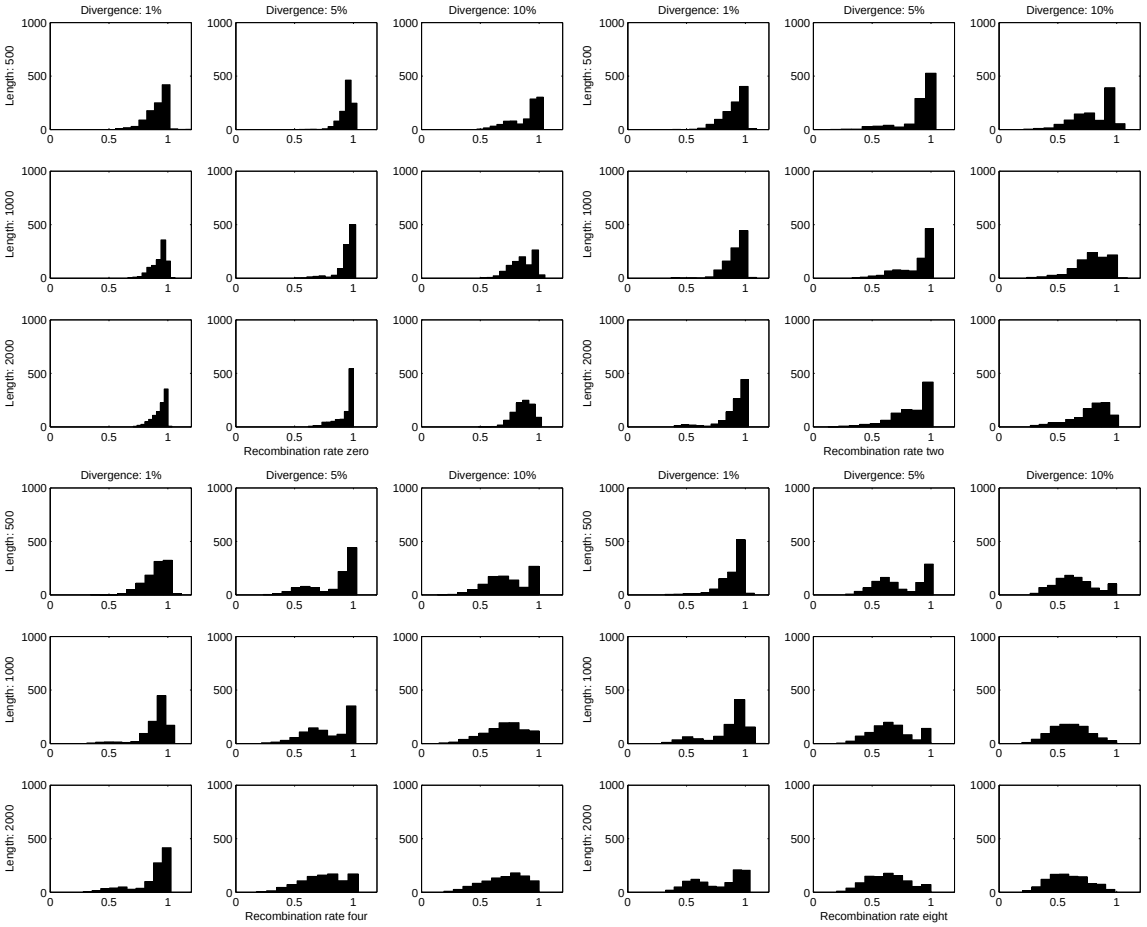

Figure B.5: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 1000 replications.

C

A comparison of the number of splits

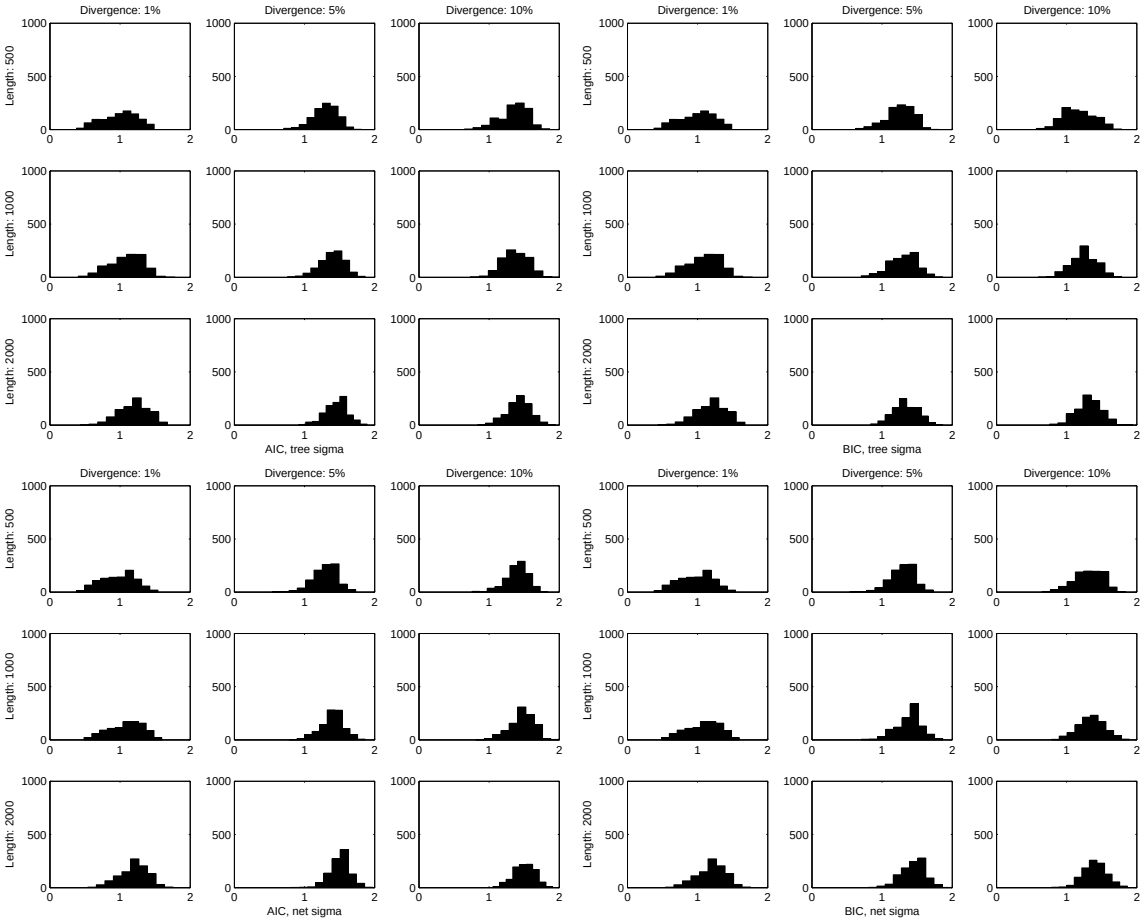

Figure C.1: Proportion of splits chosen relative to the number of splits in the tree. Number of taxa is 20, 1000 replications, no recombination.

D

Figures based on NNLS-LASSO and $\hat{\sigma}_T^2$, Covariance transformed

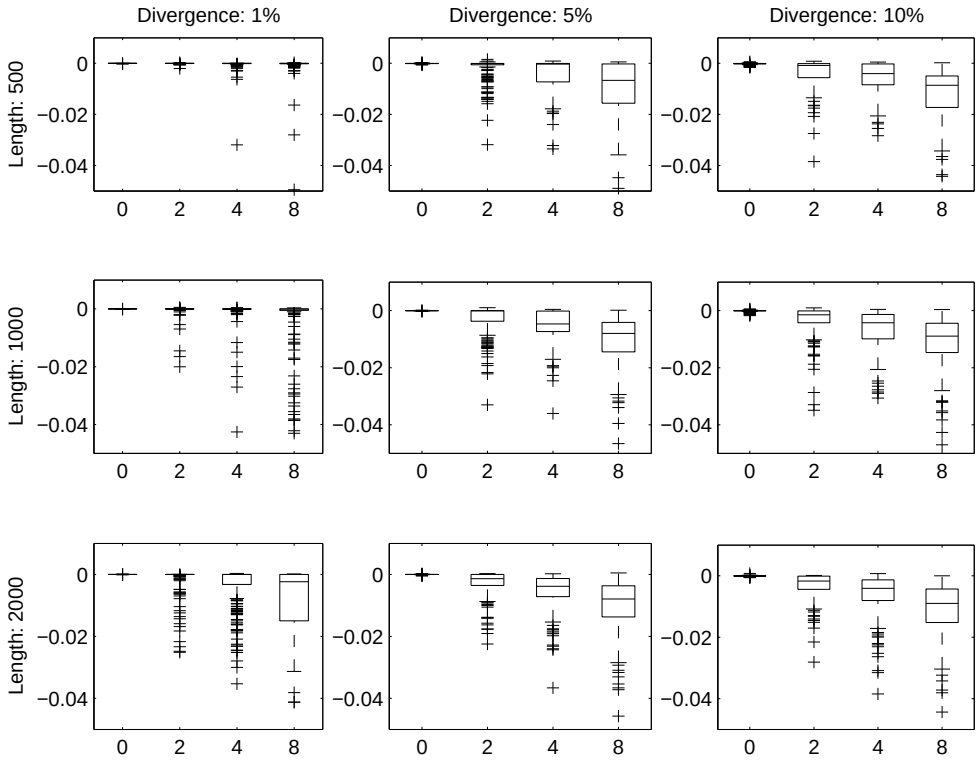

Figure D.1: Fit of the distances. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 200 replications.

