Some Applications of Statistical Phylogenetics

A thesis presented in partial
fulfilment of the requirements

for the degree

of Doctor of Philosophy

in Biomathematics at
Massey University

Klaus Peter Schliep
2009

Abstract

The increasing availability of molecular data means that phylogenetic studies nowadays often use datasets which combine a large number of loci for many different species. This leads to a trade-off. On the one hand more complex models are preferred to account for heterogeneity in evolutionary processes. On the other hand simple models that can answer biological questions of interest that are easy to interpret and can be computed in reasonable time are favoured. This thesis focuses on four cases of phylogenetic analysis which arise from this conflict.

- It is shown that edge weight estimates can be non-identifiable if the data are simulated under a mixture model. Even if the underlying process is known the estimation and interpretation may be difficult due to the high variance of the parameters of interest.

- Partition models are commonly used to account for heterogeneity in data sets. Novel methods are presented here which allow grouping of genes under similar evolutionary constraints. A data set, containing 14 genes of the chloroplast from 19 anciently diverged species is used to find groups of co-evolving genes. The prospects and limitations of such methods are discussed.

- Penalised likelihood estimation is a useful tool for improving the performance of models and allowing for variable selection. A novel approach is presented that uses pairwise dissimilarities to visualise the data as a network. It is further shown how penalised likelihood can be used to decrease the variance of parameter estimates for mixture and partition models, allowing a more reliable analysis. Estimates for the variance and the expected number of parameters of penalised likelihood estimates are derived.

- Tree shape statistics are used to describe speciation events in macroevolution. A new tree shape statistic is introduced and the biases of different cluster methods on tree shape statistics are discussed.

Acknowledgements

I would like to thank my supervisors Michael Hendy, Barbara Holland, David Penny and Peter Waddell for their support and advice during the time of my studies. I have been blessed to have supervisors with an enormous enthusiasm for science in general and phylogenetics in particular.

I also have to thank the Marsden Fund and the Allan Wilson Centre for for financial support, which made it possible for me to study in New Zealand.

I would like to acknowledge Trish McLenachan and Gillian Gibb for their heroic effort, together with my supervisors, to proof-read this thesis and fight back my German grammar and spelling.

Many people contributed with ideas, data to the different chapters of this thesis. I have to thank Peter Lockhart and Ellen Nisbet and all other biologists who came up with challenging biological problems or supplying data. I thank Elisabeth Allman and Mark Pagel for helpful discussions about multiple optima and mixture models and Berwin Turlach for some advice on the LASSO. Some of the ideas were born or enhanced during numerous discussions with Bhalchandra, Matt, Tim, Warwick, Scott and many others, involving even more coffee.

I want to thank all the assistance from AWC staff (Joy, Susan, Karen) and IMBS (Ann, Cynthia) for doing a fabulous job. Special thanks to Tim, Warwick, Jing and Nat for taking care of my computers and software.

Thanks to all the members and visitors of the Allan Wilson centre, especially the 'boffin lounge', for creating such a friendly, multidisciplinary working environment during all my studies.

I must thank all the people who made my stay in Palmerston North such an enjoyable time. First I want to thank all the Latin Americans by passport, spouse or soul in Palmy. First of all my flatmate Rogerio, who put up with me for such a long time. Katia, Paul, Carlos, Matt and many others for all the good times at salsa classes or parties, churrascos or just at a coffee and cheesecake. Furthermore all members of the 'monkeys uncle' volleyball team and everybody I have been walking across Tongariro

with (I can't mention them all here).

I want to thank to my friends in Munich who have kept in contact with me through all this time, especially those calling during night times. The main thanks goes to my family, including a new addition, who will be mostly unaware how important their role was during all the challenges in my studies.

Contents

List of Figures

List of Tables

Abbreviations

AIC	Akaike information criterion
BIC	Bayesian information criterion
C_p	Mallows C_p
JC	Jukes-Cantor (model of nucleotide substitution)
EM-algorithm	Estimation-maximisation algorithm
GLS	General Least-Squares
GO	Gene ontology
GTR	general time-reversible (model of nucleotide substitution)
LARS	Least Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
LS	Least-Squares
MCMC	Markov Chain Monte Carlo
MDS	Multidimensional scaling
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MP	Maximum Parsimony
NJ	neighbour joining
NNI	Nearest-Neighbor Interchange
NR	Newton-Raphson
PDA	Proportional to Distinguishable Arrangements
PML	Penalised Maximum Likelihood
PNJ	Parsimony Neighbour Joining
SPR	Subtree Pruning and Regrafting
UPGMA	Unweigthed Pair-Group Mean Average
WPGMA	Weighted Pair-Group Mean Average
WLS	Weighted Least-Squares

Chapter 1

Introduction

Methods to analyse phylogenetic trees were developed when there was little sequence data available. Nowadays both the number of species and the number of loci that have been sequenced are growing rapidly. The growth in the available data provides a challenge to develop new methods. Combining different genes in an analysis adds information on the history of species, but there are also some pitfalls. Different genes can have different evolutionary histories, e.g. with lateral gene transfer in bacteria, and so for some loci the gene tree may not represent the species tree. If we ignore heterogeneity in the data, our analysis can be misled. However it is possible to account for heterogeneity by introducing additional parameters for each gene. This approach also has its disadvantages, first the variance of the estimates increases, especially if the model is overparametrised, secondly it is often hard to interpret the additional parameters, and last but not least, the computational burden can make such methods infeasible.

1.1 Structure of the thesis

In this chapter I first summarise the content of this thesis, and then introduce common phylogenetic terminology and notation. I review prior knowledge of maximum likelihood as it is used for phylogenetic analysis, and introduce the Hadamard conjugation.

These methods are essential for the later chapters. I also introduce the main data sets used in analyses in the later chapters.

In **chapter 2** I introduce mixture models. These were first applied in phylogenetics to account for heterogeneity of data. When using mixture models we do not assume a single underlying structure, like the partitions of genes or codon position, but assume that several underlying processes generated the data and integrate over these processes. In phylogenetic analysis mixture models are commonly used to model site heterogeneity. The estimation of the proportion of invariant sites and the discrete Γ-model are special cases of mixture models. Although more complex mixture models (especially mixtures of trees) have been used to construct datasets and counterexamples of evolutionary processes (Kolaczkowski and Thornton, 2004), inference using these models is not common. An exception is the program BayesPhylogenies, where Pagel and Meade (2004) implemented mixture models for phylogenetic analysis using an MCMC approach allowing the estimation of mixtures of rate matrices or mixtures of different trees. Matsen and Steel (2007) and Stefankovic and Vigoda (2007a) showed some examples where a phylogenetic mixture of trees is not identifiable. In this chapter I focus on a common problem with mixture models, that is due to the often high number of additional parameters, mixture models can overfit data. Often parameters of the mixtures are correlated leading to a high variance and also slow convergence of the estimation. The results from chapter 2 are used in the next chapter.

Chapter 3 addresses the problem of multiple optima in phylogenetic tree reconstruction. There are two main instances in which multiple optima can occur in phylogenetics. First in tree space, there may exist several tree topologies, with similar high likelihood values (or other scores), which cannot be connected by a single tree rearrangement (for example by NNI, SPR or TBR, which I will define later). The second are multiple optima on a fixed topology, i.e. different sets of edge lengths which lead to local optima. I will focus on this second problem in chapter 3. Whereas in former studies this was only described theoretically, I have constructed examples where data generated from mixture models can result in multiple optima on a single tree.

In **chapter 4** I move to a different topic and discuss how to cluster genes into groups that appear to have evolved in a similar fashion. We can then use the clusters as partitions for the model in further analyses. This can reduce the number of parameters considerably if we only need to select parameters for each cluster (and not every gene) without losing important information about the individual genes. I implemented this in a new algorithm which is based on an idea similar to the k-means algorithm. In addition the clusters themselves are often of interest: for example to test if genes involved in a biological pathway have co-evolved. If we assume that genes that evolved under similar constraints show similar evolutionary patterns, we can try to find candidates for co-evolving gene families. We can compare our predictions with information already available about the biological function of genes from other sources e.g. the Gene Ontology (GO) database (Ashburner et al., 2000).

Chapters 5 and 6 are related in that both cover aspects of penalized likelihood.

Chapter 5 introduces two methods to estimate phylogenetic networks based on the LASSO (Tibshirani, 1996; Osborne et al., 2000) and ridge regression (Hoerl and Kennard, 1970). Both methods can be useful to visualise conflicting signal in phylogenetic data. The first uses distance based methods and the second the Distance Hadamard (Hendy and Penny, 1993). The LASSO and ridge regression do not appear to have been fully explored in phylogenetics.

Chapter 6 follows naturally from chapters 4 and 5 in that it discusses another way to reduce the number of parameters needed for the estimation of edge weights in partition models. If the parameters of the phylogenetic model have a high variance, a penalised likelihood (PML) approach can often help to reduce the variability of the coefficients (Hastie et al. (2001)). Especially when allowing for independent sets of edge weights in mixture (chapter 3) or partition models (chapter 4), the number of parameters will often be high, but many parameters may be correlated with each other. Penalised likelihood shrinks the parameters of corresponding edges towards each other and also reduces the variance. I developed an estimate of the effective number of parameters, which allows us to compare estimates from a PML via the AIC or BIC,

with other likelihood models. PML can not only be used to reduce the variance of the coefficients, but also to select important subsets of variables (Tibshirani, 1996).

Chapter 7 addresses the biases in hierarchical clustering. Hierarchical clustering algorithms are often based on distance matrices and provide a fast heuristic for tree estimation with computational complexity of $O(n^2)$ or $O(n^3)$, for n taxonomic units. These heuristics are often the only feasible method to estimate trees on many thousands of taxa, and are also often used to produce starting trees for further maximum likelihood or parsimony searches. Hierarchical cluster methods always return binary trees, even when internal edges are tiny and multifurcations of edges might be a better choice. Depending on the method used, these tree shape statistics can vary strongly between different methods. In this chapter I explore what biases different methods and what effect this has on tree estimation.

Most of the new models described above are available in the form of the R-package I wrote called *phangorn* at `http://www.cran.r-project.org`. **Appendix A** provides some description of how I implemented these new methods. When possible the algorithms were checked against existing software (PAUP*, phyML, BayesPhylogenies).

1.2 Background

1.2.1 Graphs, trees and networks

A *graph* $G = (V, E)$ consists of a set of nodes (sometimes called vertices or points) V, and a set of edges E. Let an edge in the graph $e \in E$ be an unordered pair $e = \{a, b\}$, where $a, b \in V$, then a and b are *adjacent*, and e is *incident* to a and b. The *degree* of a node is the number of its incident edges.

A *path* is a sequence of nodes $v_1, \ldots, v_m$ such that nodes v_i, v_{i+1}, for $i = 1, \ldots, m-1$, are adjacent. A *cycle* is a path which starts and ends at the same node. A graph G is said to be *connected* if each pair of nodes in G can be joined by a path. We will call the minimal number of edges on the path between two nodes v_i and v_j the *path length* of v_i and v_j.

A *tree* T is a connected graph without cycles. Nodes of degree one are called tips, external nodes or leaves. Nodes of degree higher than one are called internal nodes. A *phylogenetic tree* (or phylogeny) is a tip-labeled tree with label set X of taxa.

A *binary* tree has only nodes of degree one or three. A tree with nodes with degree higher than three is called a *multifurcating* tree and a node with degree higher than three is a *multifurcation*. Figure 1.1 a) and c) show binary trees and b) shows a multifurcating tree.

A *weighted* tree has weights (or lengths) assigned to each edge. In a phylogeny the edge weights are all non-negative real numbers and can represent time or the expected number of substitutions of an edge. A weighted tree induces a (pseudo-)metric. The distance between two nodes v_1 and v_2 is the sum of the weights along the path between these nodes.

A *rooted* tree is a directed acyclic graph (DAG) where all edges are directed away from a special node called the root. In a *binary rooted tree* the root has degree two, all other nodes are of degree one or three. Figure 1.1 b) and c) display rooted trees. A special case of weighted rooted trees are *ultra-metric trees* where the distance between each tip and the root is constant for all tips. In phylogenetics we speak of a *clock-like*

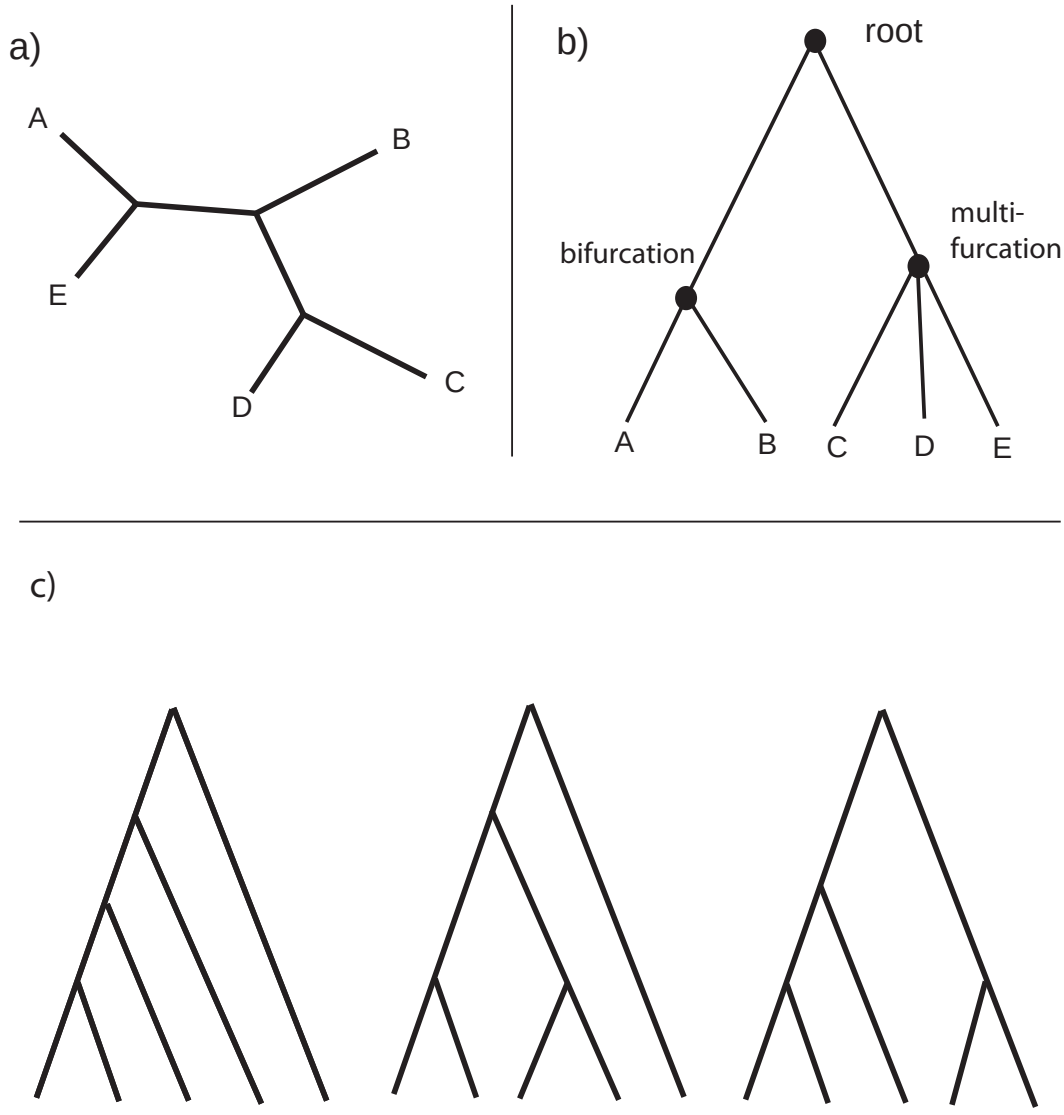


Figure 1.1: a) shows a bifurcating unrooted phylogenetic tree. b) shows a rooted tree, on the internal node on the left is a bifurcation and the node on the right is a multi-furcation. In panel c) we show all three tree shapes for rooted (unlabeled) trees with 5 tips. The biological interpretation of these graphs as evolutionary trees is discussed in Gregory (2008)

tree when we associate time with the weights of an ultrametric tree.

We refer to unlabeled trees as *tree shapes*. Figure 1.1 c) shows the three possible tree shapes for a rooted bifurcating tree with 5 tips.

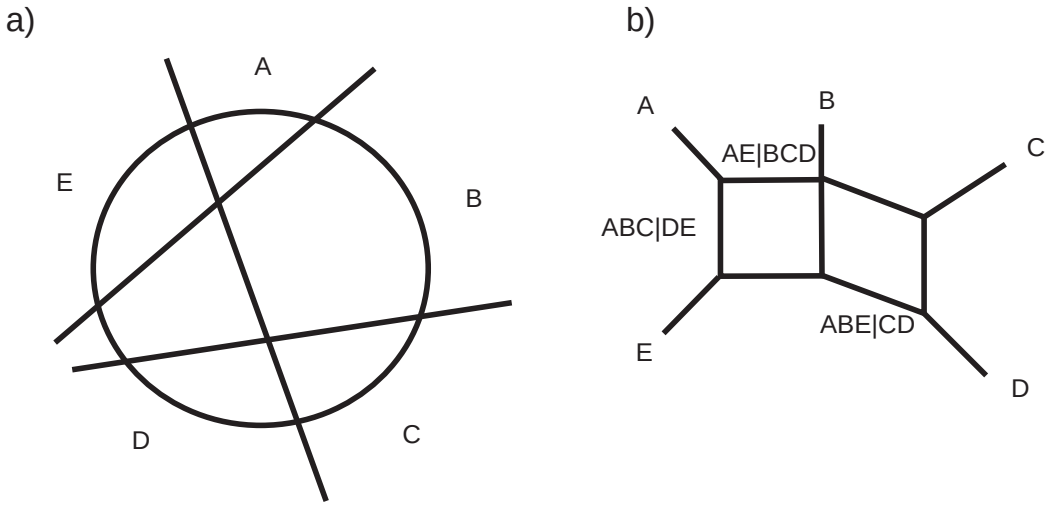


Figure 1.2: a) A circular split system, the lines represent the splits and only non-trivial splits are shown. The associated splits graph is illustrated in panel b) where parallel edges represent the same split.

Given a set of taxa (labels, species) X, a *split* is a partition of X into two non-empty subsets A and B, which we write as $A|B$. Two splits $A_1|B_1$ and $A_2|B_2$ are *compatible* if at least one of the intersections $A_1 \cap A_2$, $A_1 \cap B_2$, $B_1 \cap A_2$ and $B_1 \cap B_2$ is the empty set, otherwise they are *incompatible*. Each edge $e = \{x, y\}$ in a phylogenetic tree induces a split $A|B$ of the tip label set X, removing the edge from a phylogenetic tree creates two components (subtrees), one with label set A and one with label set B. Each pair of splits induced by edges of a phylogenetic tree are compatible and no pair of incompatible splits can be represented on any phylogenetic tree. If we have incompatible splits they can be represented on a *split graph*, that is a connected graph with cycles with tips labelled by X (see figure 1.2), where a set of parallel edges correspond to a split $A|B$, with A and B being the labels of the subgraphs when the parallel edges are deleted.

1.2.2 Data

The data we start with in a molecular phylogenetic analysis are usually *alignments* of homologous sequences. An alignment is a matrix X, where each row contains the sequence of a taxon (short for taxonomic unit), i.e. an organism or a species. The columns of an aligned sequence are referred to as sites and contain homologous characters, which means that they have evolved from a common ancestral character.

The characters in the alignment are most often nucleotides for DNA or RNA sequences with 4 different states, or amino acids for protein sequences (20 states). Fast evolving sites are sometimes RY-coded and modeled with 2-state characters.

Sites	1	2	3	4	5	6	7	8	9	10
S. cerevisiae	T	T	A	T	T	G	A	C	G	T
S. paradoxus	T	T	G	T	T	A	A	C	G	T
S. mikatae	T	T	G	C	T	A	A	C	A	T
S. kudiavzelli	T	T	G	C	T	A	A	C	G	T
S. bayanus	T	T	A	C	T	A	A	C	G	T
S. castellii	C	T	A	T	T	A	A	C	A	T
S. kluyveri	C	T	T	C	T	A	A	C	G	T
C. albicans	C	T	T	T	T	G	A	C	A	T

Table 1.1: 10 sites of an alignment of 8 species of yeast.

In table 1.1 we see 10 bases of a DNA alignment of eight species of yeast. Most phylogenetic methods make the assumption that the columns/sites in an alignment are independent.

1.2.3 Methods of tree estimation

Methods of tree estimation are algorithms which take an input data set and return a phylogenetic tree. We can distinguish methods by

- the input data
- the optimality criterion
- the search in tree space

We call a method a *distance* or *dissimilarity* method if it uses a distance or dissimilarity matrix as input data. In contrast (maximum) parsimony, and what we refer to in phylogenetics as (maximum) likelihood, uses sequence alignments as input. An *optimality criterion* gives a score to a tree, given the data. For the parsimony method, the score is the minimal number of substitutions needed to account for the data on a phylogeny. For the likelihood method the score is the likelihood (probability) of observing the sequence data given the phylogeny and a model of (nucleotide) substitutions. For distance methods the criterion is often a least squares (LS) or weighted least squares criterion fitting input distances to path lengths on a weighted phylogeny.

Finally we describe how a method searches in tree space. The number of possible trees grows super-exponentially with the number of taxa. For unrooted bifurcating trees with n labelled tips, the number of trees is

$$(2n-5)!! = 1 \times 3 \times 5 \times \ldots \times (2n-5).$$

For $n = 20$ the number of trees is about 2.22×10^{20}. It is infeasible to evaluate a function on 10^{20} individual trees. To find good trees, i.e. trees that optimise an optimality criterion or attain a value expected to be close to the optimum, one needs good heuristics or branch and bound searches.

1.2.4 Tree rearrangements

Here we describe two methods of tree rearrangements, SPR and NNI, which are used to make local searches in tree space. The idea behind these heuristics is to use a starting tree and search locally for improved scores (parsimony, maximum likelihood, LS), until no local rearrangement can lead to a tree with a better score.

Nearest-Neighbour Interchange (NNI)

For any internal edge of a binary tree there exist three different ways to connect its four subtrees, one of which is the current tree. An unrooted binary tree with n taxa

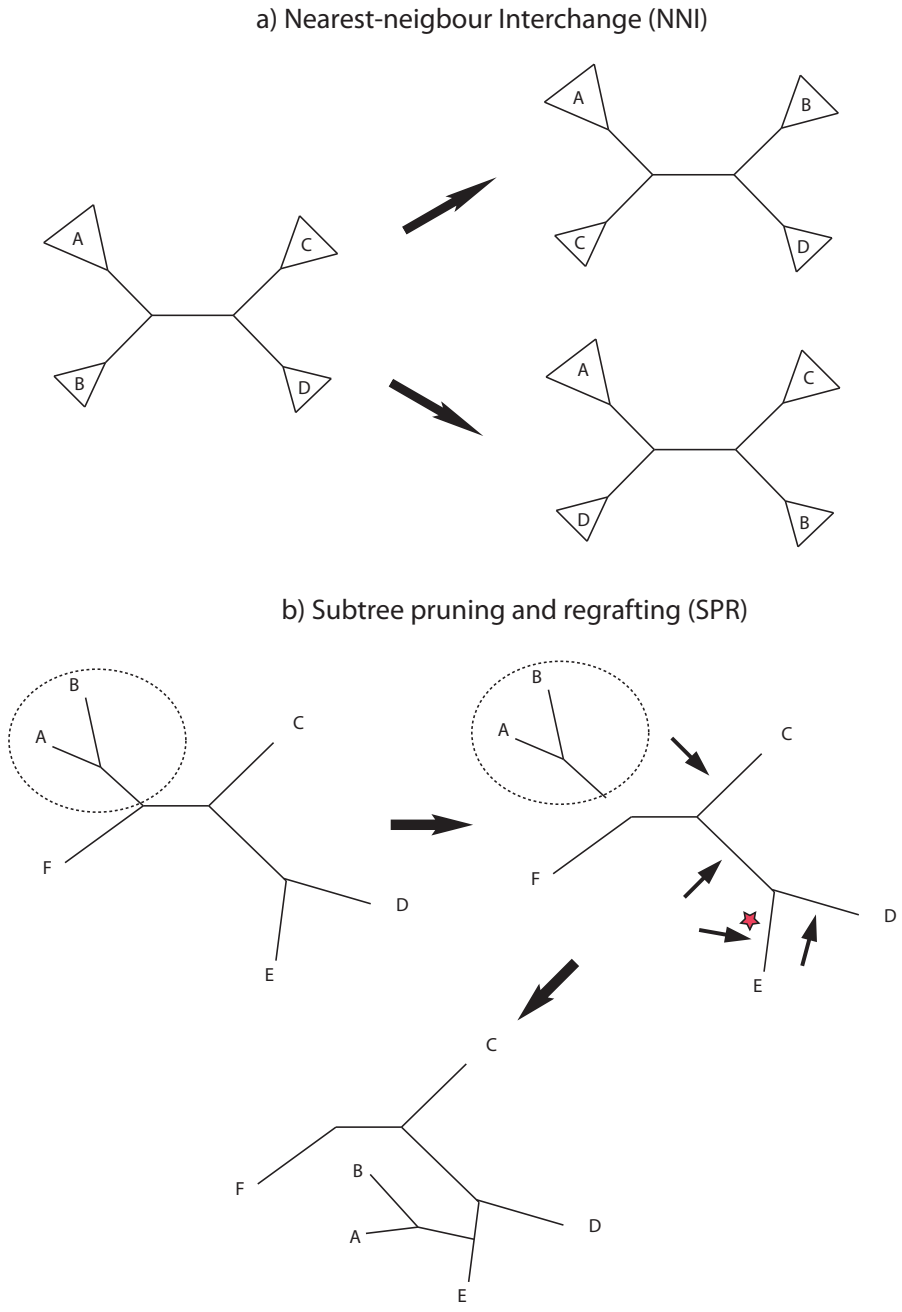


Figure 1.3: a) Nearest-Neighbour Interchange (NNI), for every interior edge there are always 3 different ways to connect four subtrees. The arrows indicate which subtrees have been replaced. b) An example of Subtree Pruning and Regrafting (SPR), first a subtree is removed from the graph, arrows indicates places where the subtree could be reattached. The star indicates the edge where the subtree is reattached.

has (n-3) internal edges so we can reach $2n$-6 different trees in one step with NNI (see figure 1.3 a).