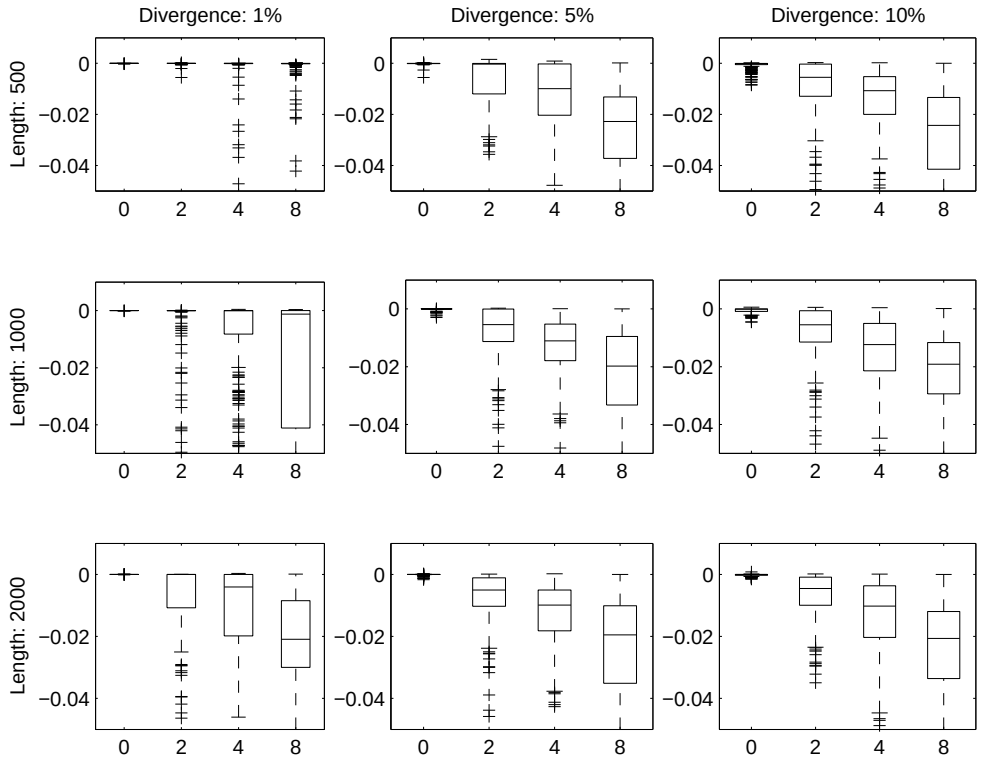

Figure D.2: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 200 replications.

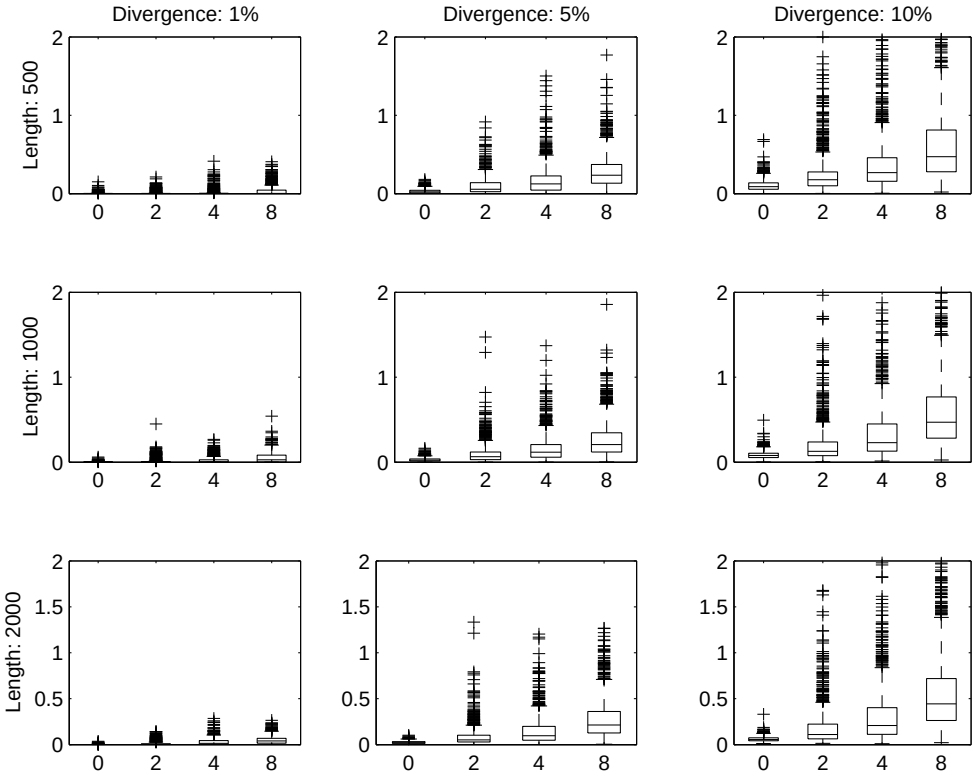

Figure D.3: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

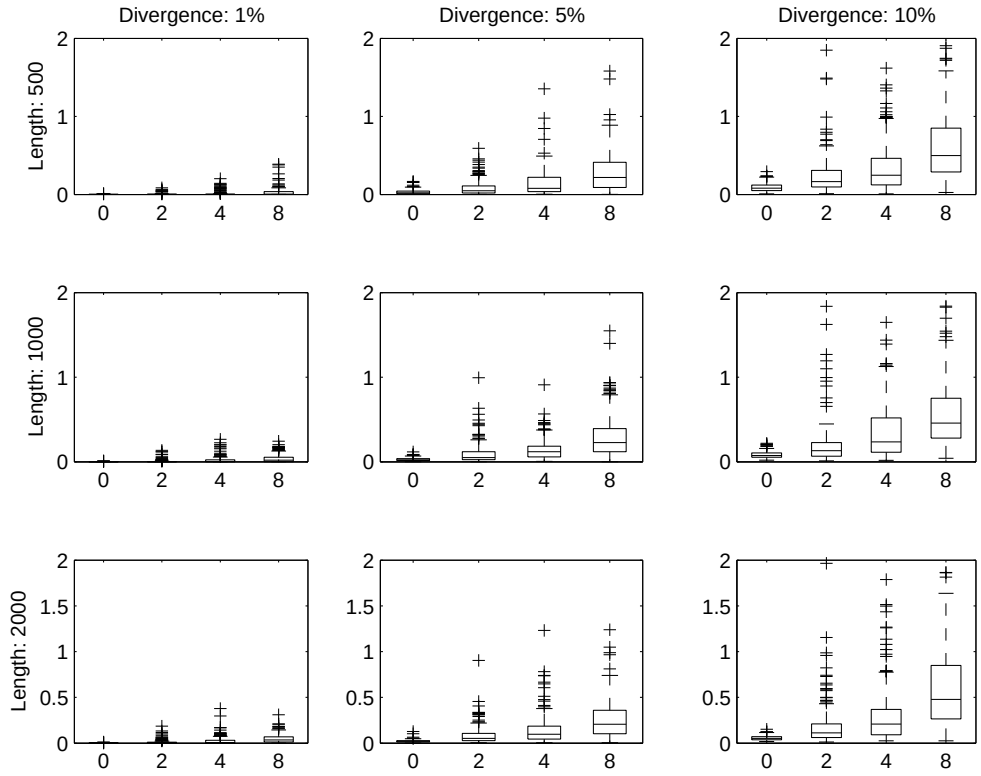

Figure D.4: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

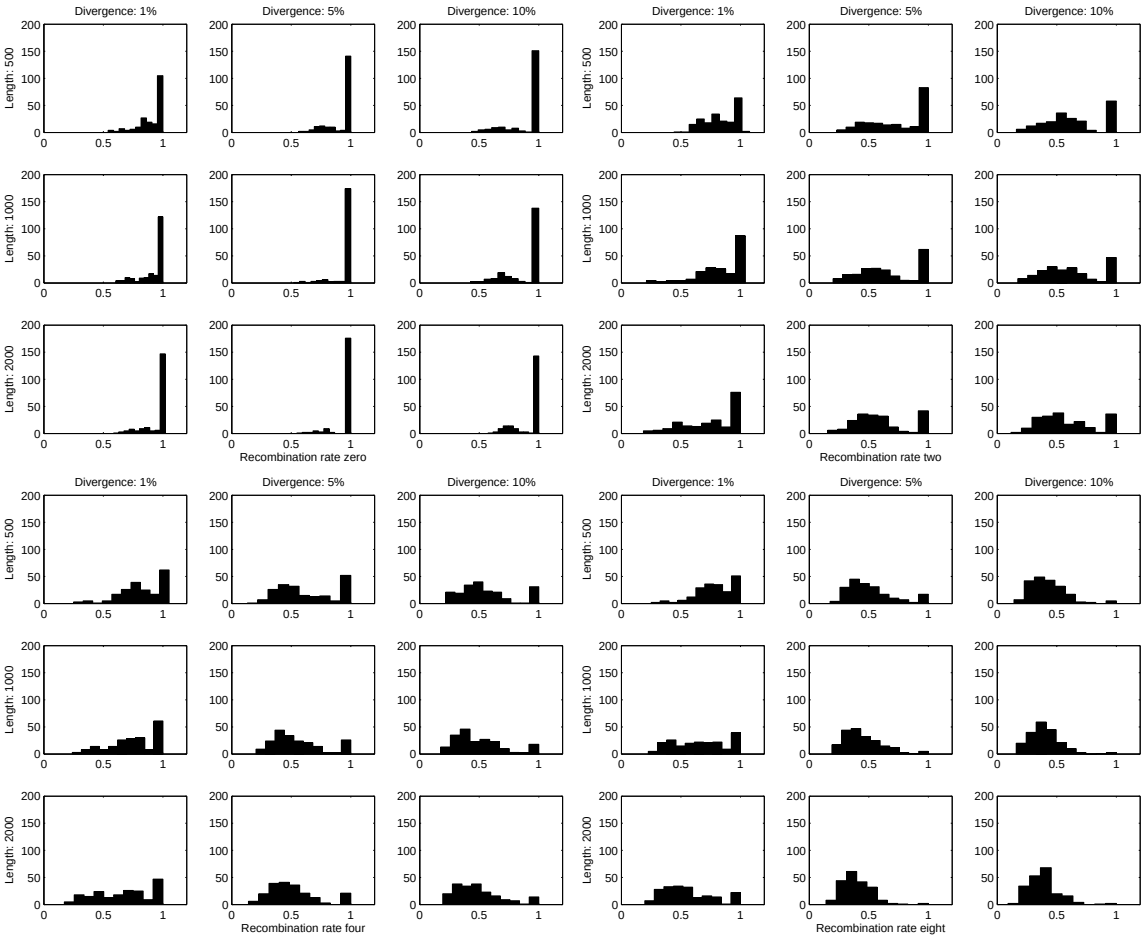

Figure D.5: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 200 replications.

E

Figures based on NNLS-LASSO and $\hat{\sigma}_N^2$, Covariance transformed

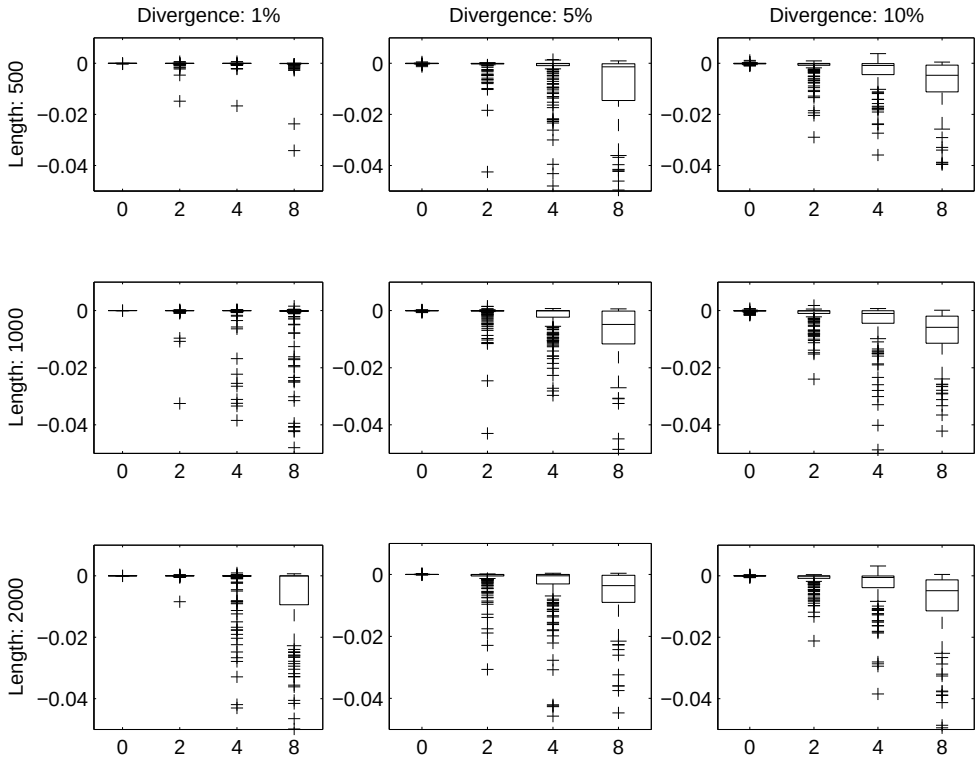

Figure E.1: Fit of the distances. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 200 replications.

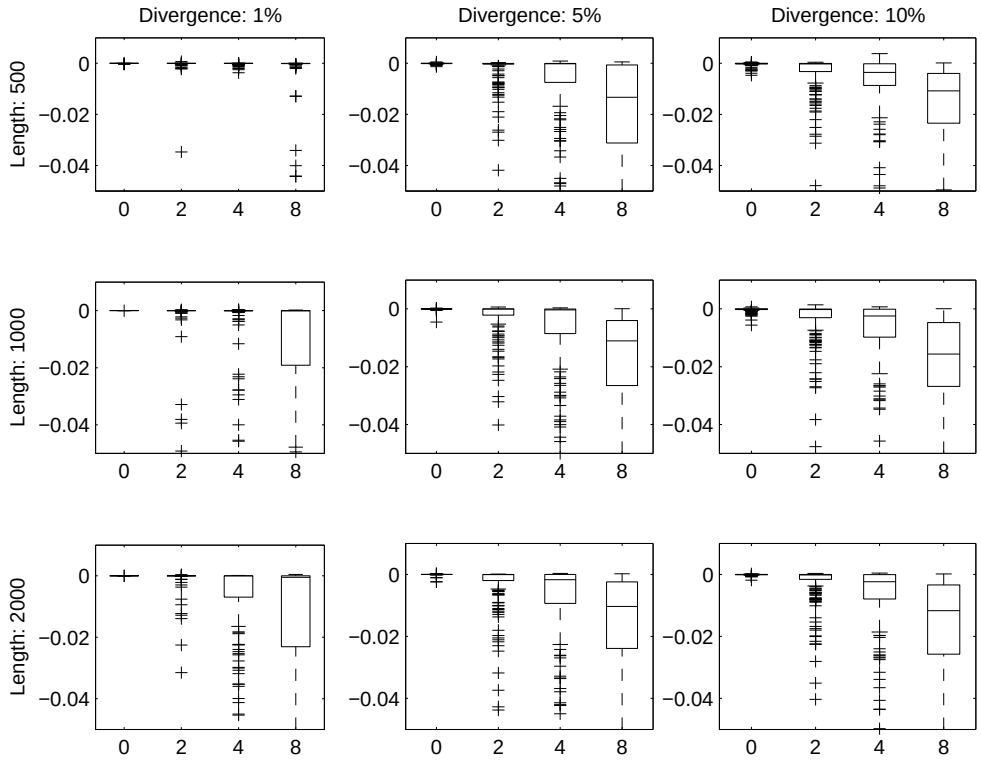

Figure E.2: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 200 replications.

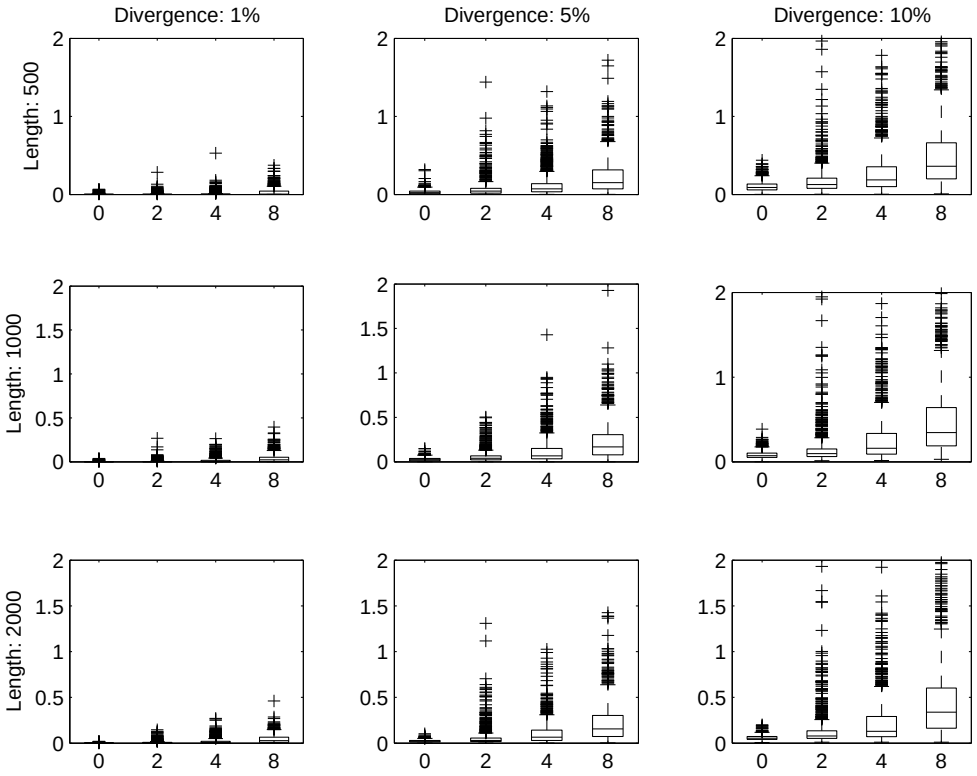

Figure E.3: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

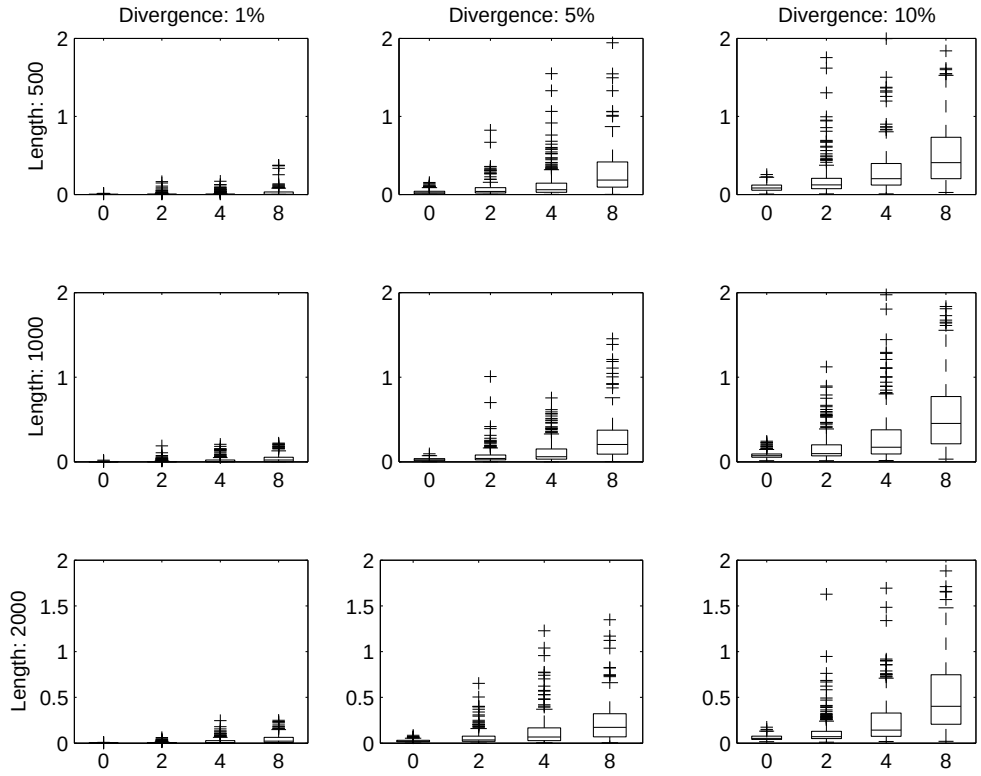

Figure E.4: Fit of the distances, absolute difference. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