Subtree Pruning and Regrafting (SPR)

NNI only exchanges subtrees which are separated by one edge. Subtree pruning and regrafting (SPR) is a rearrangement strategy which can exchange more distant subtrees. First a subtree is removed from the graph (see figure 1.3 b)) which then can be attached to any other edge of the tree. The total number of neighbouring trees which are adjacent using one SPR-move depends on the tree shape, but is in the order of $O(n^2)$, compared to $O(n)$ for NNI. Each NNI move is an SPR move.

1.2.5 Markov models of character evolution

To model nucleotide substitutions across the edges of a tree T we can assign a transition matrix to each edge of T. In the case of DNA/RNA data, with four character states, each 4×4 transition matrix has, at most, 12 free parameters.

Time-reversible Markov models are used to describe how characters change over time, and use fewer parameters. Time-reversible means that these models need not be directed in time, and the Markov property states that these models depend only on the current state. These models are used in analysis of phylogenies using maximum likelihood and MCMC, as well in simulating sequence evolution.

We will now describe the General Time-Reversible (GTR) model. The parameters of the GTR model are the equilibrium frequencies $\pi = (\pi_A, \pi_C, \pi_G, \pi_T)$ and a rate matrix Q which has the form

$$Q = \begin{pmatrix} \star & \alpha\pi_C & \beta\pi_G & \gamma\pi_T \\ \alpha\pi_A & \star & \delta\pi_G & \epsilon\pi_T \\ \beta\pi_A & \delta\pi_C & \star & \eta\pi_T \\ \gamma\pi_A & \epsilon\pi_C & \eta\pi_G & \star \end{pmatrix} \tag{1.1}$$

where we assign 6 parameters $\alpha, \ldots, \eta$. The elements on the diagonal are chosen so that

the rows sum to zero. The Jukes-Cantor (JC) (Jukes and Cantor, 1969) and Kimura's three-substitution-types (K3ST) (Kimura, 1981) model can be derived as special cases from the GTR model, for equal equilibrium frequencies $\pi_A = \pi_C = \pi_G = \pi_T = 0.25$ and equal rates set to ($\alpha = \beta = \gamma = \delta = \epsilon = \eta$) for the JC model and $\alpha = \eta$, $\beta = \epsilon$, $\gamma = \delta$ in case of the K3ST model. The transition probabilities which describe the probabilities of change from character i to j in time t, are given by the corresponding entries of the matrix exponential

$$P(t) = (p_{ij}(t)) = e^{Qt}, \qquad \sum_j p_{ij} = 1, \tag{1.2}$$

where $P(t)$ is the transition matrix across an edge spanning a period of time t.

Mathematical notation

In mathematical formulae in this thesis capital letters represent matrices, vectors are printed in lower case v and are in general, column vectors. The transpose of a vector is v'. Estimates of a parameter are identified by a hat, e.g. $\hat{\theta}$ is the estimate for the parameter θ.

1.3 Maximum likelihood estimation in phylogenetics

Felsenstein (1981) introduced Maximum Likelihood (ML) to estimate phylogenetic trees for sequence data. We will closely follow the definition of Yang (2000, 2006) who defines the likelihood

$$L(\theta, x) = \prod_{i=1}^{N} f(x_i|\theta) \tag{1.3}$$

where $f(x_i|\theta)$ is the probability of observing data x at site i given the model parameters θ, and N is the number of sites in the alignment. If we assume the site patterns are independently and identically distributed (i.i.d.), we can pool the common patterns,

and express equation (1.3) as

$$L(\theta, x) = \prod_{j=1}^{m} f(x_j|\theta)^{n_j}, \qquad \sum_{j=1}^{m} n_j = N, \tag{1.4}$$

where n_j is the frequency of the j-th site pattern and there are m unique patterns. It is common to maximise the log-likelihood function

$$l(\theta) = \sum_{i=1}^{N} \log\left(f(x_i|\theta)\right) = \sum_{j=1}^{m} n_j \log\left(f(x_j|\theta)\right) \tag{1.5}$$

which also maximises $L(\theta, x)$. $l(\theta)$ is bounded above by its maximum over observation space which occurs when $f(x_j|\theta) = n_j/N$.

To estimate the (log-)likelihood of a tree Felsenstein (1981) introduced the pruning algorithm. Assume nodes j and k have a direct ancestor h then we can estimate the conditional likelihood

$$L_h(x_h) = \left(\sum_{x_j} L_j(x_j) p_{x_j,x_h}(t_j)\right) \times \left(\sum_{x_k} L_k(x_k) p_{x_k,x_h}(t_k)\right) \tag{1.6}$$

The likelihood of the tree can be evaluated by traversing through the tree in postorder from the tips towards the root - for unrooted trees a root can be chosen arbitrarily as our models are time-reversible. We get the likelihood of the tree if we multiply the conditional likelihood of the root node r with the base composition π, as

$$f_h(x|\theta) = \sum_{x_r} \pi_{x_r} L_r(x_r), \tag{1.7}$$

where π is a vector containing the base frequencies.

1.3.1 Optimising the likelihood

We now focus on optimising the likelihood as the edge parameters are varied. There are different strategies for optimising $L(\theta)$. One can optimise all edges at the same

time using a multidimensional algorithm or optimise the edges separately, and iterate until convergence (e.g. Gauss-Seidel algorithm).

Furthermore we can distinguish between derivative-free algorithms, and those which use derivatives of the (log-)likelihood function $l(\theta)$. The rate of convergence for derivative-free methods is linear, but for methods using first and second derivatives, convergence can be quadratic (Press et al., 1992, p. 364f).

One way to optimise the likelihood is with the Newton-Raphson (NR) algorithm:

$$\begin{aligned} \hat{\theta}^{k+1} &= \theta^k - \mathcal{H}(\theta^k)^{-1} sc(\theta^k) \\ sc(\theta) &= \frac{\partial l(\theta)}{\partial \theta} \\ \mathcal{H}(\theta) &= \frac{\partial^2 l(\theta)}{\partial \theta \partial \theta'} \end{aligned} \tag{1.8}$$

The matrix of second derivatives $\mathcal{H}(\theta)$ is known as the Hessian matrix, and the vector of first derivatives $sc(\theta)$ is referred as the score vector. At a local optimum, the score vector is 0. The inverse of the negative Hessian is an estimator of the covariance matrix

$$cov(\theta) = \mathcal{H}(\theta) = \left(-\frac{\partial^2 l(\theta)}{\partial \theta \partial \theta'} \right)^{-1} \tag{1.9}$$

The NR algorithm converges quadratically, but it may diverge for poorly chosen starting values. Also the inversion of the Hessian is often numerically unstable. In statistics quite often a variant of the NR is used, the Fisher-scoring algorithm. The Hessian matrix of second derivatives is replaced with the expected Fisher Information matrix:

$$\mathcal{I}_\theta = -\mathrm{E}\left(\frac{\partial^2 l(\theta; x)}{\partial \theta_i \partial \theta_j} \right) = \mathrm{E}\left[\left(\frac{\partial l}{\partial \theta_i} \right) \left(\frac{\partial l}{\partial \theta_i} \right) \right] \tag{1.10}$$

The asymptotic covariance $cov(\hat{\theta})$ is under regularity conditions (Rao and Toutenburg, 1995) bounded below by the inverse of the information matrix $\mathcal{I}_{\hat{\theta}}$.

Fisher-scoring has two advantages over NR, the matrix (1.10) is always positive semi-definite and it only uses the first derivative, however the convergence is not necessarily

quadratic.

The most common approach (e.g. as implemented in PAUP* Swofford (2002)) is to optimise the edges separately using a one-dimensional NR algorithm and iteratively optimize one edge after another, until convergence. Instead of the NR, one can use quasi-Newton methods or derivative free methods like the Nelder-Mead algorithm or the EM-algorithm (Dempster et al., 1977). Not as common in phylogenetics is the usage of the multivariate NR method, however this would have the positive side effect of giving the covariance matrix.

As the derivatives of the $l(\theta)$ are easy to derive for edge length parameters (Yang, 2000) these are often optimised using NR. For most of the other model parameters, like the rate matrix and nucleotide composition the derivatives are unknown or not easily accessible and derivative-free methods are more frequently used.

Invariance of transformations

Let θ and ϕ be two alternative parametrisations with $\phi = g(\theta)$, where g is a monotone transformation. If $\hat{\theta}$ is the maximum likelihood estimator (MLE) for θ, then $g(\hat{\theta})$ is the MLE for ϕ (Garthwaite et al., 2002, p. 49). This means that, since the edge lengths θ are non-negative, we can use the natural logarithm of θ, as it better discriminates shorter edge lengths, which are common. If the length of an edge is zero we can omit this edge from the topology which introduces a multifurcation. Taking the logarithm allows one to optimise the edge lengths without using constraint optimisation techniques for ensuring non-negativity, often this makes the computation easier and faster.

1.3.2 Hypothesis testing

One big advantage of maximum likelihood estimation is that there exist many statistical tests to compare estimates.

There are many ways to test the significance of a hypothesis $H_0 : \theta = \theta_0$, see for example Agresti (2002) p. 12 or Fahrmeir and Tutz (1997) p. 45. Most commonly used is the likelihood ratio test (LRT) for nested hypotheses of the form $H_0 : \theta_1 = 0$, for

$\theta = (\theta_0, \theta_1)$,

$$LR = 2(l(\theta) - l(\theta_0)), \tag{1.11}$$

which compares the unrestricted maximum $l(\theta)$ with the restricted MLE $l(\theta_0)$ estimated under the restriction $\theta_1 = 0$. H_0 will be rejected in favour of H_1, if LR is large, i.e. the unrestricted maximum $l(\theta)$ is significantly larger than $l(\theta_0)$.

The test statistic is asymptotically χ^2-distributed with degrees of freedom equal to the number of parameters tested. When the parameter estimates lie on the boundary of the parameter space a different asymptotic distribution apply (Ota et al., 2000). The degree of freedom is not trivial to estimate if parameters are correlated. In chapter 5 I will describe how to estimate the approximate number of degrees of freedom for penalised likelihood models.

The likelihood ratio test can only applied for nested hypothesis. The Akaike's Information Criteria (AIC) (Akaike, 1974) or the Bayesian Information Criteria (BIC) (Schwarz, 1978) are commonly used to compare models which are not nested :

$$AIC = -2l(\theta, x) + 2df, \tag{1.12}$$

$$BIC = -2l(\theta, x) + \log(n)df, \tag{1.13}$$

where $l(\theta, x)$ is the log-likelihood of the model, df is the number of degrees of freedom, i.e. usually the number of parameter of the model used, and n is the sample size which in our case is the length of the alignment. The optimal model chosen by BIC will include fewer parameters than the optimal model chosen by the AIC.

It is important to identify individual sites which have a strong impact on the inference of the estimates. Sites with a high leverage, sometimes called outliers, can influence the estimation, e.g. the topology can change and/or bootstrap confidence intervals for splits can be reduced dramatically. Site stripping can be used to remove the sites with the most changes. Another more sensitive approach is using an influence function to detect outliers (Bar-Hen et al., 2008) and remove them if necessary. This is closely related to a leave-one-out cross validation.

Non-parametric bootstrapping (Penny and Hendy, 1985, 1986; Felsenstein, 1985) is often used in phylogenetic analysis to evaluate the topological accuracy of estimates. Bootstrap samples are generated by sampling sites from an alignment X with replacement. For each bootstrap sample the parameters for the model are evaluated. We generate a distribution of the parameters from the bootstrap samples, which enables us to infer, for example, confidence intervals for each parameter.

1.4 Hadamard conjugation

Hadamard conjugation was developed by Michael Hendy and co-workers (Hendy, 1989; Hendy and Penny, 1993; Hendy et al., 1994; Waddell, 1995) and a nice overview was recently given in Hendy (2005). The Hadamard conjugation is a helpful tool to analyse relations between observed sequence patterns and edge-weights, but it is restricted to simple phylogenetic models (Jukes and Cantor, Kimura 2-state (Kimura, 1980) and Kimura 3-state model). We will use a special family of Hadamard matrices called Sylvester matrices which we can define recursively by:

$$H_0 = (1)\,, \quad H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_{n+1} = H_1 \otimes H_n = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}$$

The Sylvester matrices are symmetric and their inverses are given by

$$H_n^{-1} = \left(\frac{1}{2}\right)^n H_n \tag{1.14}$$

the crossproduct is

$$H_n^t H_n = 2^n 1_{2^n} \tag{1.15}$$

where 1_{2^n} is the identity matrix of the same dimension as H_n.

The Hadamard conjugation is defined

$$s = H^{-1} \exp(Hq) \tag{1.16}$$

and its inverse

$$q = H^{-1} \ln(Hs) \tag{1.17}$$

where assuming 4-state (nucleotide) characters s is a vector of length 4^n containing the probabilities of all 4^n patterns of nucleotide differences, $q \in R^{4^n}$ is a vector encoding the edge weights and model parameter of a tree T and we will refer to q as edge length spectrum and H is the Sylvester matrix H_{2n} of dimensions $4^n \times 4^n$ with its inverse $H^{-1} = 2^{-4n}H$.

We will use the same notation for labeling the edges or site patterns as in Hendy (2005). In the case of 2-state data, e.g. with states R (purines) and Y (pyrimidines) and if we have 4 sequences labeled A, B, C and D, s_{AB} is the probability that sequence A and B are the same and differ from C and D, (site pattern is either RRYY or YYRR). $s_\emptyset$ is the probability at the tips in all sequences is the same (site pattern YYYY or RRRR). Similarly we also label the edge weights q_{AB} the split $AB|CD$ and $q_\emptyset = -\sum_{\alpha \neq \emptyset} q_\alpha$ are the edge weights with the associated split.

1.4.1 Maximum Likelihood Estimation using the Hadamard conjugation

We can derive the (log-)likelihood of a tree T using the Hadamard conjugation:

$$L(q, T, \hat{s}) = \prod_\alpha (s_\alpha(T, q))^{\hat{s}_\alpha} \tag{1.18}$$

$$l(q, T, \hat{s}) = \sum_\alpha \hat{s}_\alpha \ln(s_\alpha(T, q)) \tag{1.19}$$

with the product and sum over all 4^n site patterns α, where $\hat{s}$ is the observed sequence spectrum and for each site pattern α, $s_\alpha(T, q)$ is derived from formula (1.16) using the edge weights from the tree T. This formula is identical to formula 1.5. We can derive

the first derivatives of the log-likelihood

$$\frac{\partial l(q,T,\hat{s})}{\partial q_\beta} = \sum_\alpha \hat{s}_\alpha \left(\frac{s_{\alpha\beta}}{s_\alpha} - 1 \right) \tag{1.20}$$

and the second derivatives

$$\frac{\partial l(q,T,\hat{s})}{\partial q_{\beta_1} \partial q_{\beta_2}} = \sum_\alpha \hat{s}_\alpha \left(\frac{s_{\alpha\beta_1\beta_2}}{s_\alpha} - \frac{s_{\alpha\beta_1} s_{\alpha\beta_2}}{s_\alpha^2} \right). \tag{1.21}$$

This allows us to estimate the MLE, using the NR as in equation (1.8).

1.4.2 Distance Hadamard

An extension of the Hadamard conjugation is the distance Hadamard (Hendy and Penny, 1993) which allows the estimation of an edge-weight spectrum from a distance matrix. The distance Hadamard uses the relation

$$d = -\frac{1}{2} Hq \quad \text{or equivalent} \quad -2d = Hq \tag{1.22}$$

where q is the edge spectrum and d is a vector of distances between two nodes. We can also express the edge spectrum through the distances

$$q = -2H^{-1}d \tag{1.23}$$

which we can expand using equations (1.14) and (1.15) to

$$q = (H^t H)^{-1} H^t (-2d) \tag{1.24}$$

which is the well known least-squares estimate.

Some values of d come directly from the distance matrix and the remaining values can be inferred with an algorithm given in Hendy and Penny (1993). Using spectra from different genes we can compare the genes even if the evolutionary history of the

genes is unknown or different. We will make use of this in chapter 4 and chapter 5.

1.5 Data sets

Yeast data set

Rokas et al. (2003) introduced a major genomic data set and many of the examples in the following sections will make use of this dataset. This data set is a compilation of 106 genes of eight species of yeast, seven species of the Saccharomyces family (*S. cerevisiae, S. paradoxus, S. mikatae, S. kudiavzelli, S. bayanus, S. castellii* and *S. kluyveri*) and *Candida albicans* as an out-group. This data set has a total of 127,026 characters for each taxon after gap sites have been removed. Not all gene trees have the same topology as the species tree (figure 1.4 a) and genes can vary strongly from each other in their rate of evolution (Bevan et al., 2007; Gatesy et al., 2007).

Phillips et al. (2004) showed that depending on the model and the method of tree estimation used either tree a) or tree b) of figure 1.4 can be favored for the concatenated dataset. Because of the high number of characters, the sampling error is low, so the bootstrap values are close to 100% for each tree.

Chloroplast data set

The second data set was provided by Ellen Nisbet (pers. comm.) and contains an alignment of 14 photosynthetic proteins from 19 anciently diverged species. In total the chloroplast dataset contains 4,933 amino acids. The data set contains homologous genes from eight cyanobacteria and eleven eukaryotes (four red algae and six green algae or plants, and a glaucophyte). Both primary and secondary origin chloroplasts are represented in the red and green plant taxa chosen, to present a range as wide as possible. The chloroplast is of primary origin if only one endosymbiosis event happened, i.e. a cyanobacteria was engulfed by an eukaryotic cell. It is of secondary origin if the the product of the first endosymbiosis itself gets engulfed by another eukaryote.

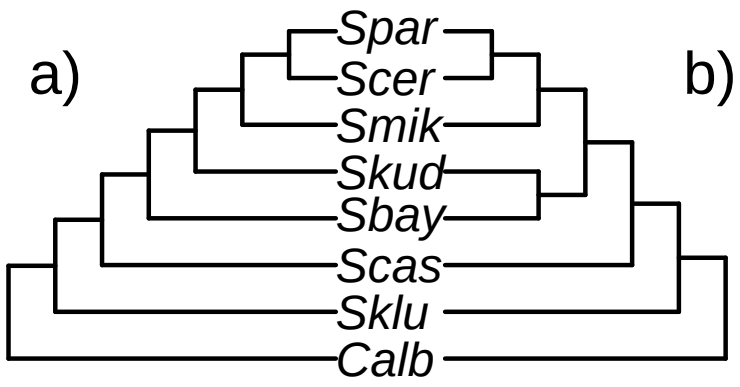


Figure 1.4: Figure a) shows the topology which is assumed to be the species tree for the eight taxa of the Rokas data set, but topology b) appears frequently among the gene trees. Trees a) and b) can both be the preferred topology depending on the evolutionary model.

A total of about 50 genes were conserved in the genomes of oxygen evolving photosynthetic organisms. From seven different functional complexes of these genes the two largest proteins were always chosen for the analysis. The proteins within each complex have obligate interactions with each other or with a common protein.

Chapter 2

Mixture models

In this chapter we will discuss mixture models or, more exactly, finite mixture models for phylogenetic tree estimation. Mixture models are used in two areas in phylogenetics. The first is to model rate heterogeneity when estimating phylogenetic trees. Modelling site heterogeneity with the discrete gamma distribution and the ratio of invariant sites model ($\Gamma + I$) can be expressed as a mixture model.

The second application of mixture models is to add more flexibility to simulation studies. Here the mixtures are often generated from a set of at least two trees, which vary in their edge lengths or their topology. This approach is used to model more complicated evolutionary scenarios such as covarion evolution (Kolaczkowski and Thornton, 2004) or lateral gene transfer.

I will give a small example of how to visualise mixture models which is inspired by Lindsay (1995). Assume we have data (in our case an alignment with k partitions), which were generated by k different distributions (evolutionary processes). Now let us further assume that the student who entered the data lost the information that told us which distribution each site was sampled from. This is now the situation we assume using a mixture model. In many cases we can still infer the k distributions and their associated mixing weights, which are proportional to the length of alignments. Often we need also to determine the number of distributions.

I will first give some more formal background on mixture models as they are both

used in this chapter and chapter 3.

2.1 Background

2.1.1 General theory of mixture models

We assume that the data are generated from a process which is a mixture of k distributions. Often we are not explicitly interested in the different distributions, but we use this framework to model heterogeneity of some parameters. We can compute the likelihood given data x and a tree T, if we integrate over all k distributions and N sites

$$l(x|\theta, T, \varphi) = -\sum_{i=1}^{N} \log \left(\sum_{j=1}^{k} p_j f(x_i|\theta_j, T, \varphi) \right) \tag{2.1}$$

where p_j, $\sum_{j=1}^{k} p_j = 1, \quad p_j \geq 0 \; \forall j$ are the mixing weights and $f(x_i|\theta_j, T, \varphi)$ is the probability of observing the character at site i given the tree T and parameters θ_j and φ. θ_j are the parameter estimates specific to distribution j and φ are additional parameters which are fixed over all mixture distributions.

We can generalise equation (2.1) to allow a different tree $T_j, j = 1 \ldots, k$ associated with each mixture, i.e. a different set of edge lengths to model covarion evolution, or a different tree topology (with a set edge weights) to model for example the influence of different gene trees.

$$l(x|\theta, T_1, \ldots, T_k, \varphi) = -\sum_{i=1}^{N} \log \left(\sum_{j=1}^{k} p_j f(x_i|\theta_j, T_j, \varphi) \right) \tag{2.2}$$

We can estimate the mixing weights p_j using the relations

$$p(j|x_i) = \frac{p_j f(x_i|\theta_j, T_j, \varphi)}{\sum_{l=1}^{k} p_l f(x_i|\theta_l, T_j, \varphi)} \tag{2.3}$$

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} p(j|x_i) \tag{2.4}$$

where $p(c|x_i)$ is the posterior distribution (equation 2.3) that site i belongs to class c. We can interpret the weights p_j as the mean of the posterior distributions.

With the equations given above it is straightforward to adapt an iterative algorithm to estimate phylogenetic (finite) mixture models.

Algorithm 2.1 Mixture model algorithm.

---

1. `Choose starting values` $p_j^{(i)}, \theta_j^{(i)}, \varphi^{(i)}, j = 1, \ldots, k, i = 0$.
2. `for i=0,1,2,...`
3. `estimate` $p(j|x_n)^{(i+1)}$ `for` n=$1, \ldots, N$, j=$1, \ldots, k$ `using equation (2.3)`

$$p(j|x_n)^{(i+1)} = \frac{p_j f(x_n|\theta_j^{(i)}, T_j^{(i)}, \varphi^{(i)})}{\sum_{l=1}^{k} p_l f(x_n|\hat{\theta}_l^{(i)}, T_j^{(i)}, \varphi^{(i)})}$$

4. `Compute` $p_j^{(i+1)}$ `(equation 2.4)`

$$p_j^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} p(j|x_n)^{(i+1)}$$

5. `Optimise` $\theta_j^{(i+1)}, \varphi^{(i+1)}$
6. `Go to step 2, as long as changes in` $p_j^{(i+1)}, \theta_j^{(i+1)}$ `or` $\varphi^{(i+1)}$ `are over a given threshold`

Dempster et al. (1977) showed that the algorithm 2.1 can be seen as a version of the EM-algorithm. The convergence of the EM-algorithm, and so the algorithm above, can be slow, especially when two or more parameters are correlated (Fahrmeir and Tutz, 1997, p. 356). As we will see later in section 2.1.4 parameters are often correlated.

I implemented these mixture models in R (Ihaka and Gentleman, 1996; R Development Core Team, 2007) using the EM-algorithm and compared the results when possible with other implementations, mainly BayesPhylogenies (Pagel and Meade, 2004). Just recently another program has been made available to estimate mixture models Kolaczkowski and Thornton (2008).

2.1.2 Identifiability of mixture models

Under some circumstances mixture models can be non-identifiable (Titterington et al., 1985). At the moment it is an active field of research to establish conditions under which phylogenetic mixtures are identifiable.

Definition 2.1.1 (Identifiability)**.** *Let* $\mathbf{H}$ *be a class of distribution functions*

$$\mathbf{H} = \{H(x) : H(x) = \sum_{j=1}^{k} p_j F(x|\theta_j, T, \varphi), \quad p_j \geq 0, \sum_{j=1}^{k} p_j = 1, \\ x \in R^d, k = 1, 2, \dots \}$$

Suppose H_1, H_2 *are both in* $\mathbf{H}$ *and*

$$H_1(x) = \sum_{j=1}^{k} p_j F(x|\theta_j, T, \varphi), H_2(x) = \sum_{j=1}^{k'} p'_j F'(x|\theta'_j, T', \varphi')$$

then $H_1 \equiv H_2$ *if and only if* $k = k'$ *and for some permutation to match the corresponding elements* $p_j = p'_j$, $F(x|\theta_j, T, \varphi) = F'(x|\theta'_j, T', \varphi')$ *then* $\mathbf{H}$ *is* identifiable (Titterington et al., 1985, p. 36)

Matsen and Steel (2007) give examples of mixture models of trees which are not identifiable. They give a mixture of two trees which share the same topology and have the same expected site frequencies as a tree with a different topology. Furthermore Stefankovic and Vigoda (2007b,a) gave further examples of non-identifiable mixtures. They also showed that examples similar to those of Kolaczkowski and Thornton (2004) can be non-identifiable for some parameter settings.

There are also many cases where identifiability is established in phylogenetics. Allman and Rhodes (2006a, 2007) and Allman et al. (2008) proved that many mixture models used in the estimation of rate heterogeneity, including the often used ($GTR+\Gamma$), the ($GTR + I$) and the covarion model (Tuffley and Steel, 1998), are generically identifiable, i.e. that the set of non-identifiable distributions has probability zero. However the ($GTR + \Gamma + I$) is so far not (yet) established to be identifiable. Pagel and Meade

(2004, 2005) used mixtures of rate matrices, which are shown to be identifiable if the number of mixtures is lower than the number of character states (Allman and Rhodes, 2006a).

One of the differences between the models for which identifiability is established and those where non-identifiability occurs is in the number of parameters. Mixture models for rate heterogeneity add only few additional parameters, whereas for mixtures of trees the number of parameters is at least doubled.

2.1.3 Mixtures to model rate heterogeneity

In phylogenetics mixture models are frequently used to model rate heterogeneity such as to estimate the proportion of invariant sites or parameters of the discrete Γ-model.

Having a proportion of invariant sites is an example of a mixture model. Let us assume that the lengths of each edge of a tree T represents the expected number of substitutions across that edge. Then we have an additional constraint

$$\sum_{j=1}^{k} p_j \omega_j = 1 \tag{2.5}$$

where p_j are the mixing weights and ω_j are the rates. Discrete gamma (Yang, 1994, 2006) and discrete log-normal distribution for rate variation are modifications of mixture models. The expression of equation (2.1) stays valid, but the parameters $p(j)$ and θ_j are derived differently from equations (2.3) and (2.4) to follow the specified distribution. For the discrete gamma model the distribution of rates is divided into k quantiles approximating the gamma distribution. The mixing weights are $p_j = 1/k, j = 1, \ldots, k$ and the ω_j are determined as the mean rate of the respective quantile (see figure 2.1). The advantage of using a parametric distribution is that only one parameter for the distribution determines all the weights p_j and ω_j. However the p_j and ω_j given by a discrete gamma function will differ from the estimates given in equations (2.3) and (2.4), when the rates cannot be approximated by a gamma distribution.

Mixture models to estimate other parameters are less common. Lartillot and Philippe

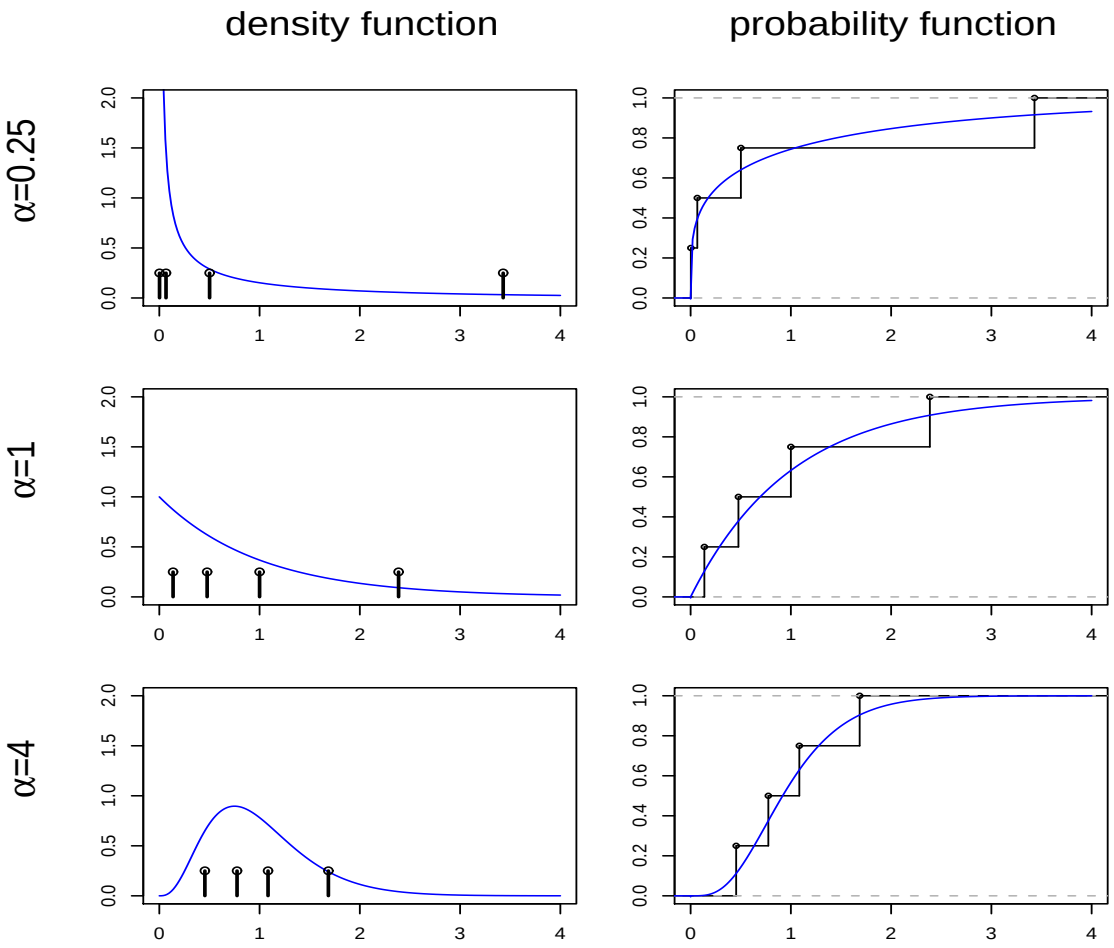


Figure 2.1: Density and distribution functions of the gamma and discrete-gamma functions with shape parameters $\alpha = 0.25$, 1 and 4. The p_k represent the jump heights of the cumulative distribution function, here all are fixed to $1/k = 0.25$. The jumps occur at the mean of the four quantiles of the discrete gamma distribution.

(2004) introduced a mixture model to allow variation in the base frequencies. Pagel and Meade (2004, 2005) modeled rate variation with a mixture of rate matrices in a Bayesian framework. Furthermore Pagel (pers. comm.) allowed tree topologies and edge lengths to vary.

2.1.4 Mixtures of sets of edge lengths and topologies

When estimating mixture models for mixtures from the same tree topology with different edge lengths or from different topologies we introduce a large number of parameters. Steel (2005) raised the concern that over-parameterization may produce a good fit in terms of the likelihood values, but increase the variance of the estimates. More alarming are the results of Matsen and Steel (2007), who have shown that mixtures of the same tree topology with different edge lengths can exactly mimic a different topology. This needs further research to identify under which conditions mixture models are identifiable. Stefankovic and Vigoda (2007a,b) and Mossel and Vigoda (2005) give examples for non-identifiability of phylogenetic mixture models. However, it is often assumed that biological data have been generated as a result of different processes (Kolaczkowski and Thornton, 2004).

2.1.5 Detecting partitions

Often we assume that mixtures were generated by k different evolutionary processes and it is very likely that sites, which are neighbors, share the same process. We can estimate for each site which mixture most probably generated it and this allows us to detect partitions. Two decision rules are commonly used (Fahrmeir and Tutz, 1996, p. 361). The Bayesian decision rule classifies a site a site x_i to the class g with the highest posterior probability (equation 2.3):

$$p(g|x_i) \geq p(j|x_i), \quad j = 1, \ldots, k \tag{2.6}$$

The maximum likelihood rule classifies a site x_i to the class g with the highest likelihood

$$f(x_i|g) \geq f(x_i|j), \quad j = 1, \ldots, k \tag{2.7}$$

Both decision rules agree for a uniform prior, that is if the mixing weights (equation 2.4) are identical ($p_1 = \cdots = p_k$).

2.2 Methods

We now explore some of the statistical difficulties that arise when applying mixture model methodology in phylogenetic analysis.

Estimation of mixture models is not possible in many of the standard phylogenetic software. An exception is BayesPhylogenies (Pagel and Meade, 2004) and some examples in this chapter use this package. For comparison with maximum likelihood analyses I have implemented algorithm 2.1 in the function pmlMix as part of the `phangorn` package, which allows greater flexibility to optimise parameters (e.g. edge weights, base frequencies) separately for each mixture. In appendix A.1 a small code example is given for generating mixture models using the pmlMix function. All the models are computed using a Jukes-Cantor model, but results are similar if more complicated models like GTR are used (examples not shown).

The Bayesian MCMC inference was performed with BayesPhylogenies (Pagel and Meade, 2004). A Markov chain was run for 11,000,000 iterations from which every 10,000th tree was sampled and the first 1,000,000 iterations were discarded as burn-in. The autocorrelation time was around 25,000 for the run. In all the inferences, Bayesian and maximum likelihood, only the edge weights were optimised for each mixture, all other parameter were kept constant.

First we generated a dataset consisting of 10,000 sites from a 60:40 mixture of the trees T1 and T2 in figure 2.3 a). This dataset is generated from the expected frequencies of site patterns using a Jukes-Cantor model. This dataset allows to test whether a algorithm converges towards the optimum. A variant of function pmlMix is

used to optimise the edge weights of the trees in the mixture for fixed mixing ratios as shown in figures 2.2 and 2.3. This variant jumps over step 4 in algorithm 2.1 which computes the estimate of $\hat{p}_j$ in each iteration.

From the data with the expected site frequencies 100 non-parametric bootstrap samples with each 10,000 were simulated. To allow for the bootstrap sampling the expected site frequencies were rounded to the next integer. The estimates of the bootstrap samples were used to infer the variances of mixing weights (figure 2.4).

2.3 Results

2.3.1 Mixture of two trees

We now consider data generated on a mixture of two trees. The aim is to discover if it is possible to recover the edge lengths and the mixing weights of the two trees. In figure 2.3 I illustrate mixtures estimated from an alignment based on the expected site pattern frequencies for the trees in figure 2.3 a). I fixed the proportion of the mixture at a ratio of 60:40. The data are not simulated, the site patterns appear in the frequencies as we expect from data with infinite sequence length. The likelihood values given in the following study correspond to a sequence length of 10,000 sites. This allows us to exactly recover the mixture. We introduced no sampling error, so for this example the model is identifiable. When we estimate the edge length only for the different tree topologies, but not a mixture of trees, the (log-)likelihood values are significantly lower than for the optimal mixture model (figure 2.3 b, c). If we estimate a mixture of trees, but mis-specify the mixing weights, the (log-)likelihood of the estimates are often very close to the optimal mixture (figure 2.3 d, e). This caught my attention, as the the EM-algorithm I used to estimate mixtures converged very slowly and terminated before reaching the optimal model. Figure 2.2 shows the (log-)likelihood function depending on the mixing weights.

The likelihood function is very flat for a wide range of mixing weights and drops steeply in the area where just one of the two trees is strongly supported. The edge weights of corresponding edges of trees in the mixture are highly correlated. When holding the mixing weights fixed, the optimal edge weights of the trees can vary considerably from the edges of the trees used to generate the data, even though the difference in the likelihood is low (compare the edge weights of the trees in panel a) with those in d) and e) of figure 2.3).

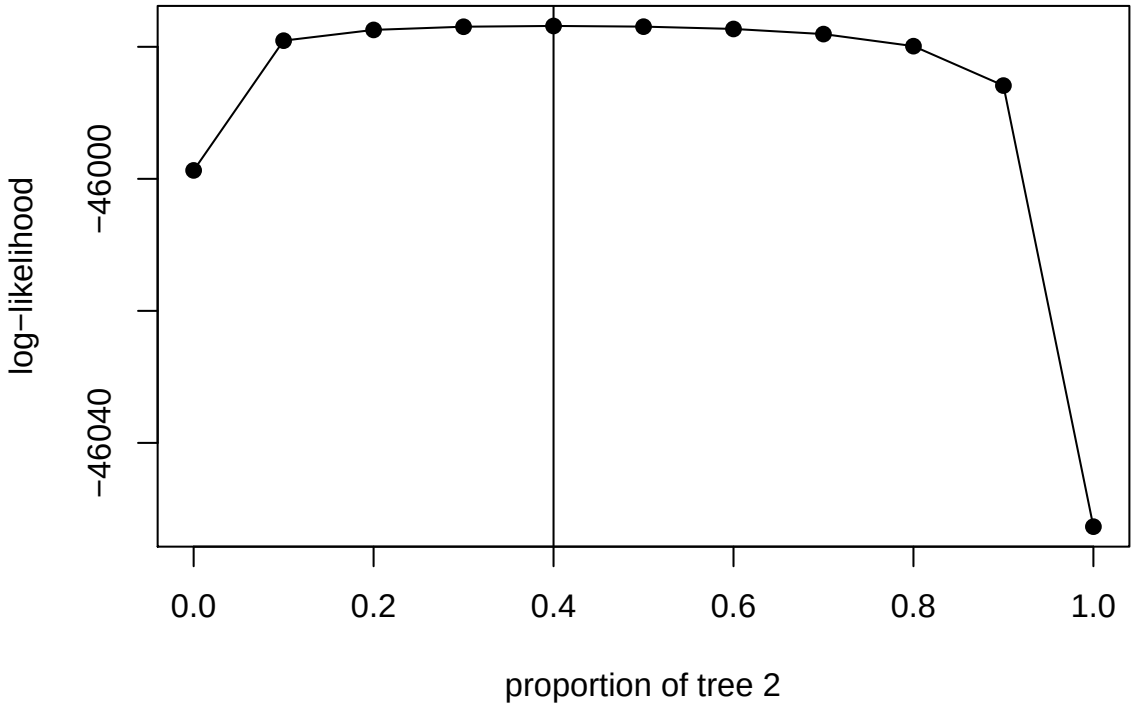
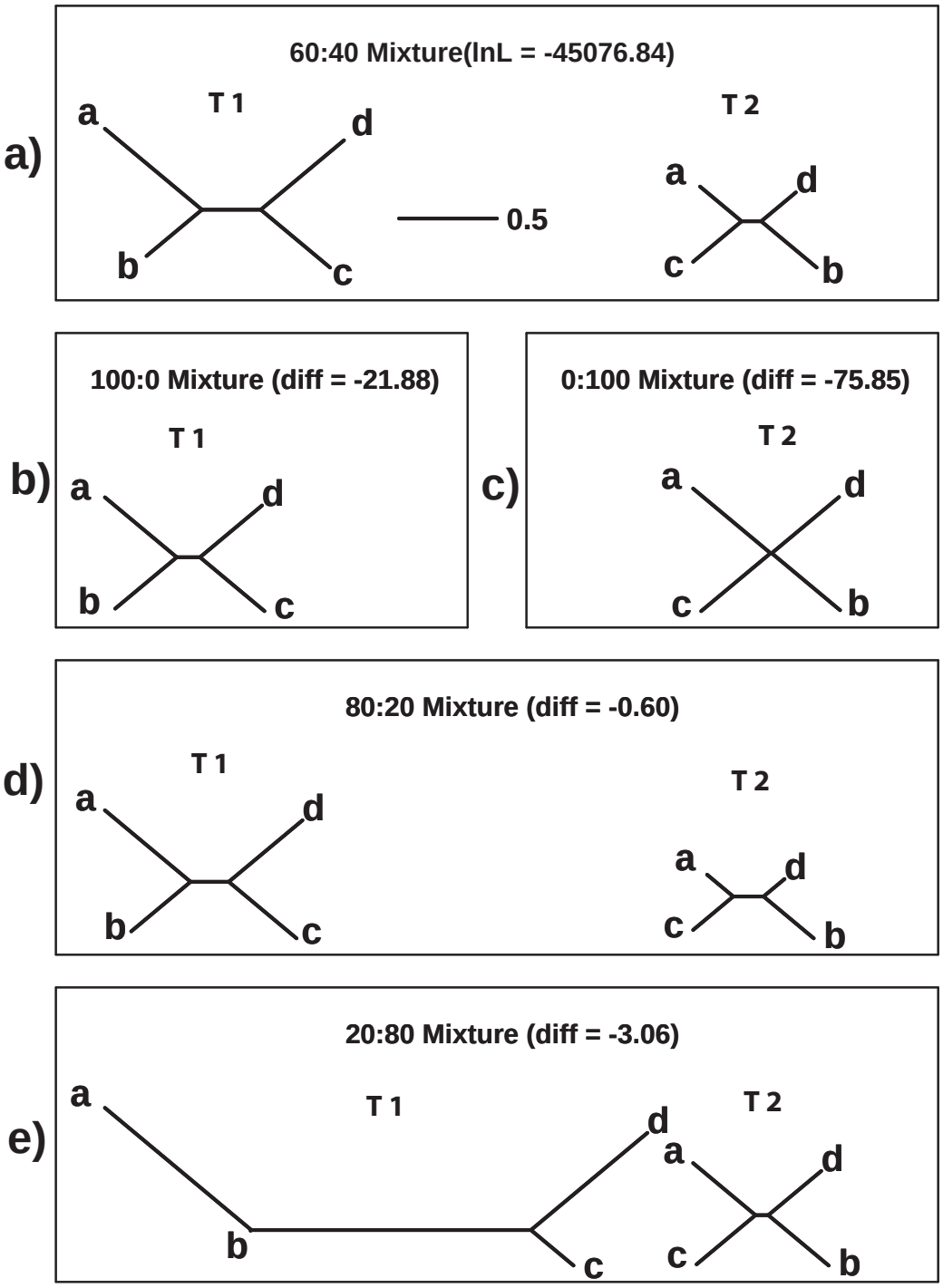


Figure 2.2: Likelihood function dependent on the ratio of the mixtures of example 2.1. Each point represents the likelihood for the specific mixture. The log-likelihood function is flat over a wide range of mixing weights, making it hard to determine the optimal value numerically.

Figure 2.3: (See next page) Mixture model based on data generated on the alignment of a 60:40 mixture of the trees in a). All edge lengths are drawn to scale. The log-likelihood values in b)-e), which are slightly lower, are given as the differences to the likelihood of fitting to the mixture of the generating topologies a). The data represent the exact expected site frequencies. b) and c) show the estimated edge lengths when restricted to each of the two tree topologies. d) and e) show the edges lengths for the two trees if the mixing ratios are specified as 80:20 and 20:80. Mixture models achieve likelihood values close to the optimal value even if the weights of the mixtures is misspecified and they are much closer to the best values than estimates based on single trees. The edge length estimates can differ considerably, depending on the choice of the mixing parameter.


a)
60:40 Mixture(lnL = -45076.84)
T 1
a
d
b
c
0.5
T 2
a
d
c
b
b)
100:0 Mixture (diff = -21.88)
T 1
a
d
b
c
c)
0:100 Mixture (diff = -75.85)
T 2
a
d
c
b
d)
80:20 Mixture (diff = -0.60)
T 1
a
d
b
c
T 2
a
d
c
b
e)
20:80 Mixture (diff = -3.06)
T 1
a
b
d
c
T 2
a
d
c
b

The alignment we used represented the exact site pattern frequencies of the mixture. Next we bootstrapped the data to introduce stochastic variation into the alignments. 100 bootstrapped samples were generated, each of 10,000 sites and we performed a non-parametric bootstrap. When estimating the mixtures for the 100 bootstrap samples we observed that the estimates for the ratios of the two topologies in the mixture have a high variance (figure 2.4). These findings are consistent with the flat likelihood function for the mixture described above. Even if the mixing weights are correctly identified, we find that edge length estimates are often biased.

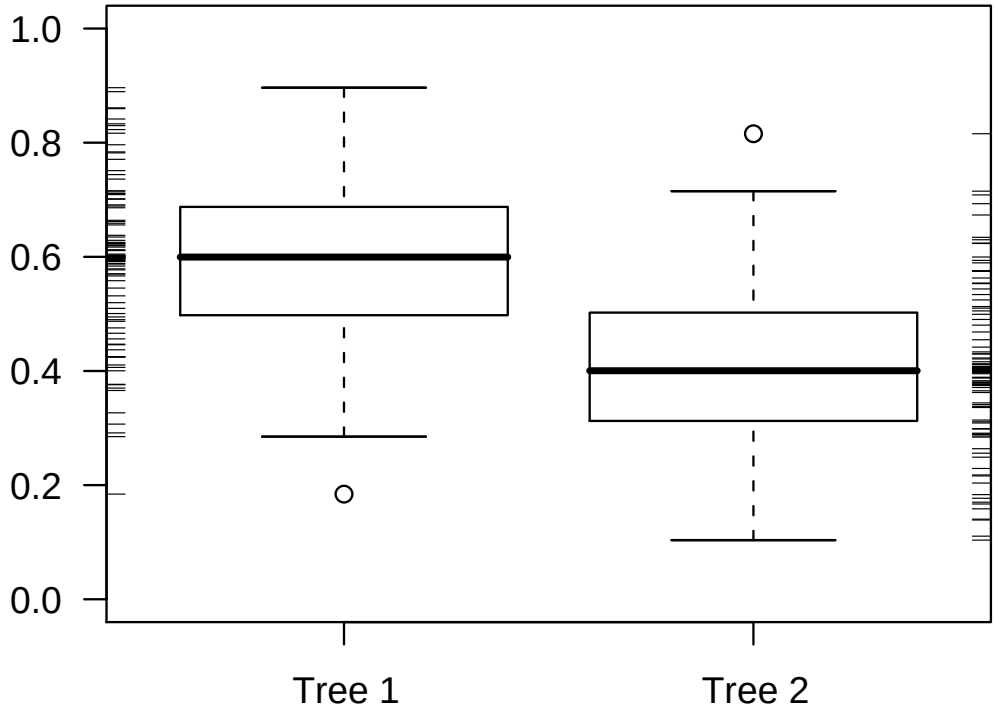


Figure 2.4: Estimates of the likelihood ratios for the two trees of figure 2.3 a) when restricted to a single tree from the mixture for 100 bootstrap samples generated from a 60:40 mixture of the two trees.

The results in figure 2.4 are surprising and make it difficult to recover the trees in the mixture. A reason behind this may be that we have a high number of parameters in our

model, which can effect the estimates. Putting constraints on some of the parameters, e.g. forcing the trees to be rooted could be one way to reduce these problems. It is obvious how well the mixture model can adapt, if we look at the covariance matrix of the edge parameters (figure 2.5). The correlation between parameters which correspond to the same edge on the different trees is almost always close to -1, whereas most of the other correlations are considerably higher.

We also used the program BayesPhylogenies (Pagel and Meade, 2004) to estimate the mixture (see figure 2.6), and we observed very similar results to the ML procedure implemented in R. The posterior variance of the Bayes estimate for the mixing weights and edge length is low in comparison to the sampling variance from the bootstrap samples above (figure 2.4). Prior information that influence the edge length will also reduce the variance of the mixing weights and lead to this effect. In addition the mixing of the Markov chain in BayesPhylogenies can be slow since many of the edge parameters are correlated (Mossel and Vigoda, 2005, 2006).