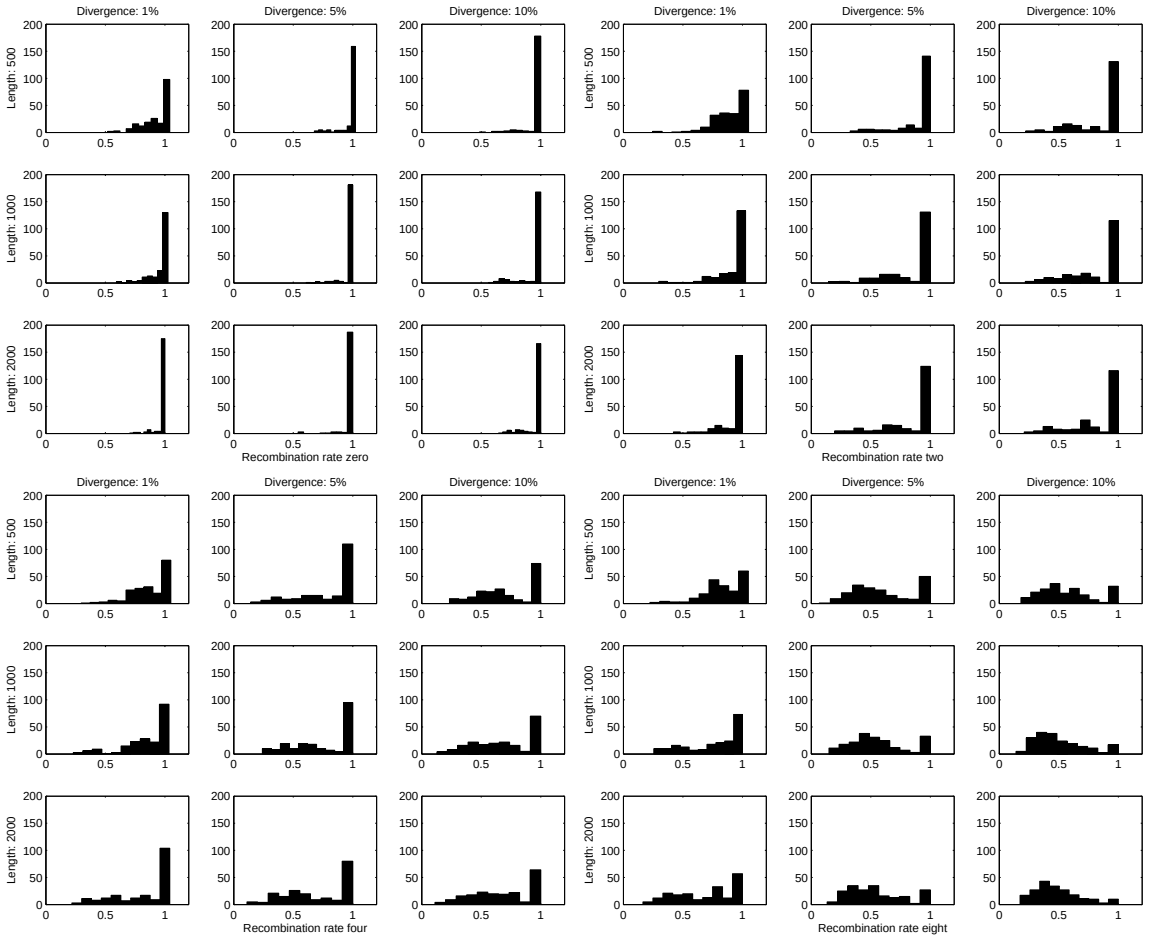

Figure E.5: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 200 replications.

F

Figures based on Hadamard likelihood

This section contains further results for the Hadamard likeilhood.

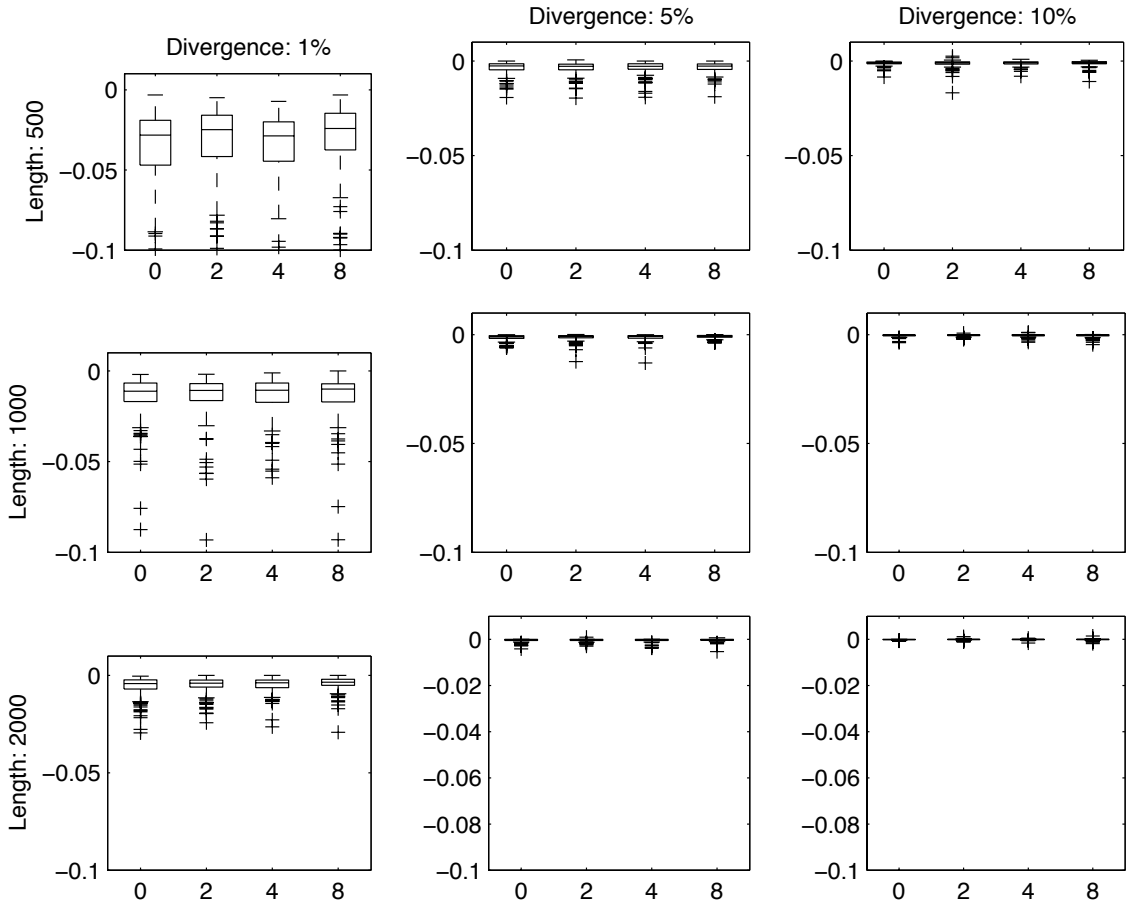

Figure F.1: AIC criterion. Box plots of the fit of the distances for sequence lengths 500, 1000 and 2000 and divergences rate 1%, 5% and 10%. Each figure contains fits for the recombinations rates zero, two, four and eight. Untransformed scenario, Hadamard likelihood.

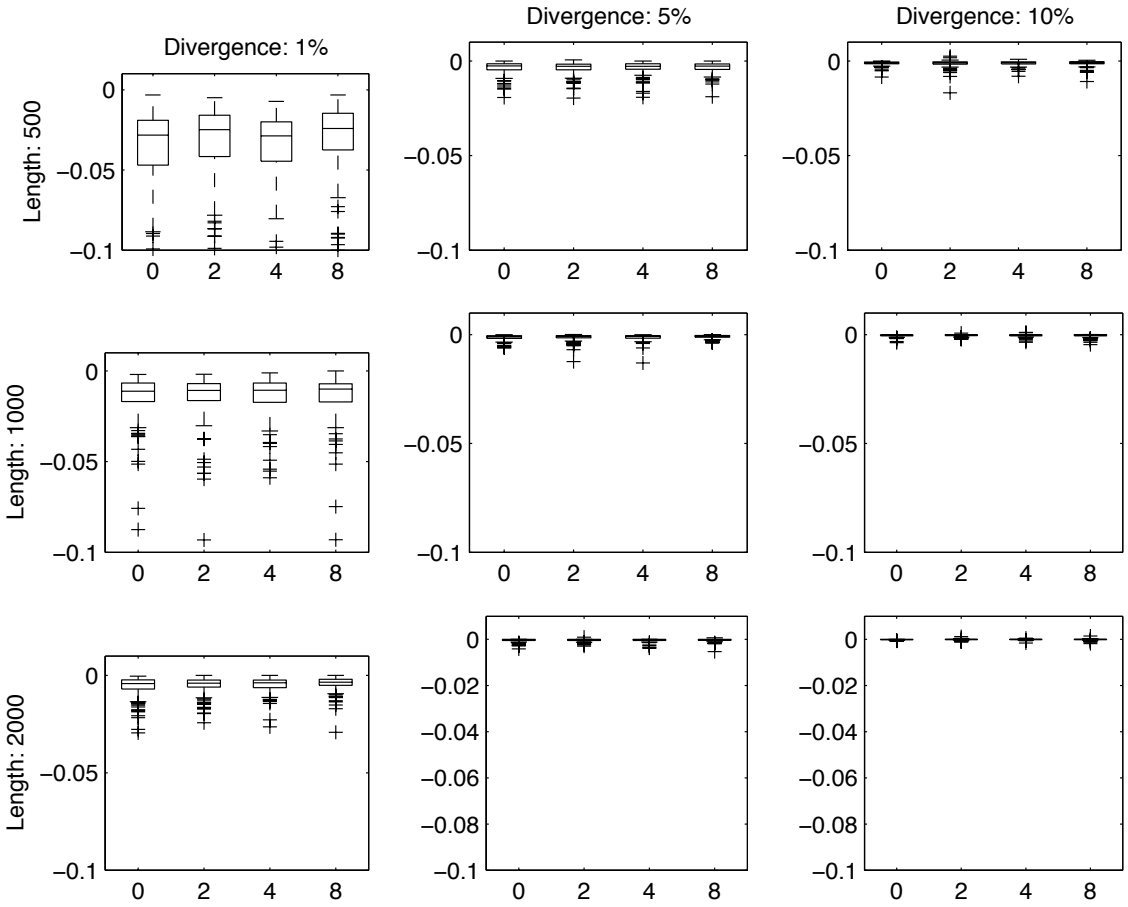

Figure F.2: BIC criterion. Box plots of the fit of the distances for sequence lengths 500, 1000 and 2000 and divergences rate 1%, 5% and 10%. Each figure contains fits for the recombinations rates zero, two, four and eight. Untransformed scenario, Hadamard likelihood.