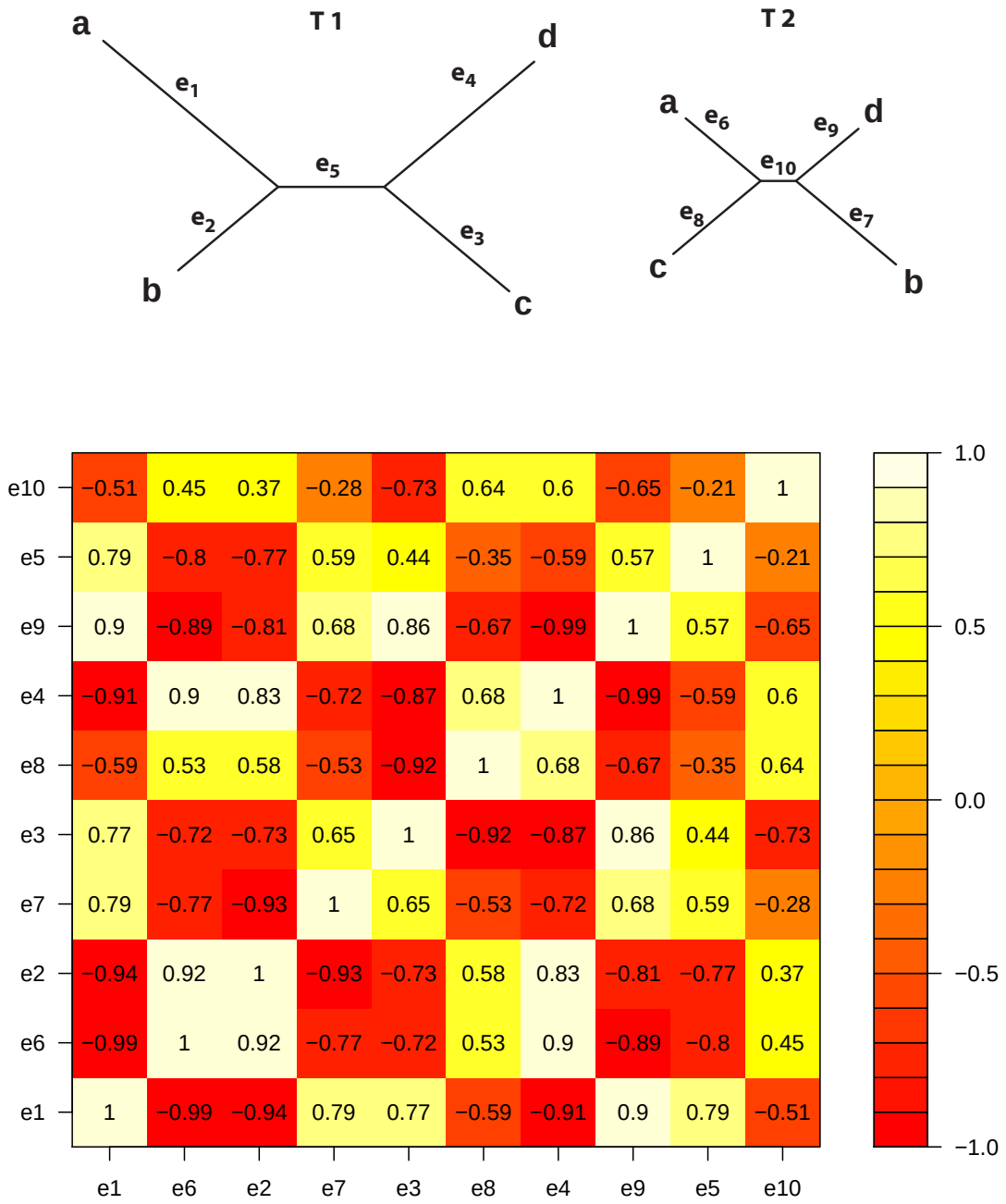


Figure 2.5: Correlation and covariance for the mixture model from example 2.1. The correlations are estimated from relation to the expected Fisher information of equation (1.10). On the diagonal all the correlations are 1. The parameters are ordered so that edges showing the same partition in the two trees of the mixtures are next to each other. The correlation between bipartitions shared in both trees are all strongly negatively correlated (close to -1).

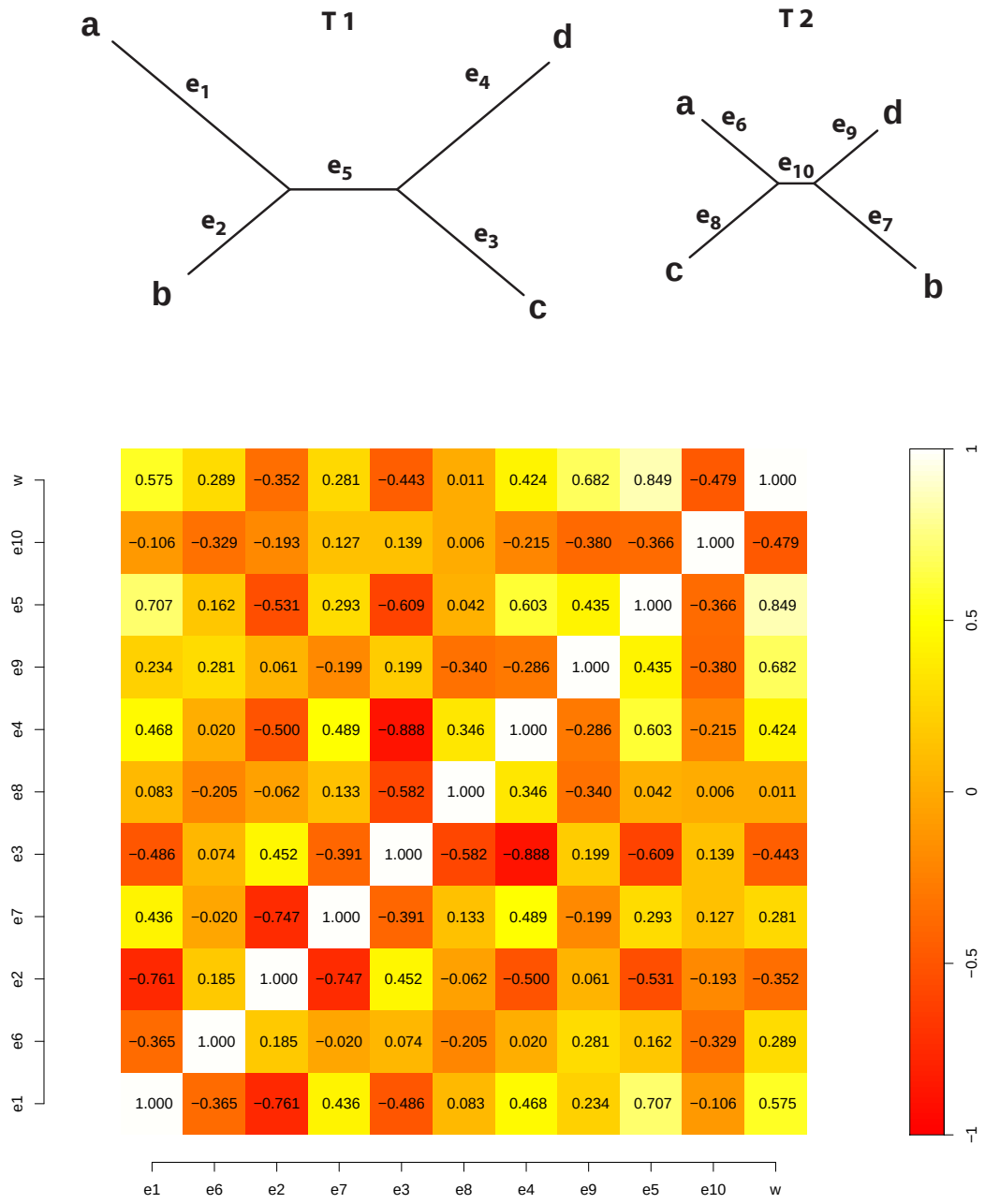


Figure 2.6: Correlation and covariance for the mixture model from example 2.1. The correlations for the edge length and mixing weight w estimated from 1000 sampled trees using BayesPhylogenies. The parameters are ordered so that edges showing the same partition in the two trees of the mixtures are next to each other. The correlation between bipartitions shared in both trees are all strongly negatively correlated, even not as strange as in figure 2.5.

Using the same dataset we can use the posterior distribution (see figure 2.7) of the mixture model to identify partitions. These partitions then may deliver a basis for further separate analysis of the partitions. Partition models are explored in more depth in chapter 4.

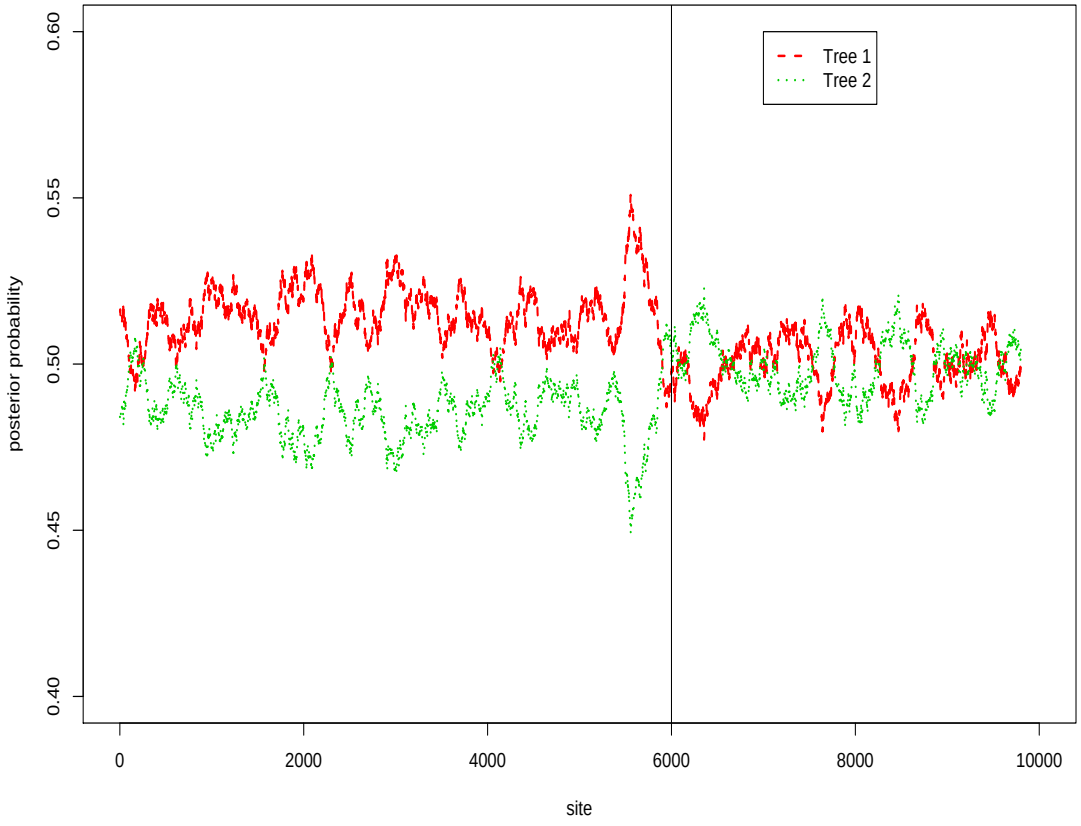


Figure 2.7: Posterior probability for tree 1) and tree 2). We used a running mean to smooth the trace of the probabilities. We observe a different pattern of the traces after the change of the partitions close to position 6000. So in this simulation the posterior probabilities can be used to identify the partitions of the alignment.

2.3.2 Model misspecification of mixture models

Here we simulate a dataset on a single tree, but estimate a mixture of two trees on that. The use of additional parameters (here the use of an additional tree in the mixture) will always increase the (log-)likelihood, even if the additional parameters are not statistically significant. The additional tree in the mixture (with a topology which

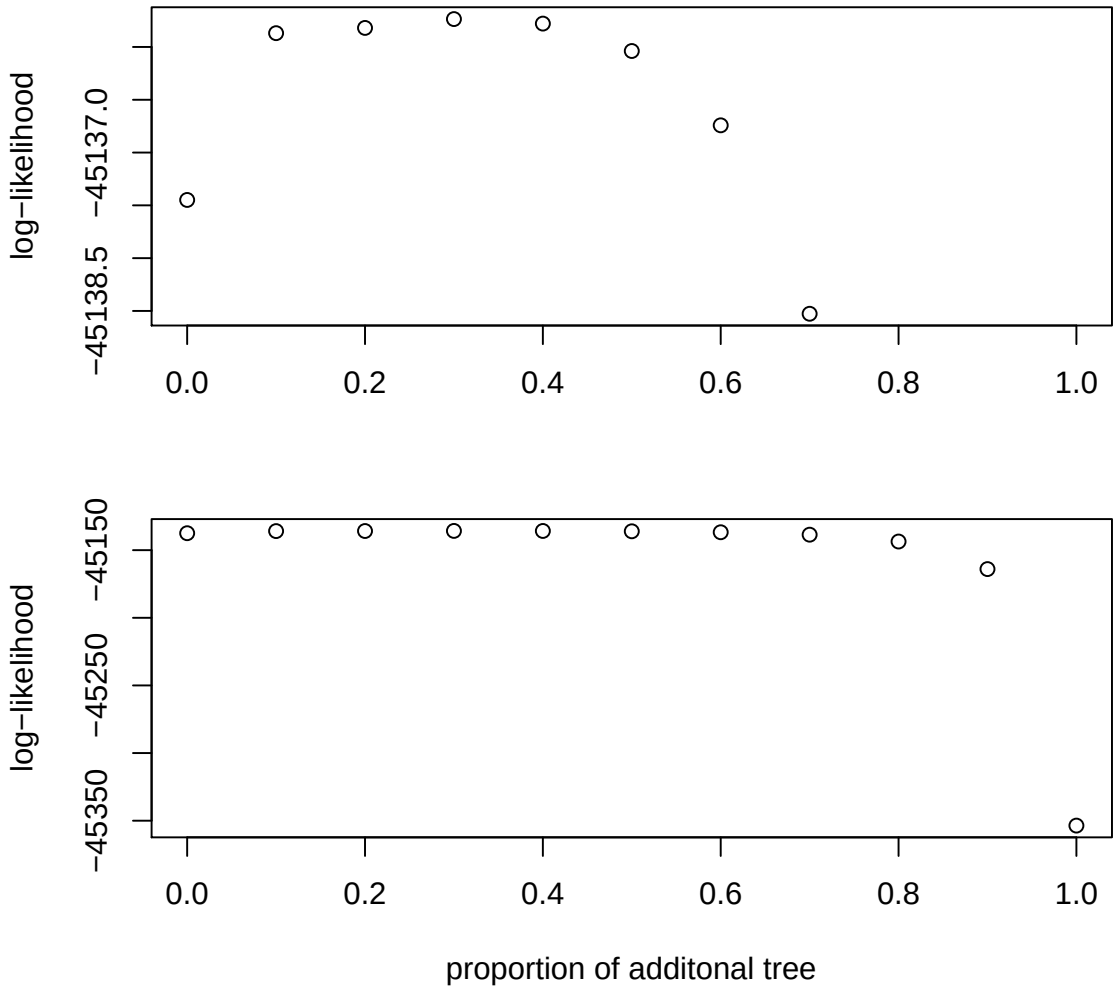


Figure 2.8: Likelihood surface dependent on the ratio of the mixtures. The (log-)likelihood function increases for mixtures even though here the likelihood difference is not significant.

did not generate the data), often has one or more edges that are very small. However the additional parameters are not significant and we would reject the mixture model on the basis of the AIC or BIC in favour of the true model. Pagel and Meade (2008) uses

a reversible-jump MCMC algorithm to avoid too many parameters joining the model.

2.4 Conclusion

In this chapter I presented an example for mixtures of trees. The convergence of the estimates of mixture models using the EM-algorithm is slow, something we expect from the EM-algorithm when many of the parameters are correlated. The likelihood function or the likelihood surface can be very flat if the mixture contains many correlated parameters (see figure 2.2). Simulation studies based on mixture models should take into account that estimates for edge weights and mixing weights can have a high variance and can be highly correlated.

The posterior variance of the Bayes estimates is low in comparison with the sampling variance of maximum likelihood estimates or bootstrap samples. Prior information on the edge length reduces the variance of the mixing weights. This offers the opportunity to reduce the variance of the mixing proportions using appropriate *a priori* information. In contrast the danger is that mixing proportions depend strongly on the prior and do not reflect the data. In chapter 6 I will present penalized likelihood models, which introduce restrictions on the variability of the edge weights. This can be used to reduce the variance also for maximum likelihood estimates, but will introduce some bias to the estimates.

In the examples here the mixture models we used were capable of correctly identifying, if the data were not generated not from a mixture.

In the next chapter we will use mixtures that lead to multiple optima if edge weights are estimated on a single tree.

Chapter 3

Multiple Optima

In applying ML in phylogenetics there are two levels of optimisation:

1. The local problem is, given a tree T, find the set of parameters on T (including edge weights) which maximises the likelihood of observing the data.

2. The global problem is to find the tree T which gives the maximum likelihood across all trees.

We would like our estimate to be a unique global maximum on both levels. I address only the first question, whether there are multiple optima on a single tree. Unfortunately the local optimisation problem cannot be solved exactly, iterative numerical estimations are found. In this chapter I will show that multiple optima can arise from simple mixtures on the tree with the highest likelihood. This problem differs from proofs of the consistency or identifiability of phylogenetic reconstruction methods (see for example Chang (1996); Allman and Rhodes (2006a); Matsen and Steel (2007); Allman et al. (2008)) as these assume perfect data, i.e. in this case infinite sequence length and generated on a single tree.

3.1 Background

3.1.1 Multiple optima in general functions

We first describe how we can identify local and global maxima. A (log-)likelihood function $l(\hat{\theta}, x)$ attains a (local) maximum at $\hat{\theta}$, if:

1. the score vector $sc(\hat{\theta}) = \frac{\partial l(\hat{\theta},x)}{\partial \theta'} = 0$ and

2. $\mathcal{H}(\hat{\theta}) = \frac{\partial^2 l(\hat{\theta},x)}{\partial \theta \partial \theta}$ is negative definite.

To ensure that there exists only one (global) maximum one could further demand that the likelihood function is strictly concave, for example $\mathcal{H}(\theta)$ is negative definite for every $\theta \in \Theta$. In this case, hill-climbing algorithms will be guaranteed to eventually find the maximum of the likelihood surface. However, it is often unknown or not fulfilled, that the likelihood function is strictly concave.

Fukami and Tateno (1989) and Renée and Tillier (1994) stated that $L(\theta)$ has a unique maximum. Fukami and Tateno (1989) came to this conclusion when looking at the Hessian independently for each parameter, when estimating the likelihood with the pulley-principle of Felsenstein (1981). Unfortunately this generalisation to the multivariate case does not hold (Tarone and Gruenhage, 1975; Makelainen et al., 1981).

When the likelihood surface is not strictly concave, many local maxima can exist. In this case hill climbing algorithms are only guaranteed to find local optima.

There do exist functions which have an infinite number of discrete (global) maxima and no other critical points. Tarone and Gruenhage (1975) gives the function

$$g(x, y) = -e^{-2y} - e^{-y} \sin(x) \tag{3.1}$$

as an example, with global maxima at $y = \ln(2)$ and $x = \frac{(4k-1)\pi}{2}$, $\forall k \in \mathbb{Z}$ (see figure 3.1). However little is known about how complex the likelihood surface can be for the phylogenetic setting. In the next section we discuss what is known.

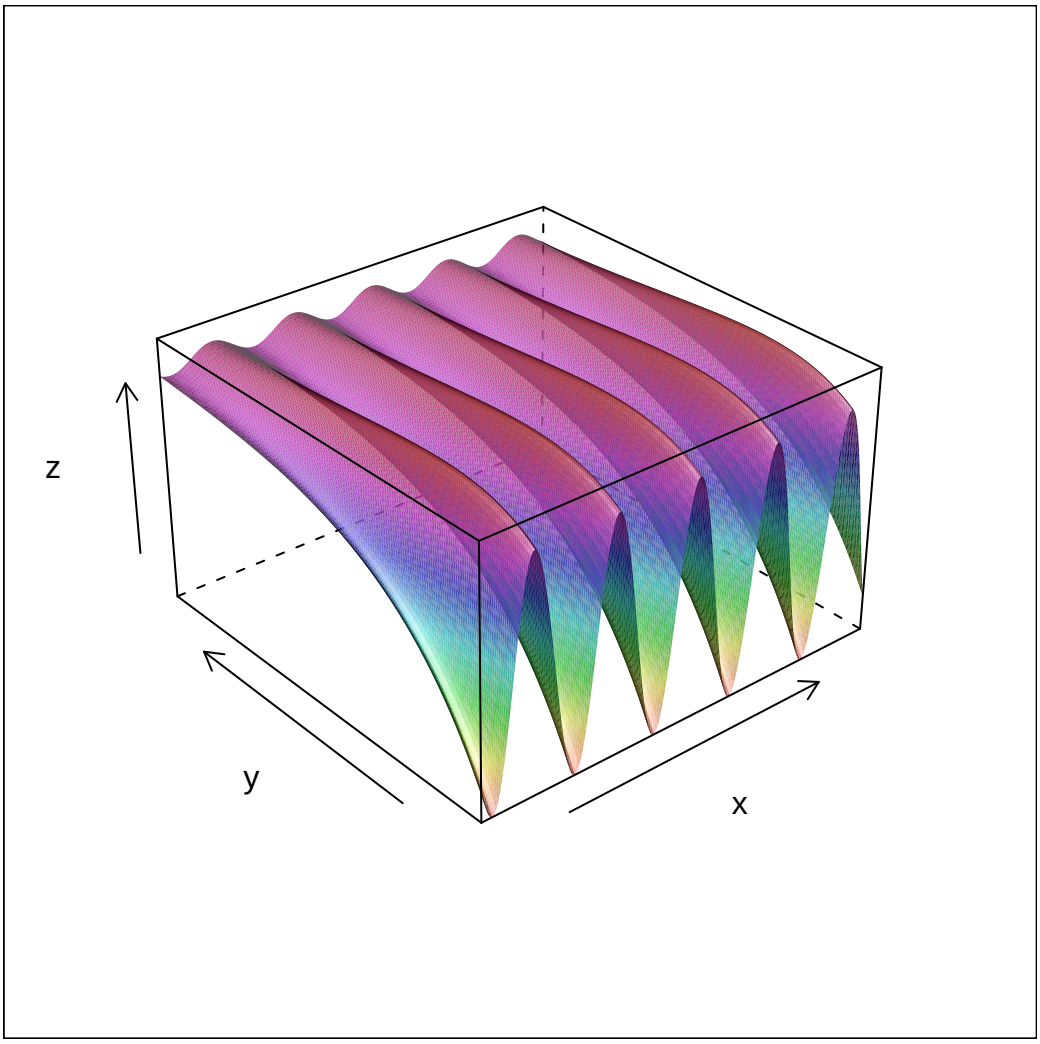


Figure 3.1: Example of a function with an infinite number of (separate) maxima and no other critical points. This is a worst case scenario for finding maxima with hill climbing algorithms.

3.1.2 Multiple optima on four taxon trees

We will now discuss some circumstances under which multiple optima can occur. I first discuss earlier work and extend this work to to give some conditions when, under a simple Cavender-Farris model (Cavender, 1978), multiple optima sometimes occur.

Steel (1994) found a simple phylogenetic counter-example to Fukami and Tateno (1989) and Renée and Tillier (1994) with two different maxima on a four taxon tree. However, a limiting feature with this counter example is that the parameter estimates lie on the boundary of the parameter space, where some edge weights are estimated to be zero or infinity. The asymptotic properties from chapter 1.3 would not apply for this case, and it is controversial if in such a situation the MLE really exists, but this example showed that there exist situations where the likelihood function is not convex.

Figure 3.2 illustrates some different cases in which multiple optima can arise. The true mechanism (parameter) by which the data are generated is normally unknown, but we try to find the best predictions in the model space to minimize the distance to the realisation of the given data. In A) there exists a single optimum in the model space. In B) we see two local optima in the model space, indicated by the two black dots. C) shows a ridge of solutions.

Rogers and Swofford (1999) conducted a simulation which generated data on a single tree T with 4 and 6 taxa. They found that trees with the topology of the generating tree T rarely have multiple optima, whereas other topologies can have multiple optima more easily. Furthermore they found that multiple optima occur more frequently when the internal edges are long in comparison to the external edges. They concluded that multiple optima are not likely to be a major concern for ML phylogenetic reconstruction in general, though of course their data was simulated on a single tree and by a single mechanism.

Chor et al. (2000) found additional counter examples in the interior of the parameter space (not just at the extremes). They also found examples where multiple optima exist on all possible tree topologies, which contrasts with the results of Rogers and Swofford (1999). In some of these examples the parameter estimates are linearly dependent, in

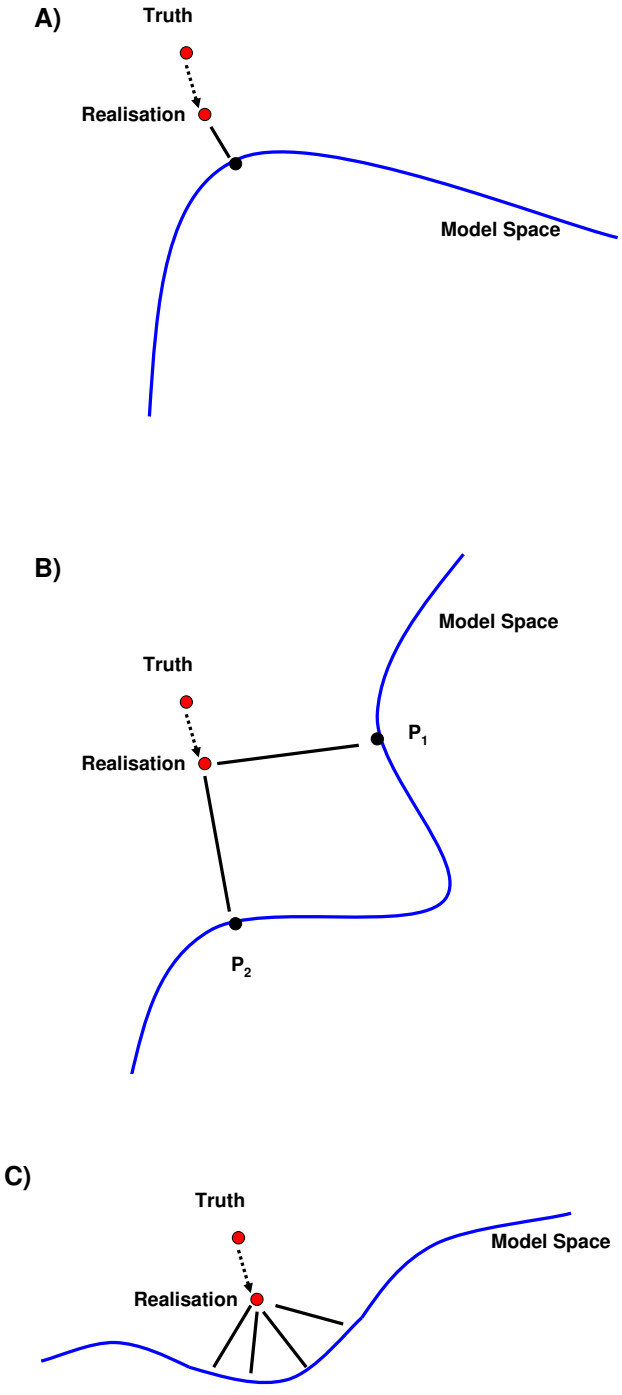


Figure 3.2: Schematic presentation of the different possibilities for maximum likelihood optima where the distance from the observation to Model space represents the likelihood. In A) there is a single optimum, B) two distinct optima and in C) an interval of equilikely model points. The model space is the set of all possible predictions from the model.

statistics this problem is called multicollinearity.

A difference between these two studies is how the data (alignments or site patterns) are generated. Whereas Rogers and Swofford (1999) simulated their alignments from an evolutionary (tree-like) model on a single tree, Chor et al. (2000) found data which cause multiple optima but without describing a mechanism to generate that data.

In constructing their examples, Chor et al. (2000) used Hadamard conjugation to express the likelihood as an analytic function of the edge length parameters. Hadamard conjugation relates vectors called spectra (Hendy, 1989; Hendy and Penny, 1993) whose entries are indexed by splits of the set X of taxa.

For the simple 2-state symmetric model of Neyman (1971) on a tree T on the 4 taxon set $X = \{1, 2, 3, 4\}$, a spectrum is $\mathbf{x} = (x_\alpha)$, where the α runs through all 8 splits of X (including the trivial split $\{X, \emptyset\}$.

They write x_{12} for the entry indexed by split $\{\{1, 2\}, \{3, 4\}\}$, etc., and x_0 for the entry indexed by the trivial split. In this chapter we will adopt the notation of Chor et al. (2000) and Hendy (2005) where the "observed data" are the 8 entries $\hat{s}_\alpha$ of the "sequence spectrum" $\hat{\mathbf{s}}$, where for example $\hat{s}_{12}$ is the number of sites where taxa 1 and 2 share one state, with taxa 3 and 4 sharing the other, and $\hat{s}_0$ is the number of constant sites. In particular the "observed data" data are labeled $\hat{s}$ and are not a estimate.

The Neyman model is parameterised by the "edge lengths" of T which are the expected numbers of substitutions per site across each edge. An edge e_α induces a split α of X, with the taxa in the different subsets separated by e. Hence, for the example in figure 3.3, the central edge e_{12} partitions the taxa as $\{\{1, 2\}, \{3, 4\}\}$. The "edge length spectrum" is the vector $\mathbf{q}$ indexed by the partitions of X, where if e_α is an edge of T, q_α is the edge length of e_α. $\mathbf{q}$ is normalised by setting

$$q_0 = - \sum_{e_\alpha \text{ an edge of } T} q_\alpha$$

while $q_\beta = 0$ for each other split β which is not an edge split of T.

Equation (1.16) enables us to describe analytically the relationship between the

spectra. Given an edge-length spectrum $\mathbf{q}$ then the expected sequence spectrum

$$\mathbf{s} = H^{-1} \exp(H\mathbf{q}) \tag{3.2}$$

is the vector of the probabilities of observing each of the 8 splits at a site. Then we can calculate the likelihood of an observed sequence spectrum $\hat{\mathbf{s}}$ as

$$L(\hat{\mathbf{s}}|\mathbf{q}) = \prod_{\alpha} s_{\alpha}^{\hat{s}_{\alpha}} \tag{3.3}$$

with the product over all splits α of X. As each s_α term can be expressed as a function of $\mathbf{q}$, we can derive the derivatives of equations (1.20) and (1.21), and find $\hat{\mathbf{s}}$ and $\mathbf{q}$ where the first derivatives are identically zero. We will write L_{12} instead of $L(\hat{\mathbf{s}}_{12}|\mathbf{q})$ for the likelihood of a site pattern of s_{12} where the observed number of patterns is $\hat{\mathbf{s}}_{12}$.

Using this approach, Chor et al. (2000) found examples of multiple optima of the likelihood function as illustrated in tables 3.1, 3.2 and 3.3.

3.1.3 Parameter correlations and multiple optima

Let us assume we have the tree $T_{12|34}$ (figure 3.3). The individual likelihood for the observed site patterns s_1, s_2, s_3 or s_{123} will increase as the corresponding edge weights q_1, q_2, q_3 or q_{123} are increased. To increase the likelihood of site pattern s_{12} we can increase q_{12}. But there are other possibilities to increase s_{12}: we can increase both q_1 and q_2 and decrease both q_3 and q_4 or vice versa. This will of course more directly change the likelihood of the site patterns L_1, L_2, L_3 and L_4, but only for common substitutions on the edges e_1 and e_2 or e_3 and e_4 will the probabilities L_{12} increase. Since there are no edges e_{13} or e_{23} for topology $T_{12|34}$ we have only the two indirect possibilities to increase the expected site frequencies s_{13} or s_{23}.

The site frequencies in table 3.1 a) of Chor et al. (2000), which lead to multiple optima, have in common that the frequencies associated with edge weights of the internal edges are higher than for the external edges. The influence of s_{13} or s_{23} on the likelihood $L(q, x)$ is therefore higher than the probabilities of external edges, especially

a)

Sequence spectrum								
	$\hat{s}_0$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_{12}$	$\hat{s}_3$	$\hat{s}_{13}$	$\hat{s}_{23}$	$\hat{s}_{123}$
A	7	0	0	1	0	1	1	0
B	14	0	0	3	0	2	1	0
C	10	2	2	4	0	1	1	0
D	1000	9	200	100	9	100	100	200
E	1000	90	90	300	90	200	100	90
F	1000	90	90	340	90	339	30	100

b)

Conjugate spectrum								
	$q_\emptyset$	q_1	q_2	q_{12}	q_3	q_{13}	q_{23}	q_{123}
A	1.92	0	0	0.13	0	0.13	0.13	0
B	2.61	0	0	0.21	0	0.13	0.04	0
C	2.19	0.15	0.15	0.40	−0.02	0.08	0.08	−0.02
D	6.86	−0.03	0.20	0.10	−0.03	0.05	0.10	0.20
E	6.84	0.06	0.06	0.29	0.06	0.18	0.04	0.06
F	6.76	0.06	0.05	0.39	0.05	0.39	−0.12	0.07

Table 3.1: Site patterns a) and split spectra b) for the examples of Chor et al. (2000). Datasets A, B and C were taken from table 1, and datasets D, E and F from table 2 in Chor et al. (2000). Many of the external edges have weight zero or are small (columns q_1,q_2,q_3, q_{123}), whereas in comparison at least two of the internal edges are high. Furthermore there are strong symmetries in the data. The entries of sequence spectrum are indexed by the taxa whose character (of two states) differs from the character at the taxon 4. For example $\hat{s}_{13}$ is the number of sites with characters at taxa 1 and 3 differing from those at 2 and 4. The Hadamard conjugation produces a conjugate spectrum q whose entries should give the corresponding edge spectrum if the data fitted a tree T exactly, with entry q_{12} being the weight on edge e_{12} (figure 3.3). However in this case there is no tree with an edge spectrum that can closely approximate this conjugate spectrum.

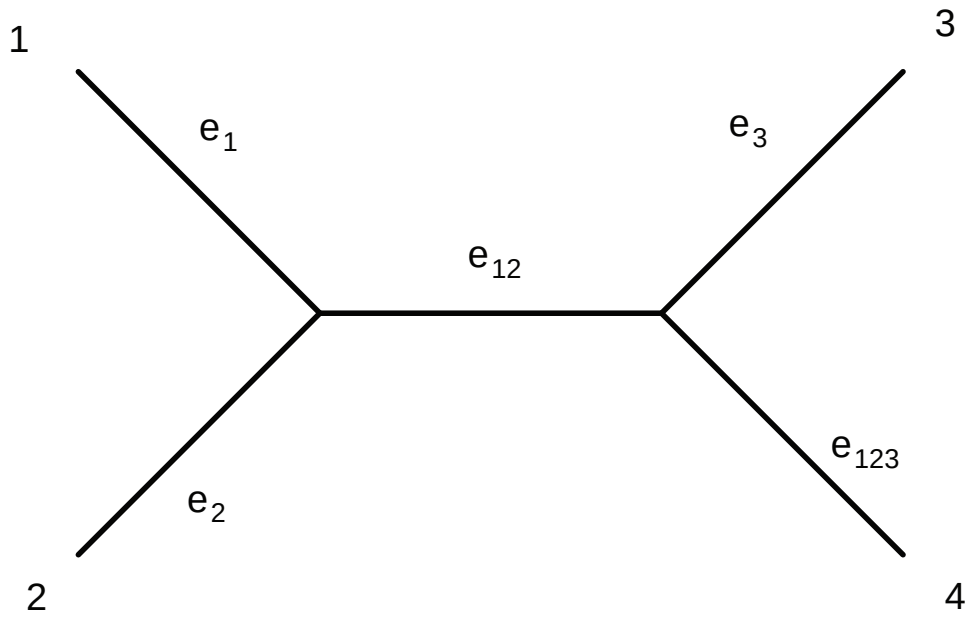


Figure 3.3: A four taxa tree on the topology $T_{12|34}$. Note the edge indexing, where the subscripts represent the set of taxa separated from taxon 4 by that edge.

since the observed site frequencies are in the exponent.

For the present work I generated a series of site patterns using the Hadamard conjugation that differ in the ratio of support for internal versus external splits (see table 3.2). The number of common site patterns and the number of total sites is fixed. Figure 3.4 shows the corresponding maximum likelihood trees for these datasets. If the support for the external site patterns grows, the differences in edge weights between adjacent external edges decreases until a critical point where only a single solution exists.

We can identify all the different kinds of multiple optima described in Chor et al. (2000) by the means of their correlation matrices. In cases without multiple optima all correlations between edge weights $cor(q_i, q_j)$, $i \neq j$ are positive. For cases of multiple

Sequence spectrum								
	s_0	s_1	s_2	s_{12}	s_3	s_{13}	s_{23}	s_{123}
A	600	0	0	300	0	200	100	0
B	600	31.58	31.58	236.84	31.58	157.89	78.95	31.58
C	600	52.17	52.17	195.65	52.17	130.43	65.22	52.17
D	600	66.67	66.67	166.67	66.7	111.11	55.56	66.67
formula	600	x	x	300 - 2x	x	200 - $\frac{4}{3}$x	100 - $\frac{2}{3}$x	x

Table 3.2: Site patterns differ in the amount of the weights of the external splits, splits separating one edge from the other. If the internal edges dominate (low values of x) then multiple optima occur (see figure 3.4).

Correlation matrix for tree A					
	q_1	q_2	q_{12}	q_3	q_{123}
q_1	1.0	0.570	-0.138	-1.0	0.0
q_2	0.570	1.0	-0.537	-0.570	0.0
q_{12}	-0.138	-0.537	1.0	0.138	0.0
q_3	-1.0	-0.570	0.138	1.0	0.0
q_{123}	0.0	0.0	0.0	0.0	1.0

Correlation matrix for tree D					
	q_1	q_2	q_{12}	q_3	q_{123}
q_1	1.0	-0.553	-0.047	-0.092	-0.092
q_2	-0.553	1.0	-0.047	-0.092	-0.092
q_{12}	-0.047	-0.047	1.0	-0.047	-0.047
q_3	-0.092	-0.092	-0.047	1.0	-0.553
q_{123}	-0.092	-0.092	-0.047	-0.553	1.0

Table 3.3: Correlation between the edge weights for the trees in 3.4. If there exist multiple optima then some of the correlations $cor(q_i, q_j)$ are positive for $i \neq j$, otherwise all the correlations are negative.

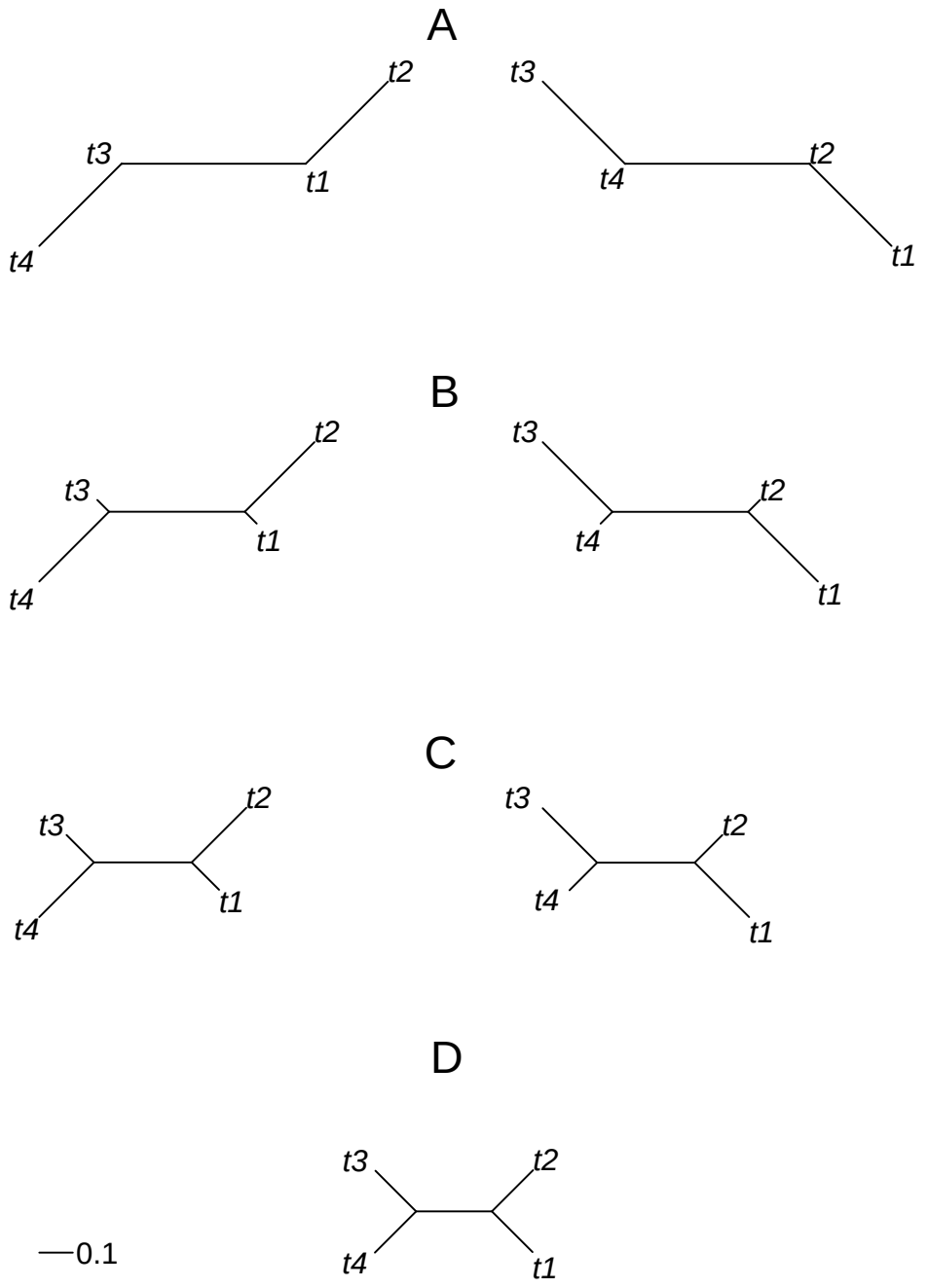


Figure 3.4: Trees of topology $T_{12|34}$ generated on the site patterns from table 3.2. For small external edge weights two possible solution exist (A-C). For data with high external edge weights the solution merge to a unique one (D).

optima where there is a ridge of parameter values defining a local optimum, we observe a variance explosion and some of the parameters are (linearly) dependent (correlations are close to 1 or -1). Cases where there are several local minima are distinct in that we can observe some positive correlations between adjacent edges. However, positive correlations are not a criterion to identify multiple optima in trees with more than 4 taxa.

Having given this background analysis, in the results section I search for (and find) simple mixtures that will generate multiple optima on the tree with the highest likelihood.

3.2 Methods

One potential criticism of the examples constructed by Steel (1994) and by Chor et al. (2000) is that it not clear that the datasets could have arisen out of any standard biological process. In what follows, we demonstrate that multiple optima arise quite readily from mixture models, and so could be present in a wide range of biological scenarios.

We generated two datasets of 10,000 sites from a mixture of two trees T_1 and T_2. The first dataset is generated from the expected frequencies of site patterns. This ensures that both optima on a tree have exactly the same likelihood. The second data set is simulated on the trees using Seq-Gen (version 1.3.2). Due to sampling error one of the optima will now have a higher likelihood than the other. This allows a comparison of how maximum likelihood and Bayesian MCMC find the different optima.

For this example both datasets were from a 50:50 mixture of tree T_1 and T_2 and the edge weights are set to 0.05 for all the external edges, and 0.5 or 0.3 for the internal edge, where the lengths are scaled to the expected number of substitutions (see figure 3.5). We used a Jukes-Cantor model to generate the data in figure 3.5, but a general time reversible model and unequal base frequencies give similar conclusions (data not shown).

For the maximum likelihood analysis we optimised the likelihood for 100 starting trees using the phangorn package (see Appendix A.2). The edge weights for each of the starting trees were randomly chosen as described in Rogers and Swofford (1999).

MrBayes version 3.1.2 (Huelsenbeck and Ronquist, 2001) was used to analyse multiple optima in Bayesian MCMC framework. Mossel and Vigoda (2005, 2006) described cases where Bayesian Inference had problems recovering phylogenies when the data came from mixtures of trees. To overcome this potential problem we run 10 chains with MrBayes on each of the two alignments as advised in Ronquist et al. (2006). Each Markov chain was run for 1,100,000 iterations from which every 1000th tree was sampled and the first 100,000 iterations were discarded as burn-in. The autocorrelation time was around 1300 for each run.

3.3 Results

3.3.1 Constructing counter-examples from mixture models

We can generate data (site patterns) that lead to multiple maximum likelihood trees with different sets of edge weights, but the same topology. We generate the data using a mixture of two trees T_1 and T_2 (see figure 3.5). A biological example would be a concatenation of two genes with different evolutionary histories.

A higher proportion of the data supports the topology T_1, because the length of the internal edge is longer than that of the second tree T_2. If we fit edge weights to the topology of T_1 we obtain trees T_3 or T_4 (figure 3.5), both of which are global maxima of the likelihood function. Fitting the edge weight on tree T_2 also leads to two equal valued maxima. In our mixture of trees the internal edges are longer than the external edges, which have to be fairly short.

If we increase the edge weights for the external edges to 0.1 in the generating trees - which is still shorter than the internal edge - and fit a tree to topology of T_1, we can observe only one (global) maximum. On the topology of tree T_2 there are still two distinct optima.

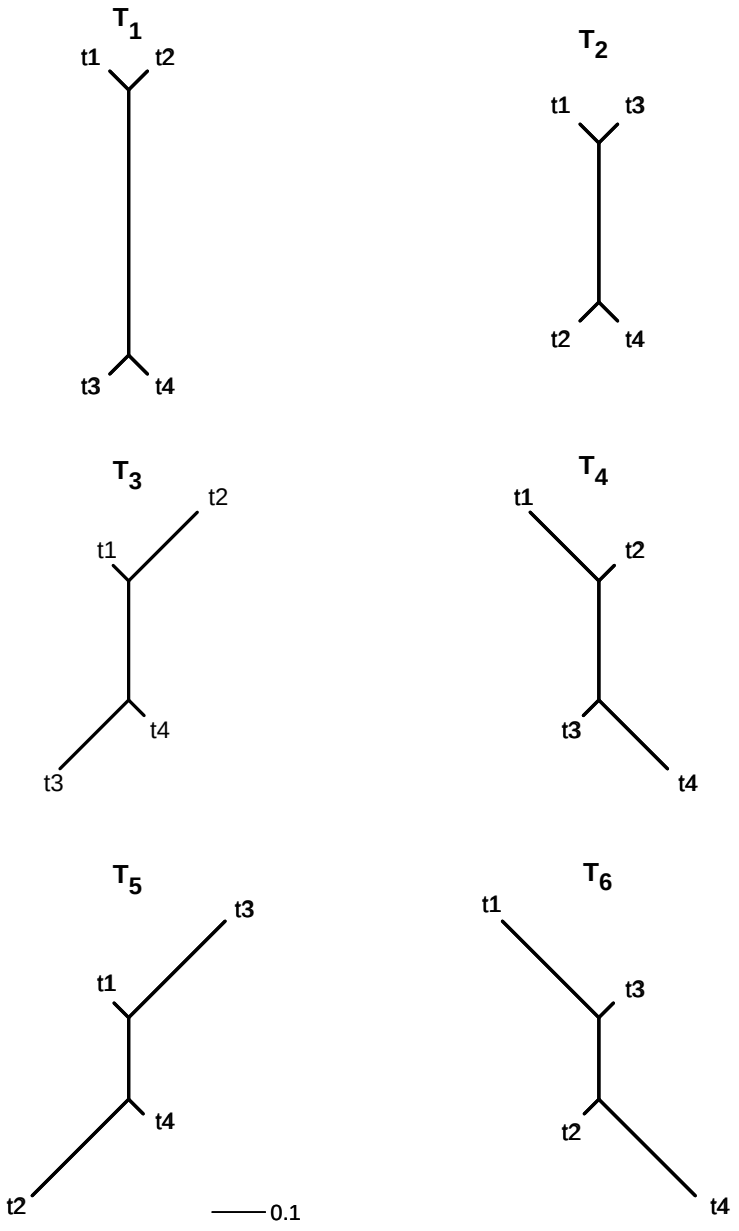


Figure 3.5: The probabilities for each DNA site pattern under a Jukes-Cantor model were calculated for trees T_1 and T_2. The probabilities were averaged to represent the expected sequence of 50:50 mixture. Fitting these mixture data to tree T_1 the likelihood function was maximised by two distinct weighted trees T_3 and T_4. By symmetry fitting towards tree T_2 would also give to equal valued maxima (T_5 and T_6).

However, this means that the history generating the data is far from being tree-like and our model for generating the data differs from that of Rogers and Swofford (1999). We can represent the conflict in the data using a splits graph representation. However the splits shown in figure 3.6 are constructed using distance methods and results will differ from a likelihood approach.

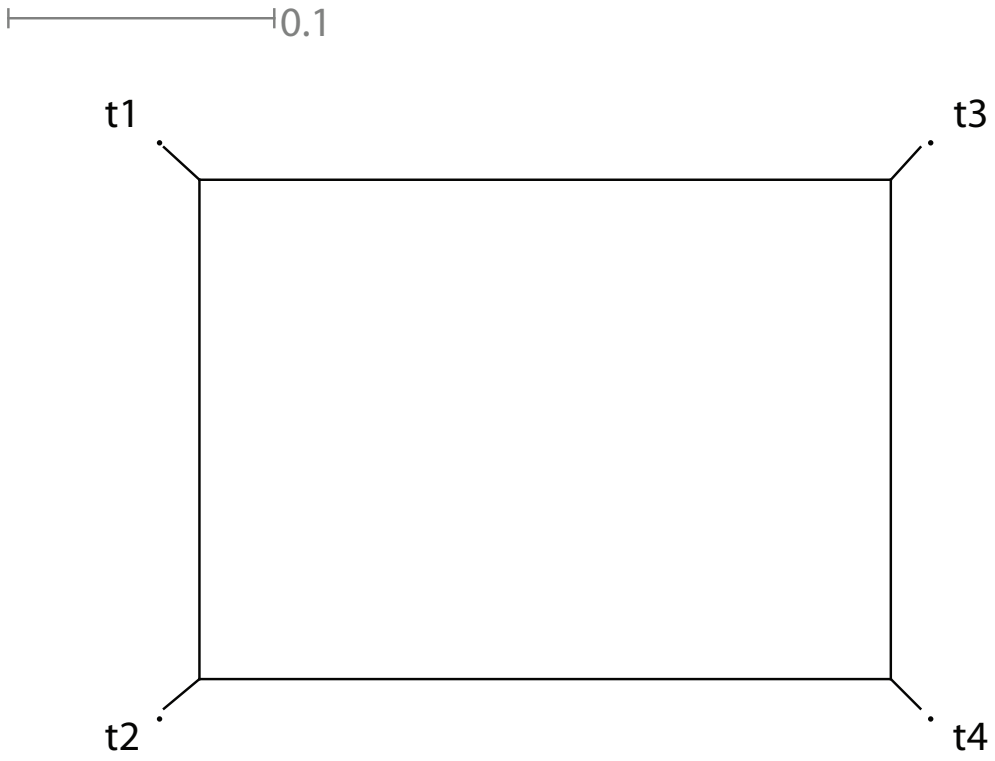


Figure 3.6: The NeighborNet fitted to the mixture data from figure 3.5. The box represents conflict in the data and shows support for both of the underlying trees T_1 and T_2. The external edges are short in comparison to the edges of the box.