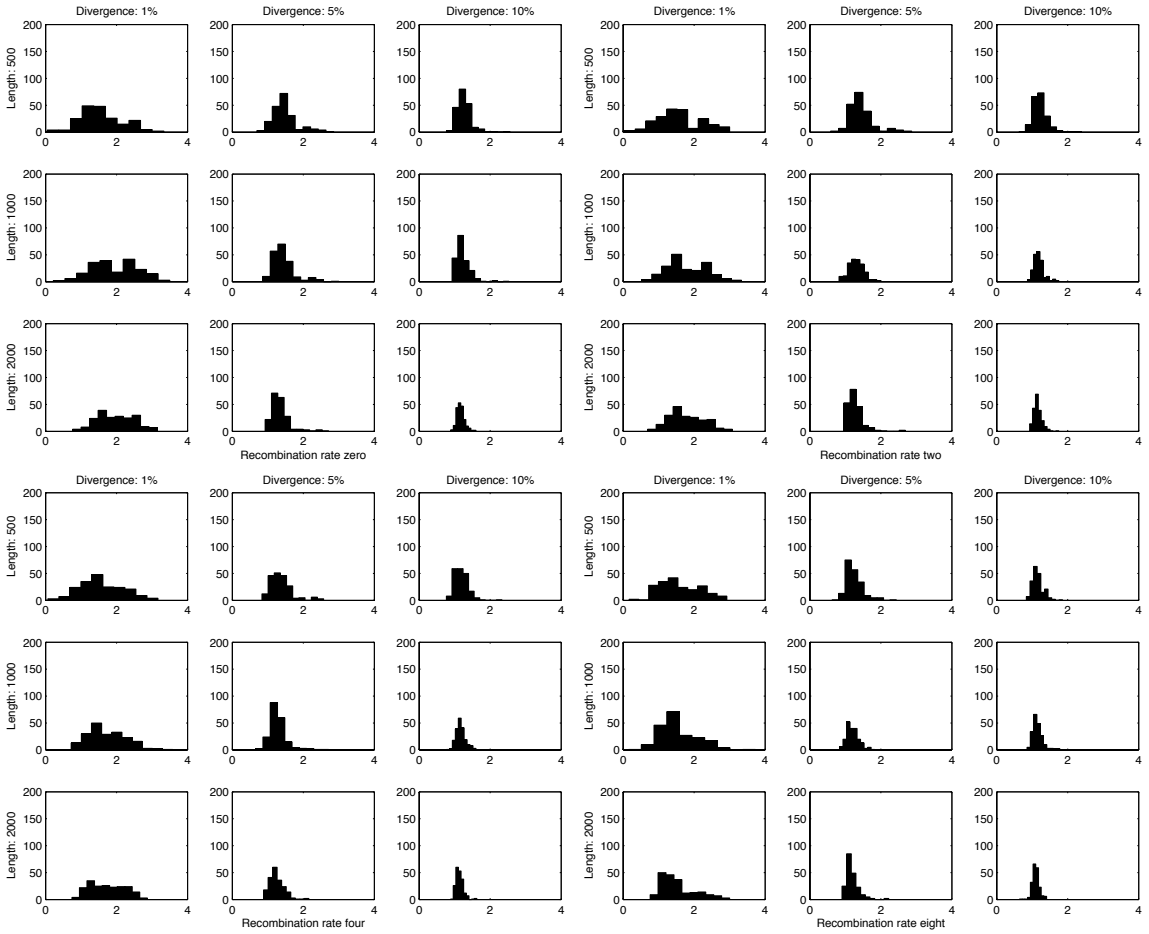

Figure F.3: Untransformed scenario, 10 taxa, Hadamard: Histograms of the number of splits chosen compared to the non-negative least squares solution based on the BIC criterion. The sequence lengths are 500, 1000 and 2000 while the divergences rates are 1%, 5% and 10%.

G

Figures based on NNLS-LASSO and $\hat{\sigma}_B^2$

This section contains the results for $\hat{\sigma}_B^2$.

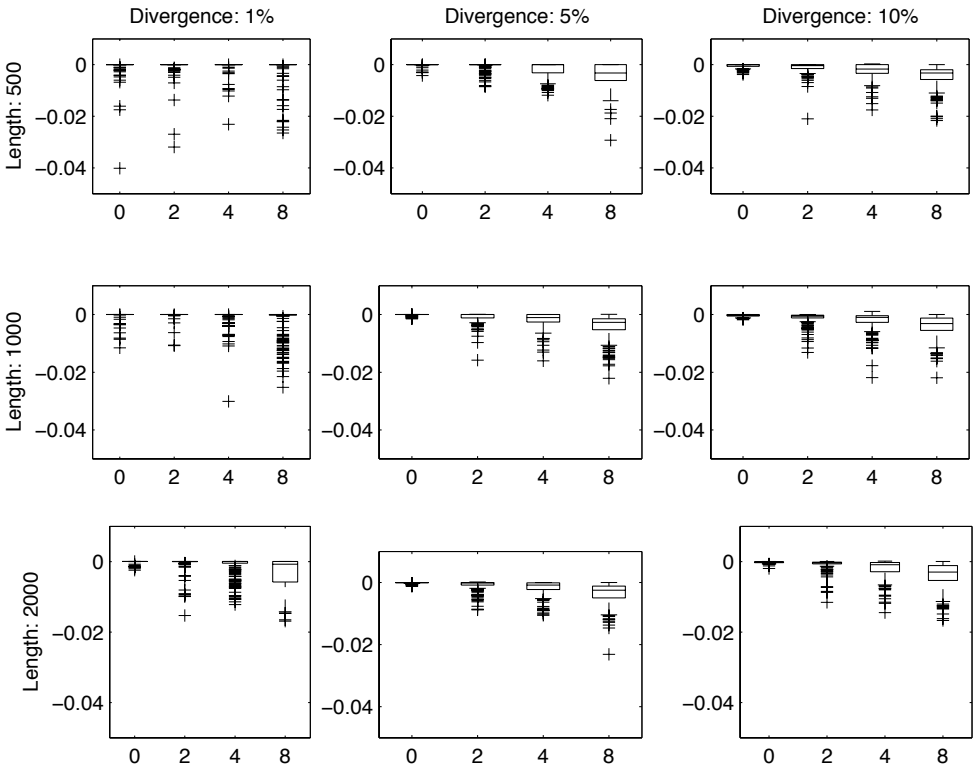

Figure G.1: Fit of the distances based on the measure $(d - \hat{d})/d$ where d is the calculated distances and $\hat{d}$ is the fitted distances. The recombination rates are on the x-axis. Number of taxa is 20, AIC criterion, 1000 replications.

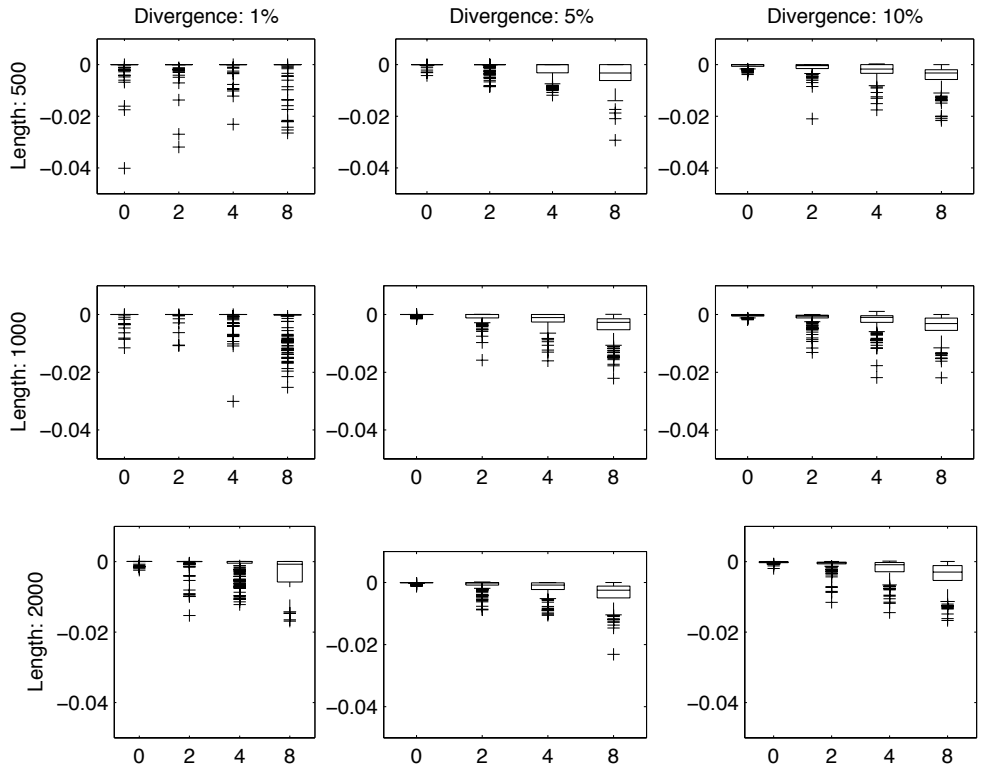

Figure G.2: Fit of the distances based on the measure $(d - \hat{d})/d$ where d is the calculated distances and $\hat{d}$ is the fitted distances. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 1000 replications.