An indication of conflict in the data (represented by a box) does not indicate a case of multiple optima. When using mixtures of trees we are introducing *systematic biases* to the data when estimating on a single tree. Systematic bias means that our model is misspecified; in our case we estimate a single tree when a mixture of trees is appropriate. In contrast to sampling error which decreases with growing sampling size, systematic error often is independent from the sampling size. In this case we have an

inconsistent estimator. If the method is not robust against these violations the results will also be biased.

3.3.2 Finding multiple optima with maximum likelihood and Bayesian methods

For the maximum likelihood analysis we optimised the likelihood for 100 starting trees where the edge weights were randomly chosen using the phangorn package. For both datasets, the one where the site patterns are equivalent to the expected frequencies and the one where we simulated with Seq-gen, we observed that each of the two optima were reached with about the same frequency. The (log-)likelihood for one of the trees for the simulated datasets was slightly higher (-36400.41 vs. -36466.53) and both these trees are shown in figure 3.7.

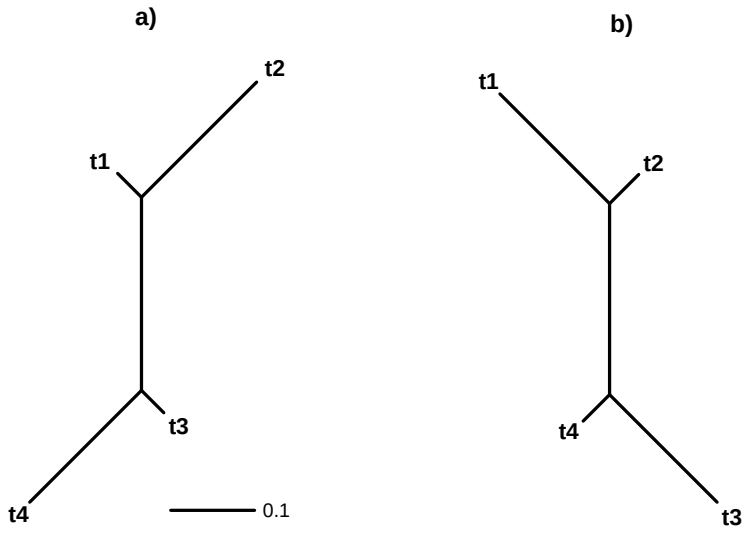


Figure 3.7: Multiple optima on tree T_1 for data simulated from a mixture of the two trees T_1 and T_2 (see figure 3.7) with Seq-Gen. The tree in a is now the global maximum with a log-likelihood of -36400.41, whereas the likelihood for the tree in b) is -36466.53.

Multiple optima affect Bayesian Inference in the similar manner as maximum like-

lihood.

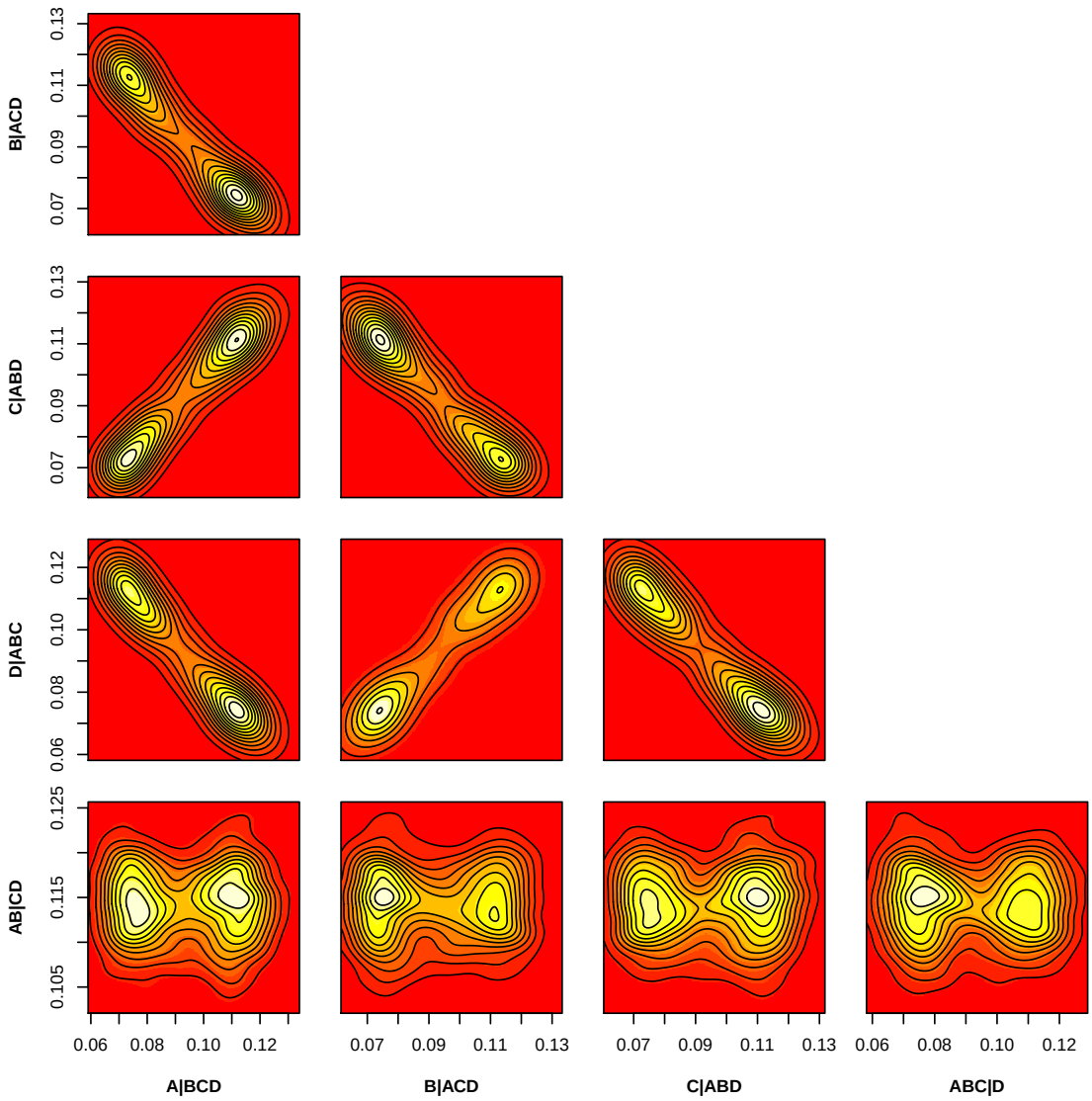


Figure 3.8: This plot shows the bivariate bimodal a posteriori distribution of the edge weights (in number of expected substitutions) of a pair of external edges. The edge weights were from a sample of a thousand trees all with the same topology. The chain found two maxima and stayed a similar amount of time in regions close to each maximum.

For the data, generated from the expected frequencies, the MCMC analysis also found two distinct global optima and stayed a similar amount of time in each of them during the sampling. Figure 3.8 shows the posterior distribution for two of the external edges. The two modes of the distribution confirm that there are two local optima for

this tree topology. We checked carefully that no other tree topology appeared in the sample, but after the burn-in was excluded, only trees of one topology remained in the sample.

We have to be careful with our conclusions, since measures such as the posteriori mean or median, which are frequently used to describe the parameters of a Bayesian analysis, are really only meaningfully defined for unimodal distributions (Gamerman, 1997). The Bayesian credible intervals and confidence intervals for maximum likelihood estimates might not be continuous if we have multiple optima, but this can be a nice indication for multiple optima.

For the data set based on the expected site frequencies, we observe that the individual Markov chains may not sample trees evenly from the two optima. However if we concatenate the sample both optima are about equally often sampled (see figure 3.8). For the simulated data with sampling error on the other hand we observe that now the tree with the higher loglikelihood is now far more often sampled (figure 3.9).

3.4 Conclusions

The primary conclusion from this chapter is that quite simple mixture models can generate data where multiple optima can occur. These optima can occur on the tree with the highest likelihood. Sampling error can lead to differences in the values of the the multiple optima. Thus the question of multiple optima on a single tree is still an important question thats needs to be considered.

Some cases Chor et al. (2000) presented arose from the very simple model and the very restricted number of site patterns they used leading to the covariance matrix being singular. This corresponds to highly correlated parameters (correlation 1 or -1), where some of the examples showed a ridge of solutions. This means an infinite number of multiple optima are found, but on the other hand these cases are easy to identify as the variance tends to infinity for these parameters.

Critical are those examples described in section 3.3.1, as we do not have a crite-

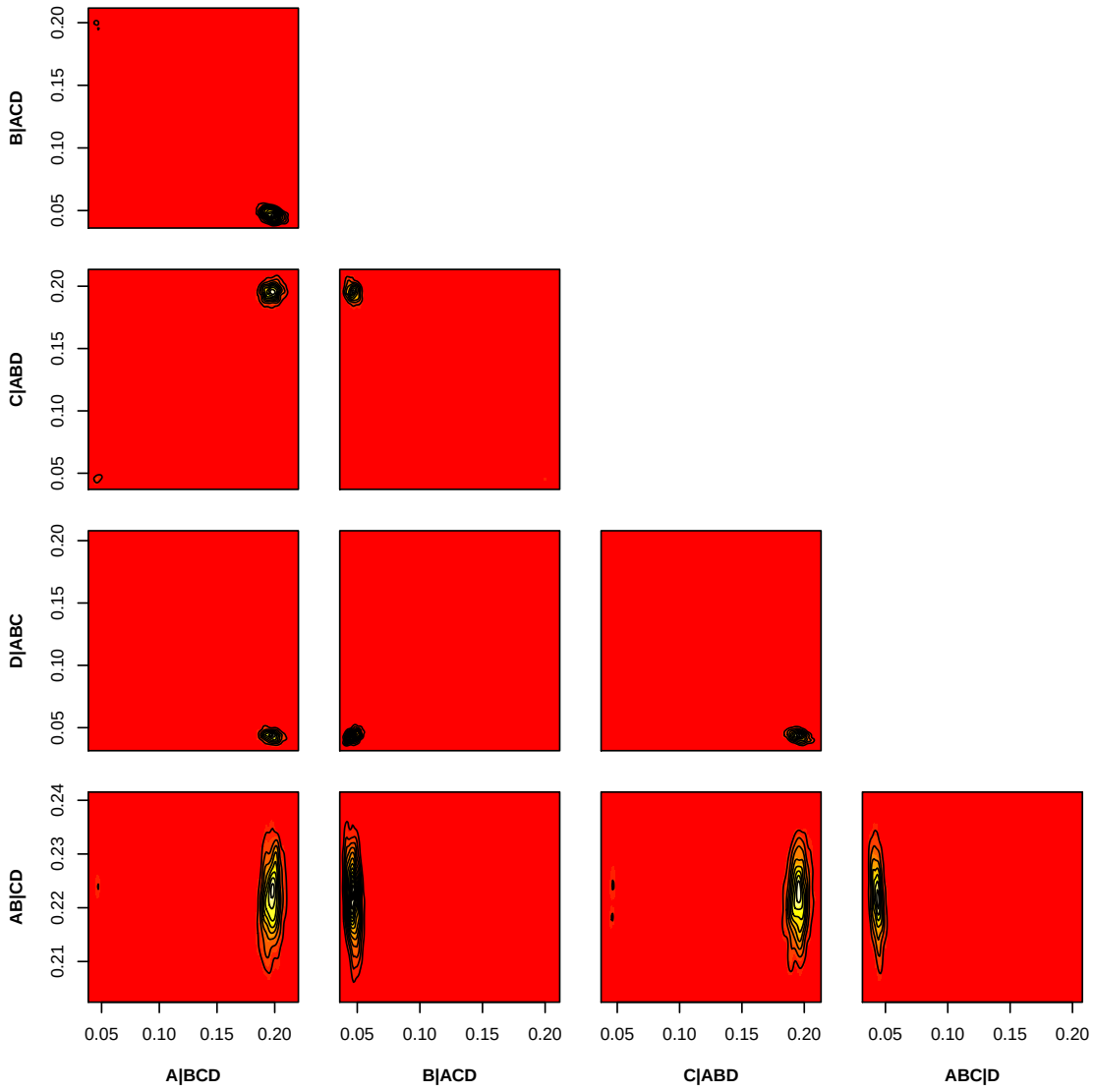


Figure 3.9: The a posteriori distribution for one of the external edges for data with sampling error. The distribution is now unimodal.

rion which immediately can be used to discover multiple solutions. Optimisation from multiple starting trees should be carried out in case of maximum likelihood analysis or running several chains for Bayesian analysis can help to identify cases of multiple optima. The behaviour between both methods differs more when sampling error is introduced. Bayesian methods tend then to favour only one (the global) maximum, whereas maximum likelihood methods (in our symmetric example) still may sample from both optima evenly.

More parameter rich models tend to make the likelihood surface smoother. This has the disadvantage that the actual optimisation will be slower and the variance of the parameters higher. But it also makes it more unlikely that symmetries in the data occur for the model. Therefore it is more likely that there will exist a unique global maximum, but local maxima are still possible.

A totally different strategy is to try to see if the model used in the inference is appropriate to generate the data. For example splits graphs can be used to visualise conflict in the data. If multiple optima are the result of a mixture of different trees or mechanism as in the covarion case of Kolaczkowski and Thornton (2004) more appropriate, but also more complicated inference models like mixture models (as described in chapter 2) or partition models, which will be considered in the next chapter, may overcome this problems. Also goodness-of-fit statistic or parametric bootstrap could be used to detect if the model choice was poor. However the opposite is not true; a poor fit does not mean the presence of multiple optima and the power of such a test is often weak Waddell et al. (2008).

Chapter 4

Partition Models models for multi-gene datasets

Partitioning data into subsets is often used where there is knowledge about the structure of the data. Examples include partitioning nucleic data by codon position (and allow different transition models for each partition) and for genomic data we can partition by genes (Bevan et al., 2007). The number of parameters grows with the number of partitions. We expect that many genes have similar evolutionary histories and be under similar biochemical constraints, so the parameter estimates for such genes should be similar. If we allow every gene to have its own set of parameters, we may overfit the model, but if we ignore these structures we may lose important information (Pupko et al., 2002). A potential compromise is to cluster the genes into groups of homogeneous genes, and use these groupings or clusters in the further analysis.

The primary aim of clustering of genes is to reduce the number of parameters in further phylogenetic analysis, but the grouping into clusters can itself contain useful information. If genes are in the same complex they may have evolved under similar constraints. Gene trees can give information about the interaction between genes (Wu et al., 2000).

There are many different reasons to use clustering or *unsupervised learning* on genes or other known partitions of a dataset. Firstly we may want to group the data into

different clusters to better account for heterogeneity in the data. Secondly we may want to explore the relationship of the clusters to each other. For example we may want to find groups of co-evolving genes or visualise the differences of gene trees using multivariate techniques like multidimensional scaling or principal component analysis.

4.1 Methods

4.1.1 Stochastic Partitioning

Here I describe an algorithm for clustering genes in groups with similar evolutionary patterns. I introduced k-cluster, a stochastic partitioning algorithm based on the algorithm described in Fahrmeir and Tutz (1996, p.513). The ideas behind this stochastic partitioning algorithm are similar to the k–means algorithm of Hartigan and Wong (1979). Similar approaches to cluster genes are described in Li et al. (2008) and Shi et al. (2008).

The algorithm classifies N genes into k classes. The number of genes may vary from about 100 to several thousands of genes. The number of classes k is often not known beforehand, but it can be inferred with the AIC or BIC, as shown below. We assume that k is much smaller than N, and k is often in the range between 2 to 20.

We first assign every gene randomly into one of k classes and optimize the unknown parameters θ_k for these. The overall likelihood is just a generalisation of equation (1.5)

$$l(x, \theta_1, \ldots, \theta_k) = \sum_{i=1}^{N} \sum_{j=1}^{n_i} \log f(x_{ij}, \theta_{c_i}) = \sum_{i=1}^{N} l(g_i, \theta_{c_i}) \tag{4.1}$$

where x_{ij} is the $j-$th site pattern belonging to gene i, c_i is the class that gene i belongs to, θ_{c_i} are the parameters corresponding to class c_i, and $l(g_i, \theta_{c_i})$ is the (log-)likelihood for gene i. After this we compute the posterior probability for each gene for all the k classes. If the the posterior probability of a gene in class c_i is higher in a different class, then this gene will be moved to that class. Every time a gene changes class, the likelihood (equation 4.1) is increased. The algorithm terminates when no changes to

the classes can increase the likelihood. Algorithm 4.1 describes the algorithm.

Algorithm 4.1 The k-cluster algorithm

1. Choose a start cluster, i.e. assign each of the N genes randomly to one of the k classes.
2. Maximize the parameters $(\hat{\theta}_1, \ldots, \hat{\theta}_k)$ for all classes.
3. Compute the posterior probabilities for each gene with the parameters from each class.
4. Move each gene into the class for which it has highest posterior probability.
5. If any gene changes its class go to step 2, else STOP.

The k-cluster algorithm is a greedy algorithm which does not guarantee to return the best partitions, therefore the algorithm is best run several times to figure out which clusters are stable. There exist many possible variations of this algorithm. One way to speed up the process would be to not recompute all the parameters (step 2) at every iteration. First there are many possible ways to initialise the first cluster. We choose k to compute the posterior distributions and then change the membership in the classes all at the same time. Updating all the class memberships at once has the advantage that is fast, but it can happen that afterwards some classes could be empty. This is most likely to happen, if we use a random initialisation of the classes. Another approach is to update the likelihoods sequentially after a single gene has changed its class, changing genes first which most increase the log-likelihood (equation 4.1).

Our approach (algorithm 4.1) generally converges in fewer than 10 iterations and has the advantage over sequential updating, that we save computing time since we have to recompute the likelihood of the clusters less often (for an example see figure 4.1).

Furthermore we can introduce restrictions on the size of the classes, for example to ensure that each class contains a minimal and/or maximal number of genes.

The computationally expensive part is optimising the likelihood in step 2 of algorithm 4.1. The computational costs increase therefore with the number of classes, as more parameters have to be optimised and often more iterations are needed until con-

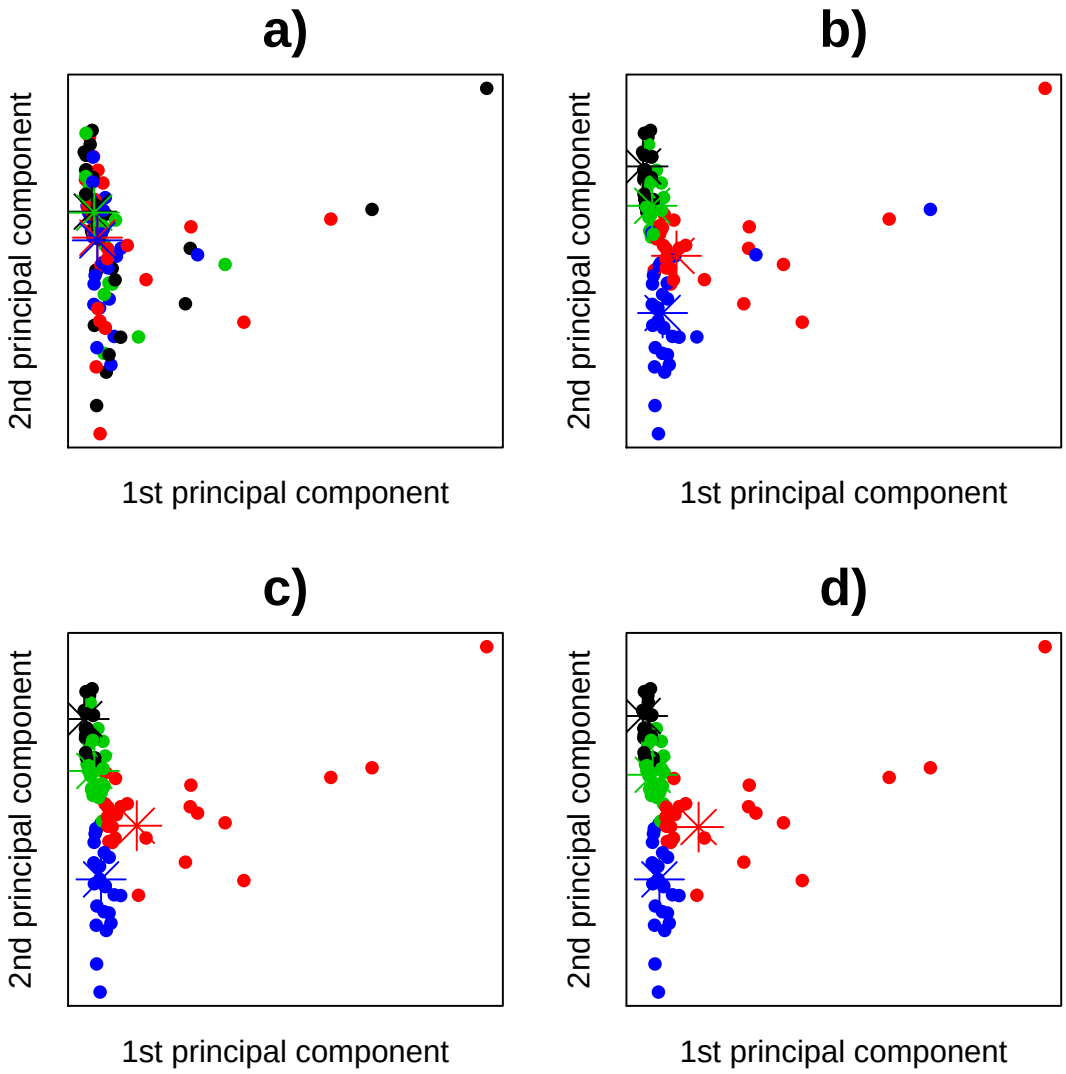


Figure 4.1: Each point corresponds to one gene in the data set of Rokas et al. (2003). The points are projected using multidimensional scaling based on the parameter estimates for the gene trees. The four colors represent the class memberships. Figure a) shows the random initialisation at the beginning, b), c) and d) show the classes after the 3rd, 5th and 7th exchange of the clusters. In this example the algorithm converged in only 7 iterations. One can observe that the clusters are getting more homogeneous during the process.

vergence of the algorithm. However if k is large, then parameters for each class are usually faster to optimise, as the sequence length for each class decreases.

The algorithm is not limited to which set of parameters are optimised for each cluster. Any combination of parameters which are usually used to optimise ML trees, like edge weights, topology, base frequencies or transition matrices, are also possible. However the computational effort depends strongly on how many parameters for each class are optimised, and if tree rearrangement is performed. This algorithm allows that different classes could have different topologies, but with an additional computational cost of optimising the topology for every class.

Choosing the number of clusters

We will often be in the situation that we do not know *a priori* the number of classes. We can choose a data driven approach to define the number of classes. We have to keep in mind that there are two main reasons for applying the stochastic partitioning algorithm. The first is to account for heterogeneity and the second is to find patterns in the data.

When we are interested in patterns in the data we will try to optimise the number of classes in a way to return stable clusters. There is huge amount of literature available from the statistical community that addresses this issue; for example Fowlkes and Mallows (1983); Ben-Hur et al. (2002); Hornik (2005). Too many classes will lead to a high variability in the clusters, too few may miss specific features of the data. When we are interested in accounting for heterogeneity, the AIC or BIC can be used to give an indication of the goodness of fit (see figure 4.2). We observed - as expected - that the optimal number of classes for the AIC is higher than for the BIC, and that the clusters were more stable for the different random starts for the optimal k suggested by the BIC.

Since the algorithm is stochastic, starting with different random initialisation of the clusters can result in different groupings. When the number of classes chosen is too high, the clusters will be not stable., i.e. the same genes may not cluster together in

different runs. A good choice of k should give stable clusters and for this reason we choose k based on the BIC.

In addition we can demand that the trees or models differ for each cluster. The Kishino-Hasegawa test (Kishino and Hasegawa, 1989) or Shimodaira-Hasegawa test (Shimodaira and Hasegawa, 1999) are both commonly applied to test whether two (or more) trees differ significantly. As Goldman et al. (2000) pointed out the Kishino-Hasegawa test is only valid, if the trees/models, which will be compared, are specified beforehand. We therefore use the Shimodaira-Hasegawa test. We have to be aware that we perform in total $k(k-1)$ comparisons, where k is the number of classes, as we always compare for each class the ML tree in that class with all other trees. The Shimodaira-Hasegawa test adjusts the p-value for the comparison of the ML tree against the other trees, but we perform such a test for each cluster, so we need a correction for multiple testing (e.g. Bonferroni-correction). This can lead to situations where we have two trees T_i and T_j which are the maximum likelihood trees of clusters i and j, for which we can reject the hypothesis that T_i and T_j are equally likely given the data of cluster i, but not for data of cluster j. Furthermore the k-cluster algorithm is not hierarchical, in that we cannot conclude that the members of the two classes, with trees that are not significantly different, will join if we reduce k.

4.1.2 GO-analysis

GO-analysis offers a way to test if the genes we find in the cluster contain genes with similar biological functions. *Gene ontology* (GO) (Ashburner et al., 2000) is a classification of genes and gene products. There exist three ontologies, describing molecular function, biological processes, and cellular components of genes or gene products. Each gene can be associated with zero, one or more GO terms. The GO terms build a hierarchy, where special terms are always part of one or more general terms (see figure 4.4).

If we observe heterogeneity in our data that we can link to different functional constraints, we should account for these, for example using a covarion model or just

performing separate analysis for each functional group. Using a concatenation of genes in cases where the data represent a mixture of different evolutionary histories can mislead the analysis as shown in Matsen and Steel (2007). To identify if the differences in the clusters are likely due to functional constraints, we try to identify biological functions which are overrepresented in any of the clusters. To connect the function of the genes with our data set we use the gene ontology. Analyses of this kind are commonly used to find differentially expressed genes in microarray analysis (Gentleman et al., 2005; Voelckel et al., 2008), but are not common (yet) in phylogenetics.

We first derived all GO terms associated with 106 genes of the yeast data set of Rokas et al. (2003). We computed for each gene all the associated GO terms, including the more general terms. In the second step we discarded all GO terms which we observed less than 10 times in total, reducing the number of the GO terms from 528 to 53.

We use a clustering of the genes from the k-cluster algorithm with six partitions. Using a low value for k ensures that the the clusters are distinct. Furthermore if we introduce more clusters we will lose power in the subsequent tests. We perform the Fisher-exact test to establish if any of the 53 terms is overrepresented in one of the 6 clusters we got from the k-cluster algorithm. That leaves us with $6 \times 53 = 318$ tests where we have to adjust the p-value for multiple testing. Due to the hierarchical character of the GO terms, the tests are correlated. We correct the p-values of the individual test for multiple testing with a Bonferroni correction, and a less conservative correction due to Benjamini and Yekutieli (2001).

Relationship to other methods

The k-cluster algorithm is closely related to algorithm 2.1 for the estimation of mixture models and the estimation-maximisation (EM)-algorithm (Dempster et al., 1977). The main difference is that the k-cluster algorithm always assigns a gene to the class with the highest posterior distribution, whereas a a EM-algorithm (or mixture models) assigns mixing weights to a gene proportional to its posterior distribution. The two algorithms would be identical, if each gene would have a posterior probability of 1 for one class

and zero for all other classes. The major advantage of the k-cluster algorithm is that, since there is a finite number of genes, the k-cluster algorithm will terminate when no genes change class. This avoids the sometimes slow convergence of the EM-algorithm described in chapter 2, caused by the iterative updating of the mixing proportions and the estimates. In addition the results of the clustering are easy to interpret; however we should keep in mind that the support for the grouping of a gene may depend on a marginal difference in the posterior support of the cluster.

4.1.3 Other approaches to clustering genes

Waddell et al. (2007) applied gene clustering algorithms to detect potential protein-protein interaction. They used hierarchical clustering algorithms (UPGMA, NJ) and multidimensional scaling (MDS) (Gower, 1966). These algorithms use a distance matrix as input. The distances were estimated from edge length estimates of the different gene trees. Therefore it is assumed that all the genes evolved under a common unweighted tree.

The algorithms used by Waddell et al. (2007) are based solely on the edge weights, other differences such as differences in the base composition of the genes are not included in these models. It is possible to expand these models with additional parameters, for example when analysing the base composition of genes, one needs to account for the correlation between these additional parameters and the edge weights.

The models described here assume that we know or can estimate with a certain confidence the phylogenetic trees for the clusters, although in the k-cluster algorithm the trees are allowed to vary between genes.

4.1.4 Hadamard and distance Hadamard

We will compare the edge spectra of the Hadamard conjugation to determine the relationship between the different genes. With nucleic acid data the Hadamard conjugation can be calculated on either directly from the original data or be derived from pairwise distances (distance Hadamard). We use the distance Hadamard in the example of the

chloroplast, where the data are given as amino acids. Therefore we have first to compute a matrix of pairwise distances between the species for each gene. The elements of such a distance matrix are highly correlated, and to compare two of such matrices we could use the Mantel test (Mantel, 1967; Sokal and Rohlf, 1995). The elements of the edge spectra on the other hand are far less correlated (Waddell et al., 1994).

So in the first step we compute the edge spectra using the Hadamard conjugation for each gene. Having constructed the distance matrix we can apply multidimensional scaling (MDS) (Gower, 1966) on the distances to visualise the relationship between the different genes, as can be seen in figures 4.3 and 4.6).

These methods are restricted to data sets with low number of species as the Hadamard conjugation creates an array of size $4^{(m-1)}$ for the Hadamard conjugation for nucleic acid data and $2^{(m-1)}$ for the distance Hadamard, where m is the number of species. The advantage is that no assumption is made about the tree topology. More work needs to be done to explore whether other distance measures or the correlation between edge spectra serve the purpose of visualising the relationship between genes better (Waddell et al., 2007).

4.2 Results

4.2.1 Yeast data

We applied the k-cluster algorithm to the data set of Rokas et al. (2003), which is an alignment of 106 genes. Table 4.1 shows a summary of results with the number of classes varying between 2 and 20. We compared these cluster models with the concatenated dataset ($k = 1$) and with the model where each gene is allowed to have its own set of edge lengths (k=106). Models with a higher number of classes generally have better likelihood values, but as the algorithm starts with a random partition, the algorithm can get stuck in local optima. To avoid this, several runs of the algorithm are advisable. The main problem which arises is to find the optimal number of classes. We use the BIC as a optimality criterion, but one still has to try several values of k.

Partitions	l(θ)	df	AIC	BIC
1	-682292.09	23	1364656.18	1365007.25
2	-680445.38	36	1360962.76	1361313.84
3	-680023.17	49	1360144.35	1360622.20
4	-679667.28	62	1359458.56	1360063.19
5	-679467.32	75	1359084.63	1359816.04
6	-679396.85	88	1358969.70	1359827.89
7	-679291.02	101	1358784.05	1359769.01
8	-679209.11	114	1358646.22	1359757.96
9	-679179.24	127	1358612.49	1359851.01
10	-679117.90	140	1358515.81	1359881.11
11	-679098.16	153	1358502.33	1359994.41
12	-679043.47	166	1358418.95	1360037.80
13	-678985.42	179	1358328.83	1360074.47
14	-678962.98	192	1358309.96	1360182.37
15	-678898.19	205	1358206.39	1360205.58
16	-678876.21	218	1358188.42	1360314.39
17	-678838.34	231	1358138.68	1360391.43
18	-678782.34	244	1358052.68	1360432.20
19	-678763.64	257	1358041.27	1360547.58
20	-678754.95	270	1358049.90	1360682.98
106	-677685.76	1388	1355443.51	1355794.59

Table 4.1: Summary of runs for the stochastic partitioning algorithm. Always the highest log-likelihood value of 10 random starts for each k is presented. The likelihood values increase with growing number of clusters, the AIC reaches its optimum at about 19 clusters. The number of degrees of freedom is the sum of the 10 parameters for the GTR + Γ + I - model and the number of edge weights estimated. There is a considerable difference between the AIC and BIC due to the long sequence. The data are plotted in figure 4.2

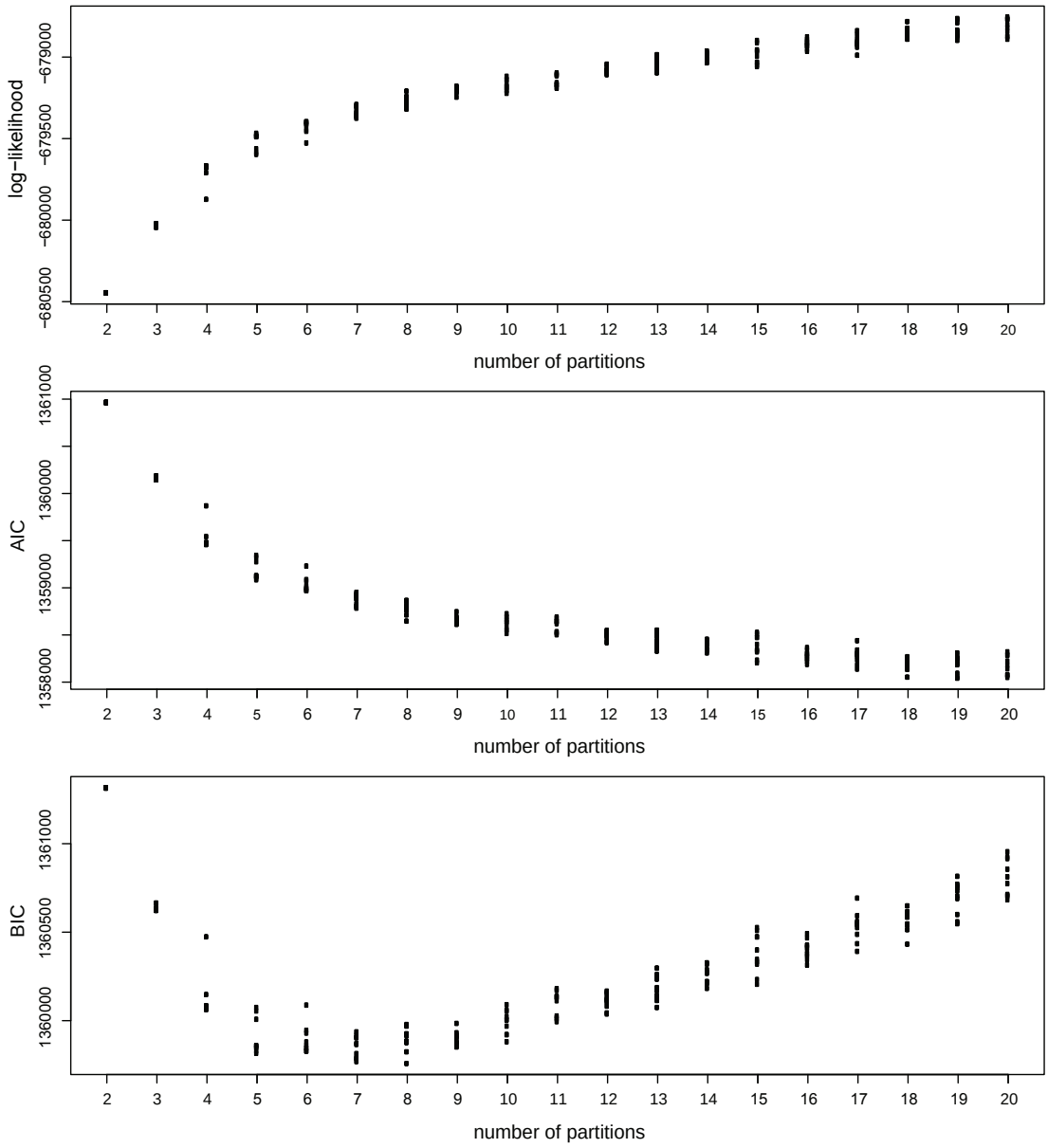


Figure 4.2: Likelihood, AIC and BIC for different number of partitions for the Rokas et al. (2003) dataset. The model with one class uses the concatenated data set, for the 106 classes each gene represents a own class. The likelihood increases with the number of classes, however the optimal number of classes is at 6-8 if we use the BIC as optimality criterion and at about 18-20 for the AIC. The individual points are from 10 different runs.

In the example above we allowed only the edge lengths to vary in the different classes. We can optimise all the parameters of the model (topologies, rate matrix, base frequencies, invariant sites etc.) chosen in this algorithm. With the complexity of the model the run-time of the algorithm increases.

The k-cluster algorithm does not assume that there is only one topology, but assumes only that genes belonging to a cluster have evolved homogeneously, for example they share the same topology and have similar base frequencies, mutation rates etc. The membership of a gene to a cluster depends only on the posterior likelihood of the gene. However the clusters are allowed to have evolved independently.

We also visualise the relationship between the 106 genes using the multidimensional scaling on the edge spectra of a Hadamard conjugation. As the yeast data contains

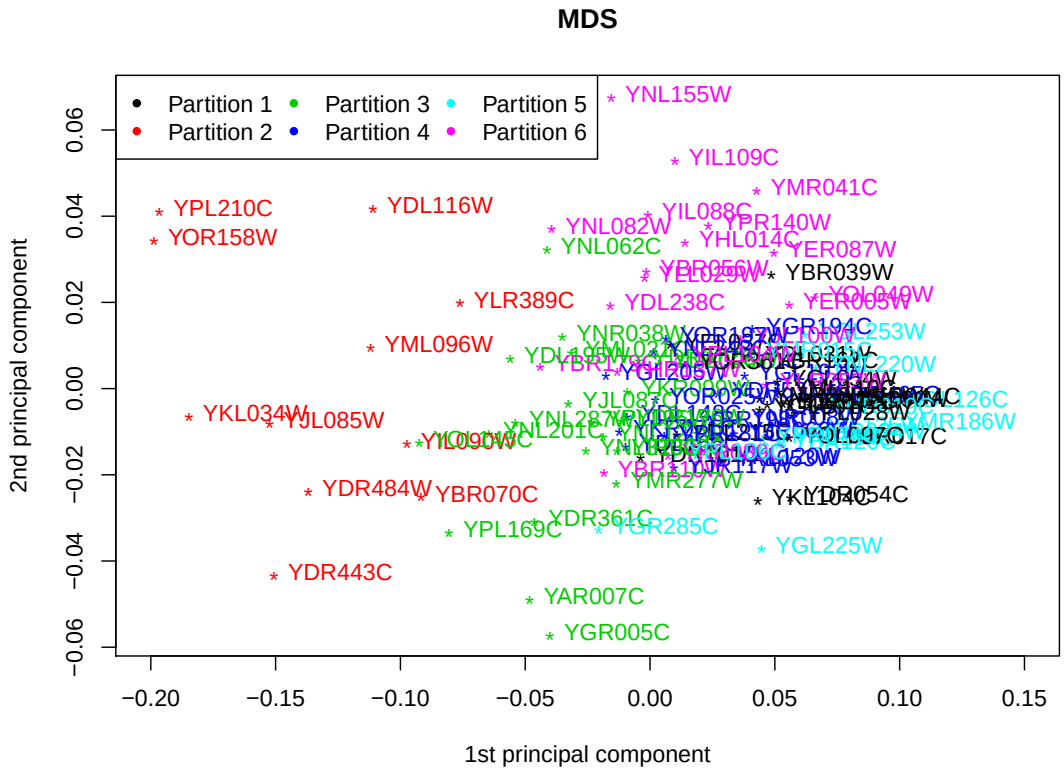


Figure 4.3: First two principal components of a multidimensional scaling based on edge spectra of the yeast data set. Each color represents a cluster estimated by the k-cluster algorithm (see table 4.2).

genes for only 8 different species we can use the (4-state) Hadamard conjugation. We use Euclidean distances to compute a matrix of pairwise dissimilarities for the 106 edge spectra. In figure 4.3 the first two principal components of the MDS are plotted and the genes a colored according to the clusters estimated by the k-cluster method and the clusters group together.

In addition the Shimodaira-Hasegawa test was applied to test if the differences between the trees found in the different clusters are significant (see table 4.2). For a choice of $k = 6$ classes, all the trees are significant to a significance level of $\alpha = 0.05$, including a Bonferroni correction for multiple testing.

Partition	Comparison	ln L	Diff ln L	p-value (raw)
1	Tree 1	-131933.60		
	Tree 1 vs. 2	-134287.19	2353.60	0.0000
	Tree 1 vs. 3	-132763.74	830.14	0.0000
	Tree 1 vs. 4	-132305.66	372.07	0.0000
	Tree 1 vs. 5	-132628.95	695.36	0.0000
	Tree 1 vs. 6	-132991.00	1057.40	0.0000
2	Tree 2	-76845.91		
	Tree 2 vs. 1	-78310.61	1464.69	0.0000
	Tree 2 vs. 3	-77180.13	334.21	0.0000
	Tree 2 vs. 4	-77628.66	782.74	0.0000
	Tree 2 vs. 5	-79816.70	2970.78	0.0000
	Tree 2 vs. 6	-77527.30	681.39	0.0000
3	Tree 3	-140248.95		
	Tree 3 vs. 1	-141141.82	892.87	0.0000
	Tree 3 vs. 2	-140660.69	411.74	0.0004
	Tree 3 vs. 4	-140561.63	312.68	0.0000
	Tree 3 vs. 5	-142909.33	2660.38	0.0000
	Tree 3 vs. 6	-140832.77	583.82	0.0000
4	Tree 4	-98058.53		
	Tree 4 vs. 1	-98329.86	271.33	0.0000
	Tree 4 vs. 2	-98867.11	808.59	0.0000
	Tree 4 vs. 3	-98226.66	168.13	0.0057
	Tree 4 vs. 5	-99313.93	1255.41	0.0000
	Tree 4 vs. 6	-98396.92	338.39	0.0000
5	Tree 5	-106830.27		
	Tree 5 vs. 1	-107356.63	526.36	0.0000
	Tree 5 vs. 2	-110979.01	4148.74	0.0000
	Tree 5 vs. 3	-108902.31	2072.04	0.0000
	Tree 5 vs. 4	-108192.46	1362.19	0.0000
	Tree 5 vs. 6	-109417.39	2587.12	0.0000
6	Tree 6	-125455.49		
	Tree 6 vs. 1	-126527.39	1071.91	0.0000
	Tree 6 vs. 2	-126371.88	916.39	0.0000
	Tree 6 vs. 3	-125988.31	532.82	0.0000
	Tree 6 vs. 4	-125980.21	524.72	0.0000
	Tree 6 vs. 5	-128568.05	3112.56	0.0000

Table 4.2: Result of six Shimodaira-Hasegawa tests comparing the trees (Shimodaira and Hasegawa, 1999), one for each partition. The (raw) p-values need to be adjusted for multiple testing for the 6 partitions, for example using a Bonferroni-correction all (raw). All comparisons are significant to a level of $\alpha = 0.05/6 \sim 0.0083$.

4.2.2 Comparison with Gene Ontology

In the last section we used a clustering algorithm to account for model heterogeneity. Now we will change the focus to test whether the observed heterogeneity is likely to be due to functional constraints.

Figure 4.4 shows the ontology of the molecular function of the 53 filtered GO-terms in form of a directed acyclic graph for the yeast dataset. Each node represents an individual GO-term and edges indicate the hierarchical relationship between them. The segments in the nodes display the relative frequencies of observing that GO-term in any of the cluster.

Table 4.3 shows the results of the Fisher exact test, for all GO terms which are over-represented in any of the cluster, starting with the lowest p-value. After the correction for multiple testing (Bonferroni correction or Benjamini and Yekutieli (2001)), none of the GO terms is over-represented to a level of $\alpha = 0.05$ in any of the clusters.

This kind of analysis is likely to be used more frequently, as more genomic datasets with several thousands of genes become available. This approach may help to identify the functions of genes from complexes of co-evolving genes. Identifying clusters of co-evolving genes is similar to clustering genes according to their expression levels of microarray data (Eisen et al., 1998). If we gain a better understanding of how and which biological constraints effect, for example, the rates of evolution in different genes, we can use this classification in further analysis. For example, mitochondrial and nuclear data show different evolutionary patterns because of different heritage patterns. We may identify other factors which impose evolutionary constraints, for example if genes on the same chromosome cluster together.

Depending on how sensitive these analyses can be, it may also be possible to predict the function of unknown genes if, for example, all other genes in the cluster belong to the same complex.

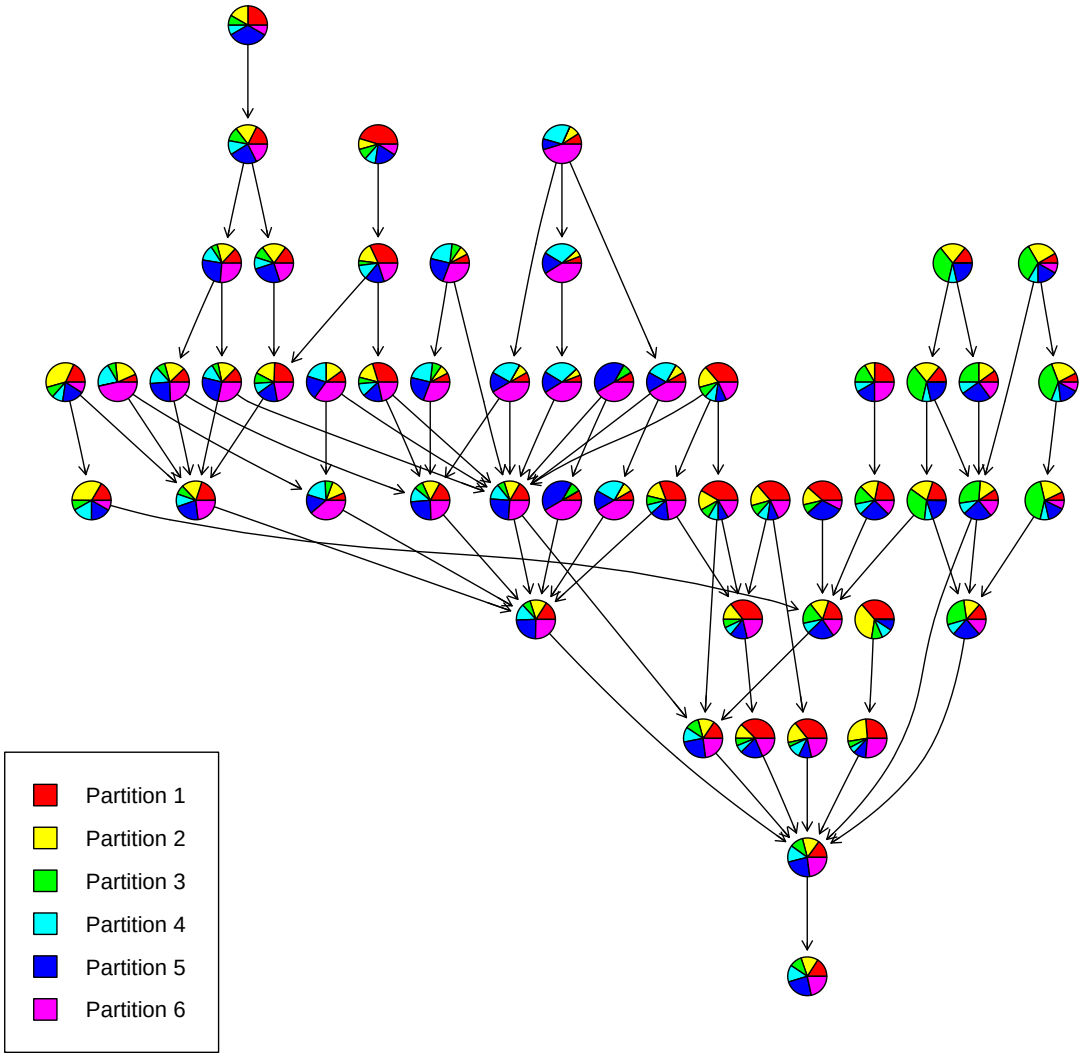


Figure 4.4: Directed acyclic graph of the gene ontology for the biological processes of the 106 genes in dataset of Rokas et al. (2003). Nodes are representing the individual GO terms, edges are representing the relationship between the terms, with the most general term on the root of the tree. The segments of the nodes are proportional to the relative frequencies of observing the GO term in the specific cluster.

	GO-term	Cluster	raw p-value	Bonferroni	BH	observed	expected
1	GO:0033036	3	0.0006	0.1803	0.1803	6	1.453
2	GO:0010467	1	0.0021	0.6660	0.2863	10	4.491
3	GO:0008104	3	0.0039	1.0000	0.2863	5	1.349
4	GO:0046907	3	0.0057	1.0000	0.2863	5	1.453
5	GO:0051649	3	0.0057	1.0000	0.2863	5	1.453
6	GO:0006139	1	0.0065	1.0000	0.2863	11	5.934
7	GO:0051234	3	0.0073	1.0000	0.2863	6	2.179
8	GO:0009058	6	0.0080	1.0000	0.2863	12	6.726
9	GO:0051641	3	0.0081	1.0000	0.2863	5	1.557
10	GO:0051179	3	0.0096	1.0000	0.3048	6	2.283
11	GO:0032774	1	0.0150	1.0000	0.4160	5	1.764
12	GO:0016070	1	0.0185	1.0000	0.4160	8	4.009
13	GO:0009059	6	0.0186	1.0000	0.4160	7	3.255
14	GO:0044237	6	0.0206	1.0000	0.4160	22	18.226
15	GO:0045184	3	0.0207	1.0000	0.4160	4	1.245
16	GO:0065007	1	0.0209	1.0000	0.4160	6	2.566
17	GO:0050794	1	0.0230	1.0000	0.4309	5	1.925
18	GO:0006810	3	0.0317	1.0000	0.5482	5	2.075
19	GO:0022613	1	0.0336	1.0000	0.5482	5	2.085
20	GO:0016043	3	0.0345	1.0000	0.5482	8	4.670
21	GO:0043283	1	0.0399	1.0000	0.5630	11	7.217
22	GO:0006082	6	0.0407	1.0000	0.5630	7	3.689
23	GO:0019752	6	0.0407	1.0000	0.5630	7	3.689
24	GO:0008152	6	0.0442	1.0000	0.5635	22	18.877
25	GO:0050789	1	0.0469	1.0000	0.5635	5	2.245
26	GO:0006950	2	0.0478	1.0000	0.5635	4	1.557
27	GO:0065003	2	0.0478	1.0000	0.5635	4	1.557

Table 4.3: Biological function associated with the 6 clusters. The 27 GO terms that had a raw p-value ≤ 0.05 are presented. However none of the GO terms is significantly over-expressed after the p-values have been adjusted for multiple testing (Bonferroni correction or Benjamini and Yekutieli (2001))

4.2.3 Exploring relationships between genes within the chloroplast

The next example examines the relationship of different protein complexes encoded by the genomes of 19 anciently diverged species. The main interest in this study is to investigate if structural constraints lead to lineage specific evolution of genes, i.e. if gene trees which are involved in the same biological complex show a closer similarity. We have chosen two amino acid sequences for each of 7 different photosynthetic complexes. Table 4.4 shows a summary of the sequences.