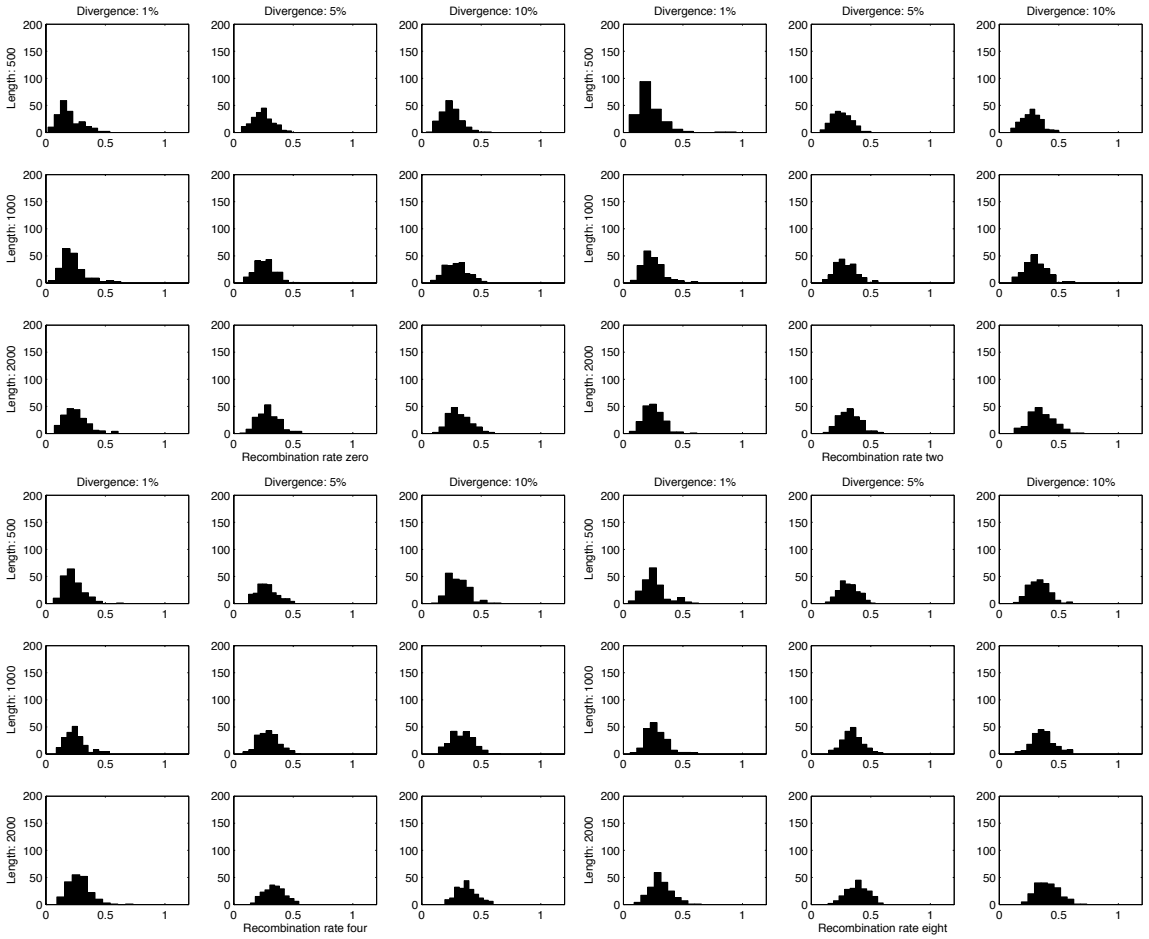

Figure G.3: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 1000 replications.

H

Figures based on the partial LASSO with NNLS hybrid and $\hat{\sigma}_T^2$

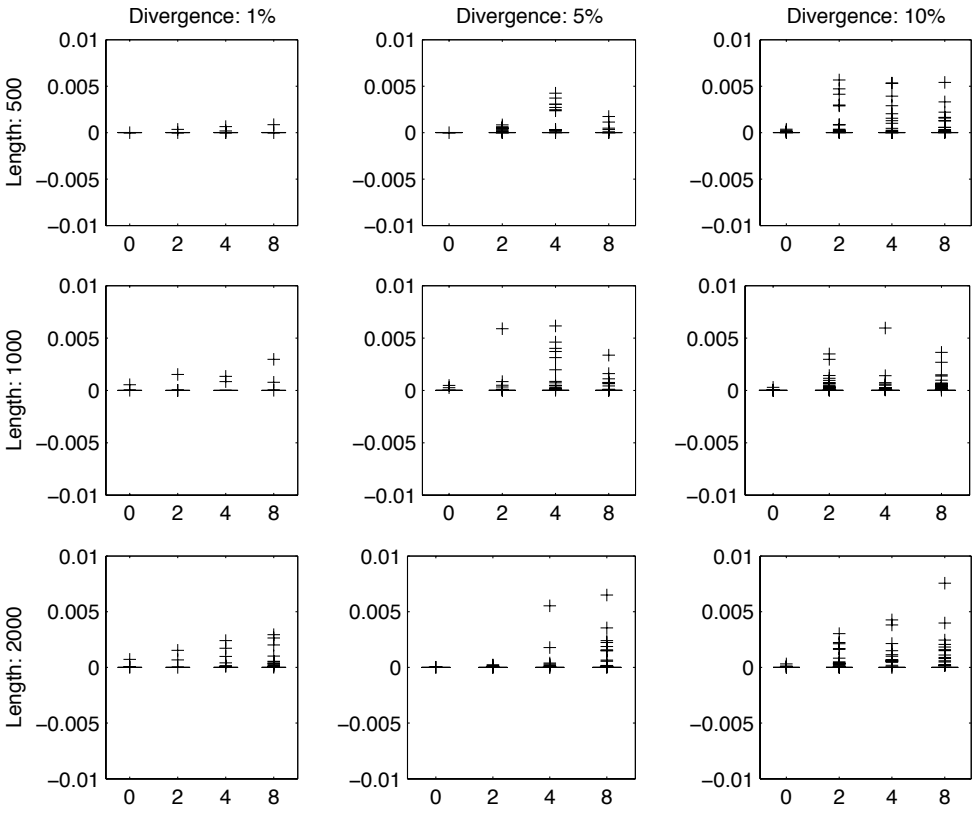

Figure H.1: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 200 replications.

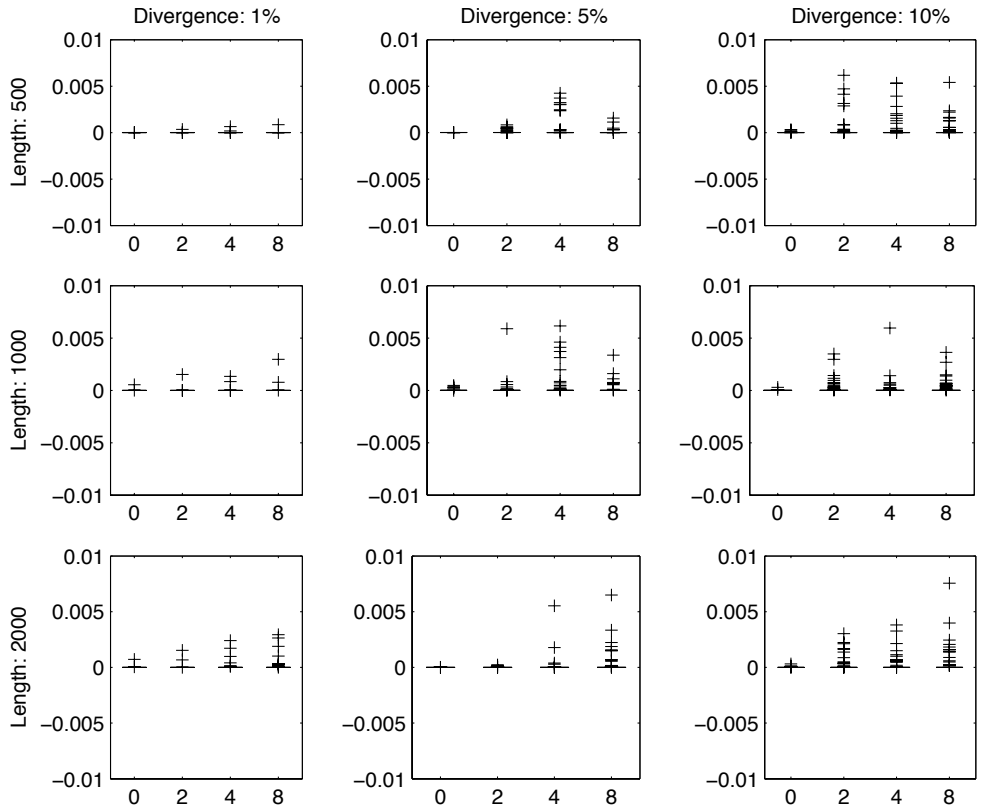

Figure H.2: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, AIC criterion, 200 replications.

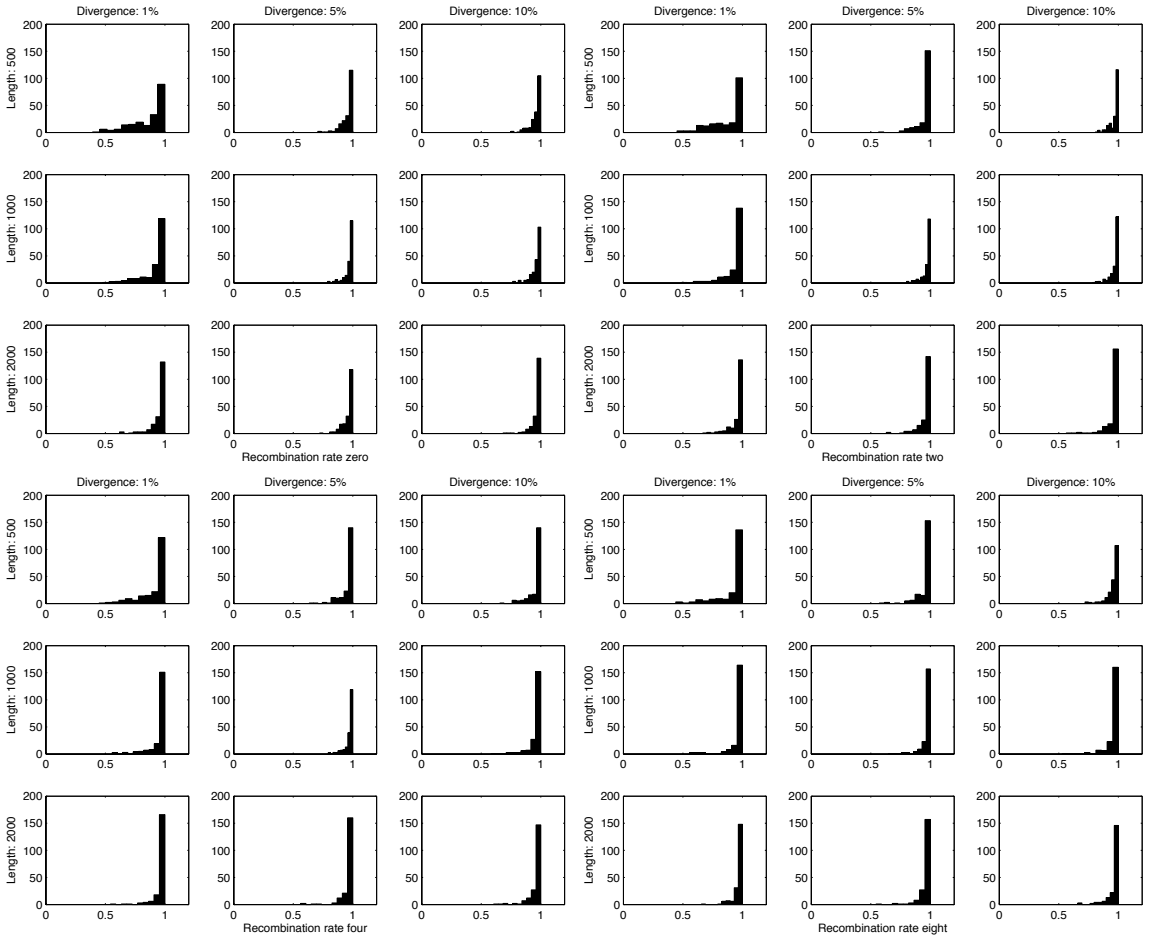

Figure H.3: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 200 replications.

I

Figures based on the partial LASSO with NNLS hybrid and $\hat{\sigma}_N^2$

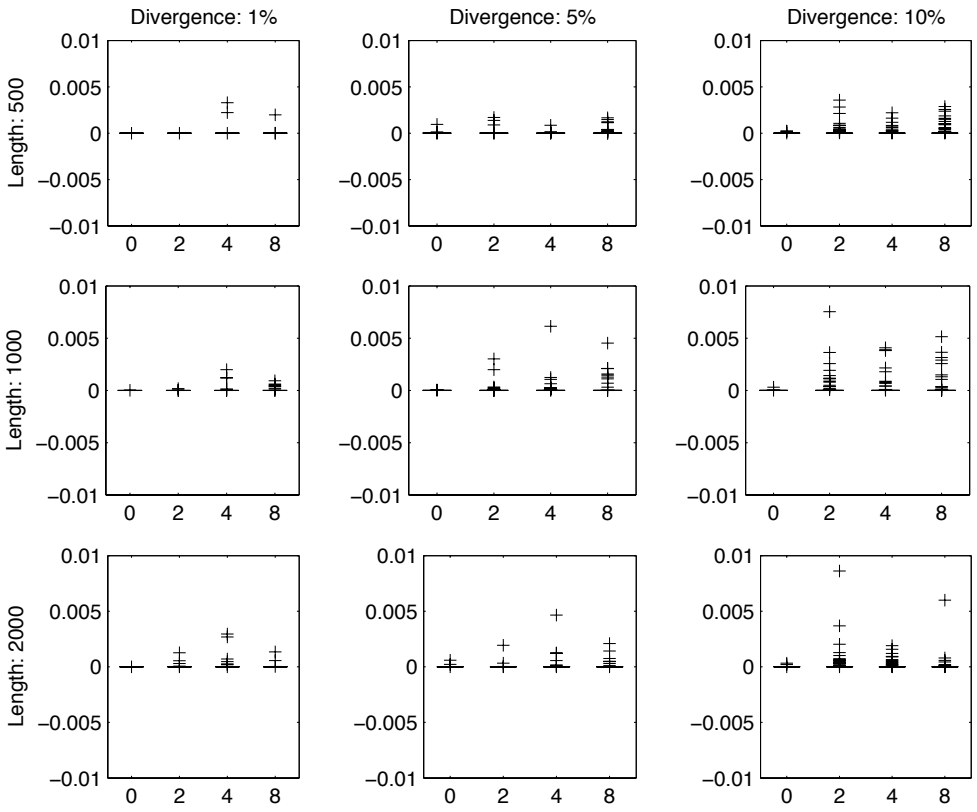

Figure I.1: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 200 replications.

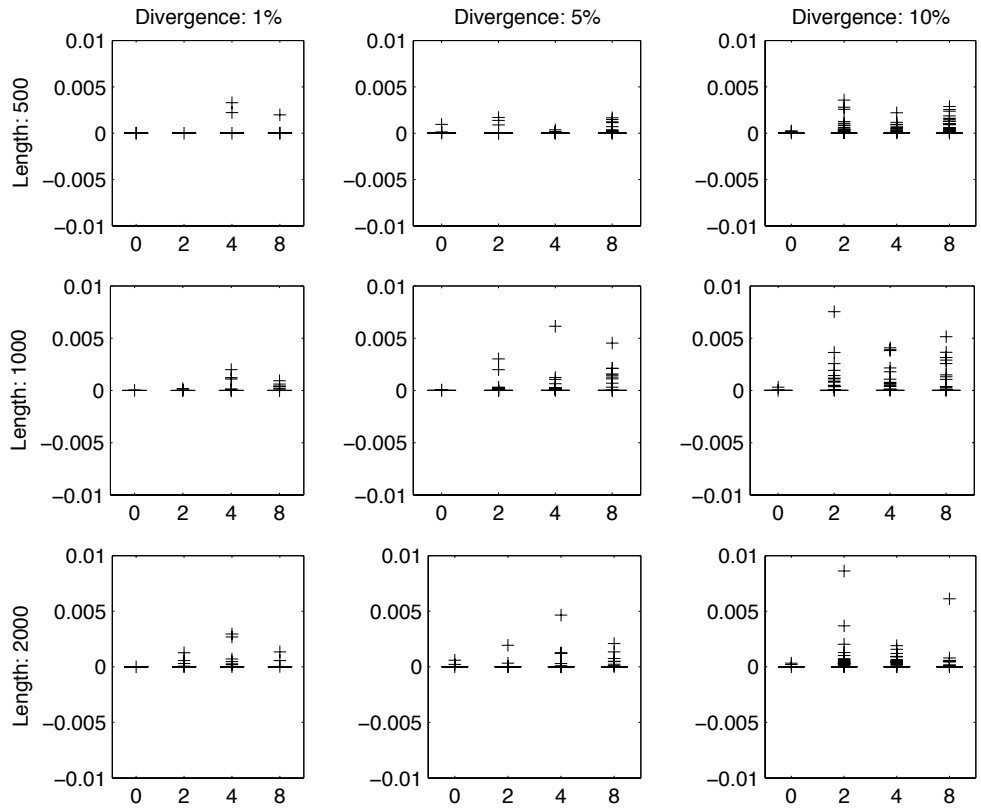

Figure I.2: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, AIC criterion, 200 replications.

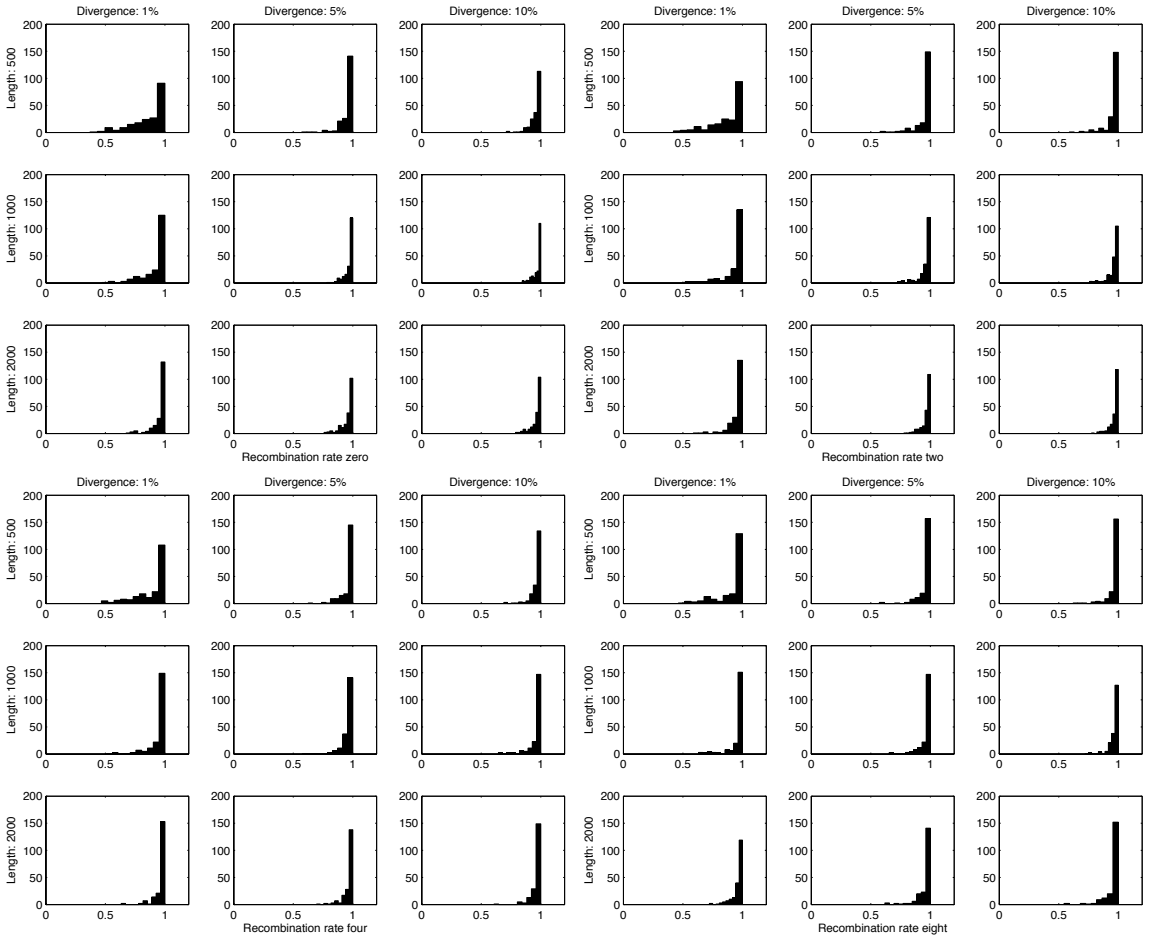

Figure I.3: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 200 replications.

J

Figures based on the partial LASSO and $\hat{\sigma}_N^2$

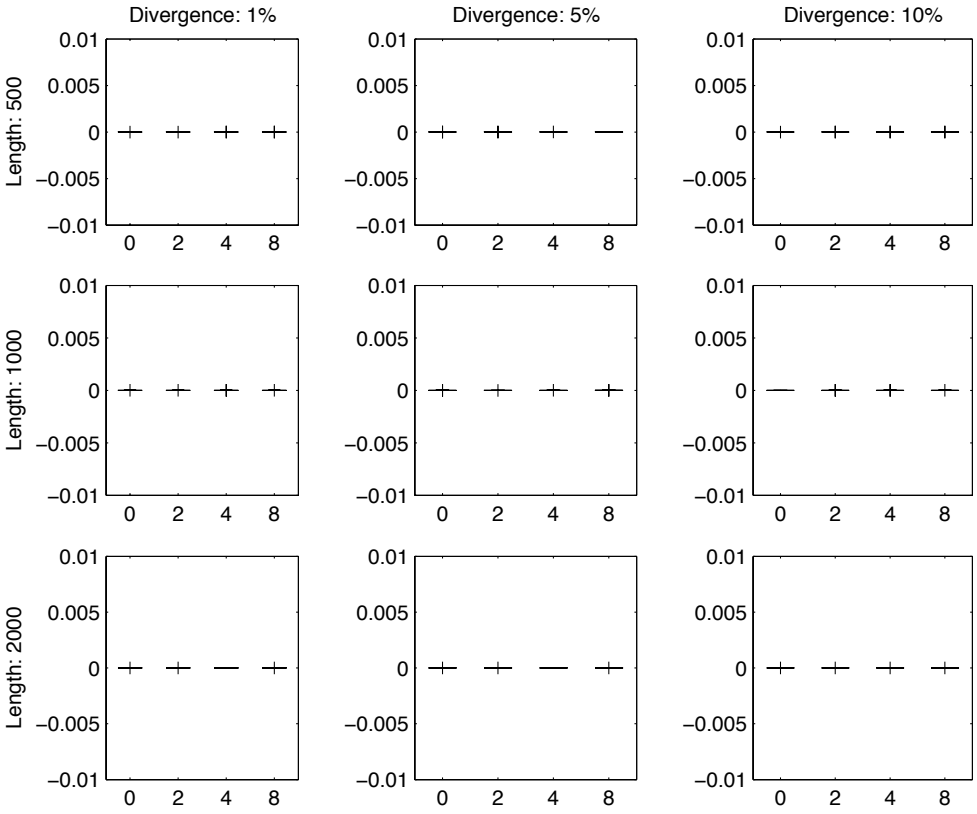

Figure J.1: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, BIC criterion, 200 replications.

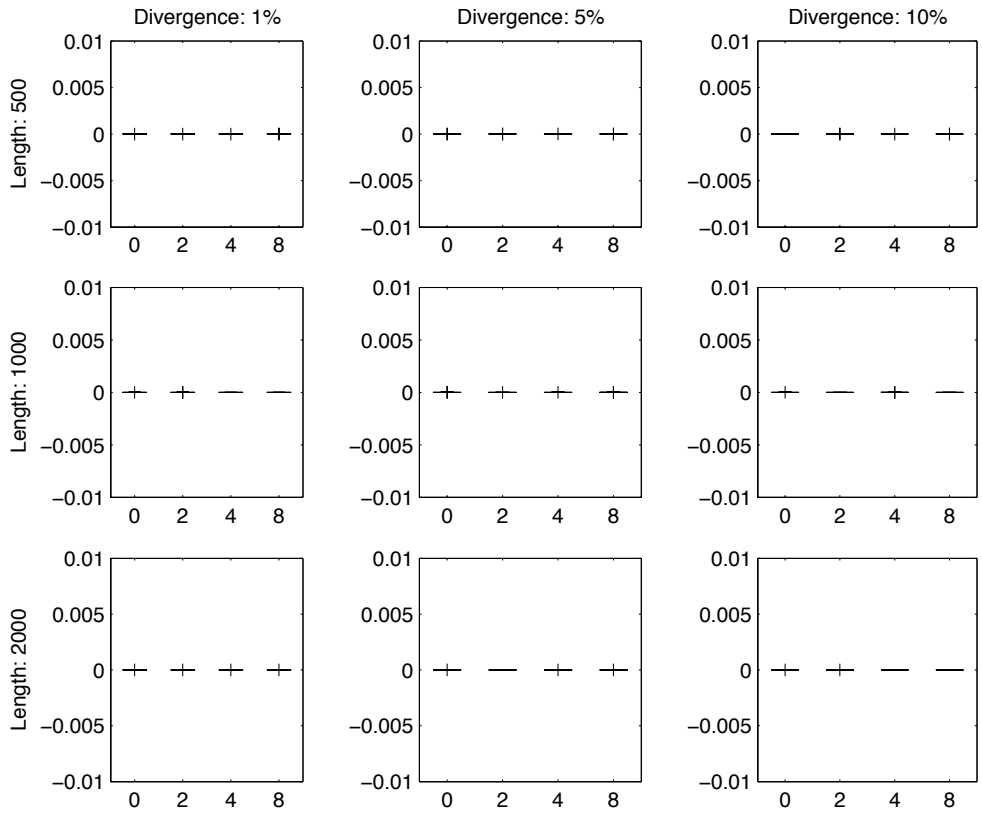

Figure J.2: Fit of the distances. The recombination rates are on the x-axis. Number of taxa, 20, AIC criterion, 200 replications.

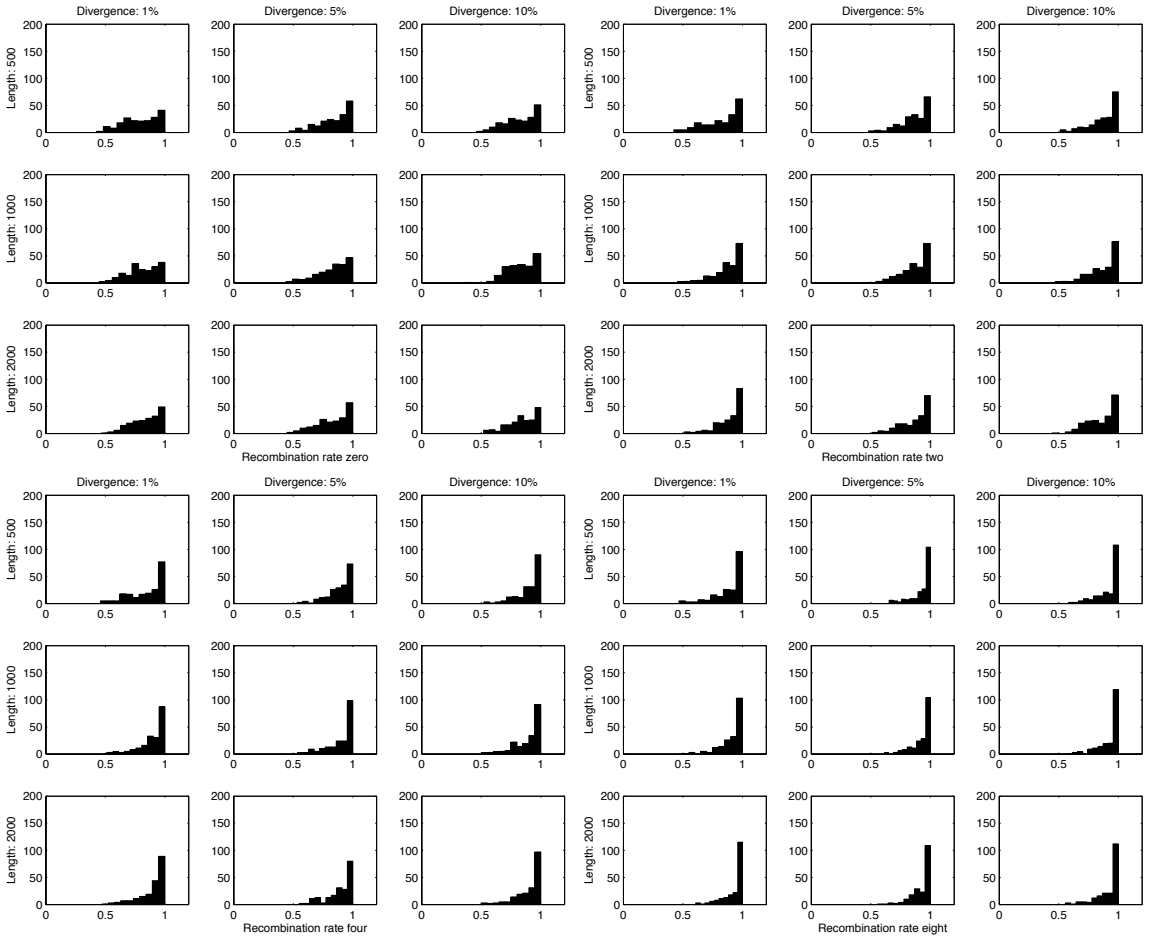

Figure J.3: Proportion of splits chosen relative to the number of splits chosen using non-negative least squares. Number of taxa is 20, BIC criterion, 200 replications.

K

Acronyms

AIC Akaike Information Criteria

ART Atheoretical Regression Trees

BIC Bayesian Information Criteria

BP Bai and Perron

COI Cytochrome oxidase I

DNA Deoxyribose Nucleic Acid

EF1a Elongation factor 1a

GARD Genetic Algorithm Recombination Detection

GTR + I + G General time reversible with Gamma and proportional invariant sites

K2P + Gamma Kimura two parameter model with Gamma distributed rates

KKT Karush-Kuhn-Tucker

LASSO Least Absolute Shrinkage and Selection Operator

MCMC Markov Chain Monte Carlo

MPR Most-Parsimonious Reconstruction

NNLS Non-Negative Least Squares

OLS Ordinary Least Squares

PHI Pairwise Homoplasy Index

RASA Relative Apparent Synapomorphy Analysis

RSS Residual sum of squares

SDNB Single Distribution Non-parametic Bootstrap

SPR Subtree Prune and Regraft

SYM + I + G Symmetrical model with Gamma and proportional invariant sites