Complex	Sequence length	Function	gene rate
atpA	485	atp synthase complex	1.0142756
atpB	448	atp synthase complex	0.7009842
petA	224	Photosystem	0.6387993
petD	153	Photosystem	0.9262201
psaA	701	photosystem I	0.7952582
psaB	674	photosystem I	0.7350496
psbA	338	photosystem II	0.5143546
psbB	476	photosystem II	0.9264860
rpl16 *	125	large subunit rRNA	1.0934016
rpl2 *	239	large subunit rRNA	1.3604424
rpoB *	482	RNA polymerase	1.1817668
rpoC1 *	251	RNA polymerase	0.9960589
rps2 *	196	small subunit rRNA	1.7945407
rps3 *	141	small subunit rRNA	1.4573229

Table 4.4: Function and sequence length of 14 protein sequences from the chloroplast. Sequences marked with * are involved in translation and transcription.

We can use a clustering algorithm to group the proteins. Figure 4.5 shows the different trees obtained from clustering the genes into 2 classes which differ in both the topology and edge weights. When using the k-cluster algorithm to divide the genes into two classes, all genes which are involved in translation and transcription cluster together (see table 4.4). The trees representing the two clusters are distinct. There are some changes in the topology, some short edges differ, but more importantly we observe a difference in the external edges of the trees (see figure 4.5). For example in cluster a),

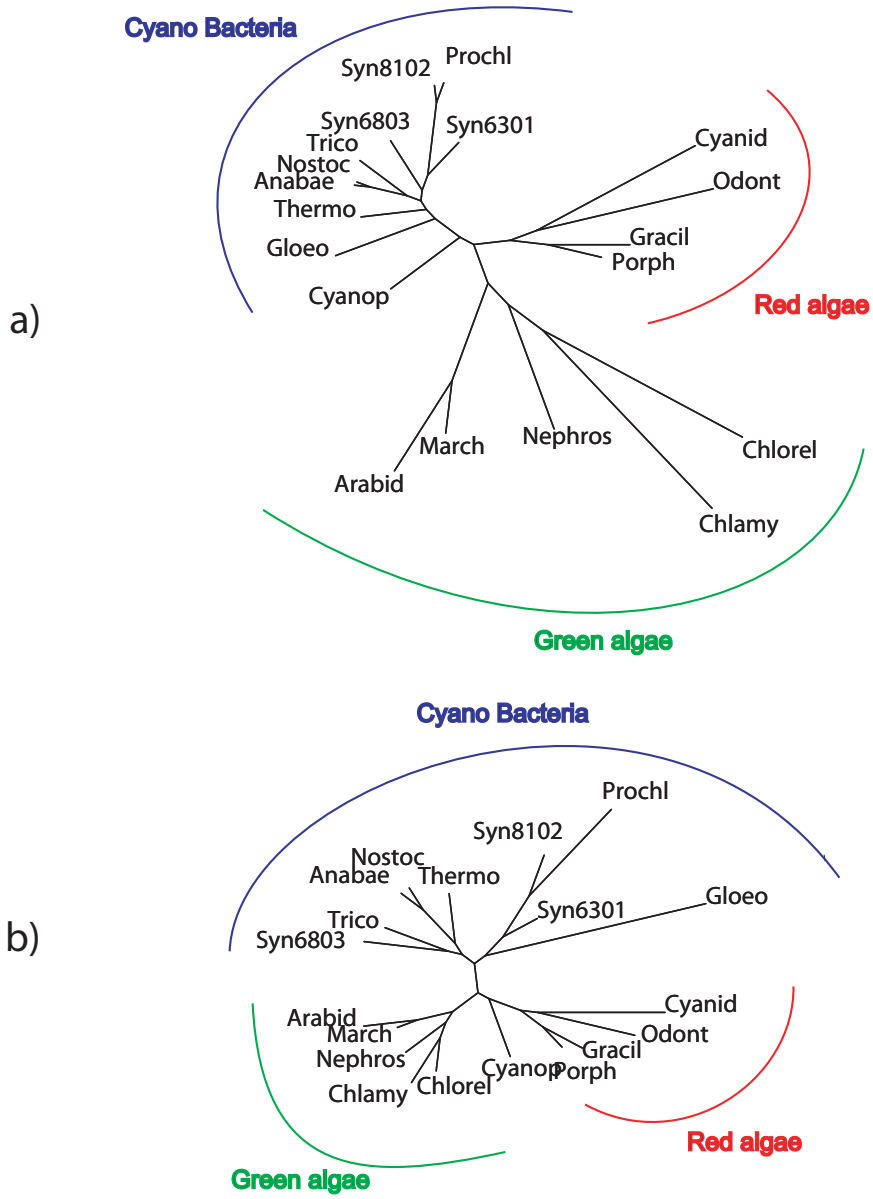


Figure 4.5: Trees for the two main classes found by the k-cluster algorithm. The clustering separates all genes which the involved into translation and transcription a) (see table 4.4) from all other genes b). The edge lengths leading towards the the green algae differ considerable between the two trees.

the edges leading towards the green algae in the transcription and translation proteins are relatively long while in the other cluster b) they are relatively short. This implies that there has been some change in the translation and transcription in the green algae. It would be good in future to test what genes follow this pattern. The k-cluster method has some disadvantages when we use more clusters. However, since we have only 14 genes to distribute into k clusters, we will often get local optima.

We can also visualise the relationship between the genes based on multidimensional scaling (MDS) of the distance Hadamard spectra for each alignment. In this case we used the distance Hadamard as we have amino acid sequences and we cannot apply the Hadamard conjugation directly as in the example of section 4.2.1. In a first step the pairwise genetic distances between the 19 species for each of the 14 amino acid sequences were computed. Split spectra are derived from the distance matrices for each gene via the distance Hadamard. Most weights of the splits are close to zero and we include only splits in the further analysis, if the split weight exceeds a certain threshold (in this example ≥ 0.002) in at least one of the gene spectra. We used additionally 100 bootstrap samples of each amino acid sequence to obtain the distribution around the spectra. Having constructed the spectra we use pairwise Euclidean distances between all the spectra to determine the dissimilarity of spectra. Figure 4.6 shows the first two principal components of a multidimensional scaling based on the of the fourteen amino acid sequences. Often the genes in the same complex are neighbors, which would support the hypothesis that there is lineage specific evolution of interacting genes. Lockhart et al. (2006) have shown that genes are under different evolutionary constraint in the the different lineages. Lineage specific evolution complicates the estimation of divergence times for the different clades and will also affect the positioning of the root.

4.3 Summary

The k-cluster method does appear to be a very useful development and has been applied to two data sets; one DNA sequences and one protein chloroplast.

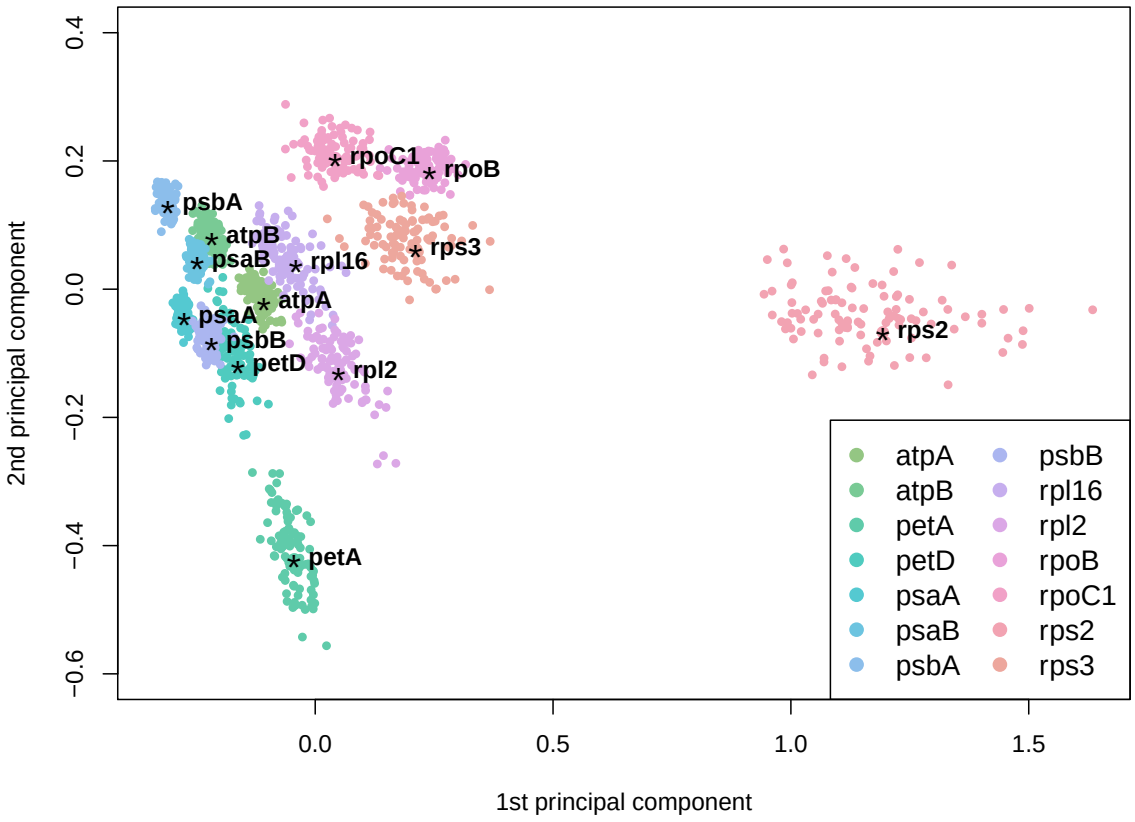


Figure 4.6: First two principal components of a multidimensional scaling based on edge spectra of fourteen amino acid sequences from 7 complexes. The complexes have the the same color and most of them cluster together. The first principal component is influenced by the different tree topology the the different complexes would choose. Each dot represents one of the bootstrap samples.

The main difficulty with this method is choosing number of classes. For the yeast dataset with 106 genes 6-8 clusters are optimal regarding the BIC, whereas the AIC would suggest 18-20 clusters. The Shimodaira-Hasegawa test also tends to chose the number of clusters closer to the BIC for this dataset.

The Shimodaira-Hasegawa test and the GO-analysis, are multiple testing problems, and both lose power when the number of clusters increases. In addition the GO-analysis is adapted from micro-array analyses and the micro-arrays often contain 5,000-30,000 different genes. But genomic datasets of this order are likely to be available for phylogenetic analysis soon.

For the chloroplast dataset, the k-cluster separated the genes into two set of genes, where one set contained all genes which are involved in translation and transcription.

Furthermore the Hadamard conjugation in combination with multidimensional scaling offers a possibility to visualise the relationship between genes without imposing a specific tree. However this method is restricted to datasets with a low number of species.

Chapter 5

Penalized least-squares and phylogenetic networks

In order to describe phylogenetic data as realistically as possible, complex models such as partitioning or mixture models, which need many parameters, are often applied. With the availability of concatenated data from several hundred genes and for reasons outlined in the previous chapter, model selection becomes an important issue in phylogenetic analysis. Introducing additional parameters to a model will increase the likelihood and decreases the bias, but can lead to overfitting the data. Additional parameters will also increase the variance of the parameter estimates (Kelchner and Thomas, 2006). Penalised Maximum Likelihood (PML) is a method which addresses the problem of a bias-variance trade-off (figure 5.1). PML is commonly used for spline regression and variable selection (Simonoff (1996); Hastie and Tibshirani (1990); Hastie et al. (2001); Miller (2002)). In phylogenetics, Sanderson (2002) introduced PML for rate smoothing. PML offers two advantages over ordinary maximum likelihood. Generally penalized likelihood estimates have a lower variance than estimates from ordinary maximum likelihood, but introduce a bias. PML can lead to a higher prediction accuracy and the estimates will then have a lower AIC or BIC than the unpenalized version. PML also offers a framework for variable selection, especially using *Least Absolute Shrinkage and Selection Operator* (LASSO) (Tibshirani, 1996) and *Least Angle Regression* (LARS) (Efron et al., 2004) which we describe below (equation 6.3), and which allows an easier interpretation of the phylogenetic model.

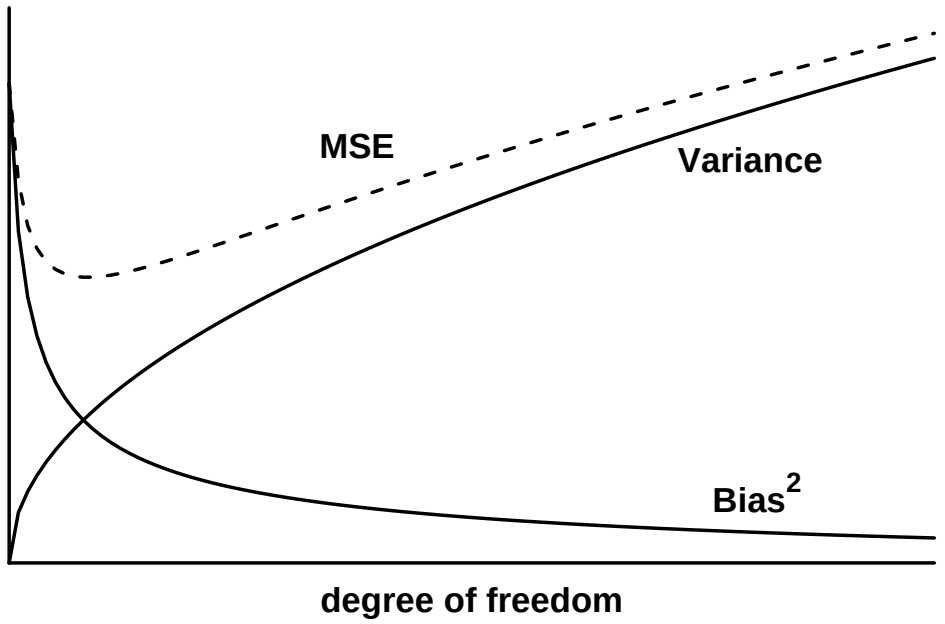


Figure 5.1: Bias-Variance trade-off. This example shows the bias decreasing and variance increasing as the complexity of the model grows. The dotted line indicates the prediction accuracy of the estimates.

The framework of penalized likelihood is very general and we will start by introducing some potential applications for phylogenetic networks. This allows the introduction of the LASSO and ridge estimators in a least-squares framework. Later on I will show applications of both methods for phylogenetic mixture and partition models in the more general likelihood framework. As far as I am aware, LASSO and LARS methods have not been used previously in phylogenetics.

5.1 Background

In this chapter I will introduce ridge regression, LASSO and LARS. I will also give a short revision how to present design matrices for rooted and unrooted trees. We will build on these to introduce some new network algorithms. We will also show close relationships between networks, the distance Hadamard and LARS algorithm.

5.1.1 Overview of distance based methods

Before I describe the new applications I will give an short overview of distance based methods. Least-square (Least Squares (LS)) methods were some of the first phylogenetic methods used. They follow the model

$$d = X\beta + \epsilon \tag{5.1}$$

where d is a vector of $k = \binom{n}{2}$ pairwise distances between two taxa, n is the number of taxa, $X = (x_{ij})$ is a matrix, and β are the edge weight parameters and ϵ is an error term to be minimised. We will call X the *design matrix*, where the columns correspond to the splits in a tree or network. There are $2^{n-1} - 1$ different splits possible, but they will not usually all be included in X. The objective is to find a vector β that minimizes the sum of squared residuals

$$\min_{\beta}(d - X\beta)^T(d - X\beta) \tag{5.2}$$

We can estimate edge weights $\hat{\beta}$ with a least-square method

$$\hat{\beta} = (X^T X)^{-1} X^T d \tag{5.3}$$

Let d_{AB} be the distance between taxa A and taxa B, then the design matrix X for an unrooted tree is

$$x_{AB,j} = \begin{cases} 1 & \text{if } A, B \text{ are separated by split } j \\ 0 & \text{else} \end{cases}$$

We can also estimate split networks if we add further columns which correspond to the additional splits of the network. Table 5.1 presents an example of the design matrices for the unrooted and rooted trees of figure 5.2 and a splits network with 5 taxa.

For the ultrametric case, the design matrix can be written in a similar form. Table 5.2 a) gives an example of an design matrix for the ultrametric tree in figure 5.2 b). The weights of the external edges can be easily computed using this representation, e.g. the edge leading to taxa A (or B) is $e_0 - (e_1 + e_2 + e_3)$. Table 5.2 b) shows a

	Tree							*additional splits*							
	A|BCDE	B|ACDE	C|ABDE	D|ABCE	ABCD|E	AB|CDE	ABC|DE	AC|BDE	AD|BCE	AE|BCD	BC|ADE	BD|ACE	BE|ACD	CD|ABE	CE|ABD
AB	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0
AC	1	0	1	0	0	1	0	0	1	1	1	0	0	1	1
AD	1	0	0	1	0	1	1	1	0	1	0	1	0	1	0
AE	1	0	0	0	1	1	1	1	1	0	0	0	1	0	1
BC	0	1	1	0	0	1	0	1	0	0	0	1	1	1	1
BD	0	1	0	1	0	1	1	0	1	0	1	0	1	1	0
BE	0	1	0	0	1	1	1	0	0	1	1	1	0	0	1
CD	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1
CE	0	0	1	0	1	0	1	1	0	1	1	0	1	1	0
DE	0	0	0	1	1	0	0	0	1	1	0	1	1	1	1

Table 5.1: Design matrix for an unrooted tree or a splits network on 5 taxa. Each column corresponds to a split in a tree or network. Each row corresponds to a path between two species/ taxa The first seven columns represent the splits of the unrooted tree in figure 5.2 a). The remaining columns represent all the other internal splits which could additionally appear in a splits network.

a)

	e_0	e_1	e_2	e_3
AB	2	-2	-2	-2
AC	2	0	-2	-2
AD	2	0	0	-2
AE	2	0	0	0
BC	2	0	-2	-2
BD	2	0	0	-2
BE	2	0	0	0
CD	2	0	0	-2
CE	2	0	0	0
DE	2	0	0	0

b)

	e_0	e_1	e_2	e_3
A|BCDE	1	-1	-1	-1
B|ACDE	1	-1	-1	-1
C|ABDE	1	0	-1	-1
D|ABCE	1	0	-1	-1
ABCD|E	1	0	0	1
AB|CDE	0	1	0	0
ABC|DE	0	0	1	0

Table 5.2: a) Design matrix for the rooted tree of figure 5.2 b) and associated b) contrast matrix. There are many different ways to specify the design matrix. The design presented here contains in the first column the distance between the root and the tips and in the other columns the internal edges.

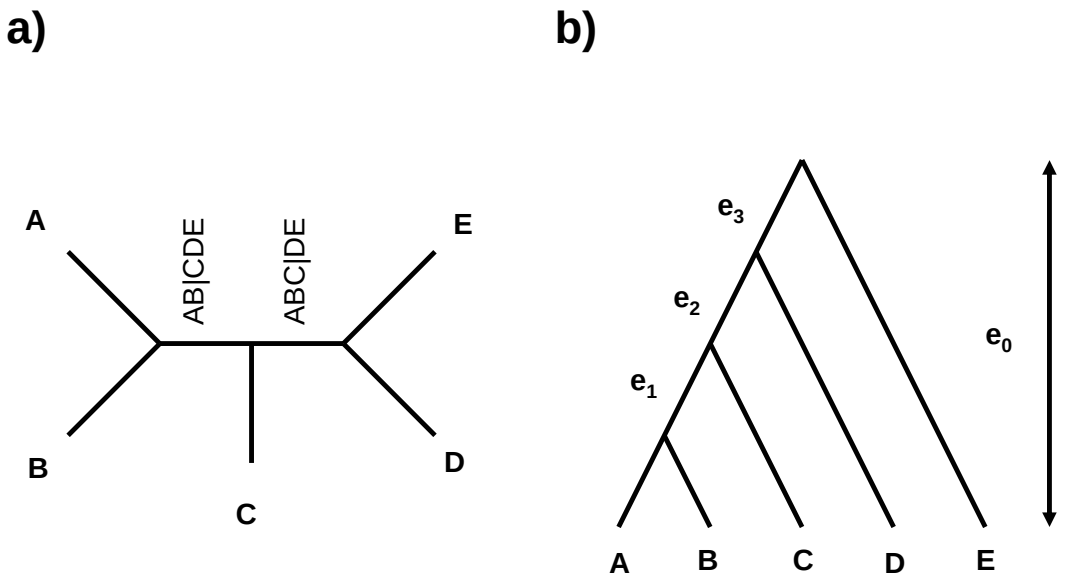


Figure 5.2: a) shows an unrooted tree with 5 taxa, b) a rooted tree with the same topology.

contrast matrix, which can be used to compute all the edge lengths. There are many equivalent parameterisations possible, but this one is easy to describe and implement for our purpose.

The ordinary least-squares method assumes the errors are identical and normally distributed $N(0, \sigma^2)$. For phylogentic analysis this assumption is not realistic. The variance of d depends on the distances, for the Jukes-Cantor model the variance of the distance is given

$$\mathrm{Var}(d) = \frac{3}{16}\left(e^{\frac{4}{3}d} + 3\right)\left(e^{\frac{4}{3}d} - 1\right) \tag{5.4}$$

(Felsenstein, 2004). To avoid this assumption, there have been several weighting schemes proposed that can be fitted with a weighted least-squares (Weighted Least Squares (WLS)) approach

$$\hat{\beta} = (X^T W X)^{-1} X^T W d \tag{5.5}$$

where W is a known positive definite matrix. When the identity matrix is chosen for

W, equation (5.5) simplifies to ordinary least-squares (5.3). Often W is a (diagonal) matrix containing weights accounting for the variances of the pairwise distances, as for example is given in equation (5.4). The edge weight estimates of several popular distance based methods are estimated or can be expressed by equation (5.3). Examples are the distance method of Fitch and Margoliash (1967), UPGMA (Unweigthed Pair-Group Mean Average), WPGMA (Weighted Pair-Group Mean Average) (Sneath and Sokal, 1973), UNJ (Gascuel, 1997b), GLS (Bulmer, 1991) and fastME (Desper and Gascuel, 2004, 2005). For NJ (Saitou and Nei, 1987; Studier and Keppler, 1988) the weights of the WLS are so far unknown. Gascuel and Steel (2006) assume that NJ is an approximation of the minimum evolution criterion; the weights in this case would be close to those of the fastME or WPGMA methods. I observed that the edge weights agree for caterpillar trees and this needs to be followed up. Furthermore least-squares measures can be used to estimate the weights of splits for a phylogenetic network. For example Neighbor-Net (Bryant and Moulton, 2004) uses least-squares to determine the weights of the splits. The difference between the methods is mainly in the choice of the design matrix X, or the choice of the weight matrix W. Table 5.3 gives an overview of how X and W are defined for several popular distance methods. Many of the methods avoid the matrix computation of equation (5.5) and some details are given in chapter 7. The methods mentioned have in common that they build the tree (or

Method	Design matrix	weights
Fitch	Unrooted	$1/\mathrm{Var}(d_{ij})$
UPGMA	Rooted	Identity matrix
WPGMA	Rooted	$2^{-w_{ij}}$
NJ	Unrooted	unknown
UNJ	Unrooted	Identity matrix
fastME	Unrooted	$2^{-w_{ij}}$
Neighbor-Net	Network	$1/\mathrm{Var}(d_{ij})$

Table 5.3: Least-squares representation for different distance methods. w_{ij} is the path length between tips i and j.

network) agglomeratively. FastME additionally performs some tree rearrangements to find better solutions. The agglomerative approach always joins two tips or branches

and joins then into a new branch. The disadvantage of this approach is that when two branches are joined all further steps depend on this decision. The weights w_{ij} for a bifurcating tree can easily be derived from the design matrix in table 5.1. The weights are the sums of each row for the design matrix of the unrooted tree. Semple and Steel (2004) described how to get weights if the tree is not bifurcating. The weights w_{ij} depend on the topology, but agree for rooted and unrooted trees (WPGMA and fastME) on the same topology.

The relationships between UPGMA, WPGMA, FASTME and weighted least-squares are not well known nor used. An important benefit of least-squares is that we can apply Analysis of Variance (ANOVA) to get an indication of how well splits are supported, without using time consuming resampling methods like bootstrapping or cross-validation.

Let us assume we have a high number of candidate splits for a network. This can be all possible splits for a network with a low number of tips (say ≈20) or we may want to represent splits from bootstrap sample of trees. We can solve equation (5.5) only if the matrix $X^T W X$ is non-singular, and a necessary, but not sufficient condition, is that the number of rows of the design matrix X is larger than the number of columns. However there are $2^{n-1}-1$ possible splits, but only $n(n-1)/2$ pairwise distances, for a network with n tips. So the design matrix X will be of dimension $n(n-1)/2 \times (2^{n-1}-1)$. In the next section we will introduce ridge regression to overcome this problem.

5.1.2 Ridge regression

To ensure that $X^T X$ is non-singular we can add additional rows to the design matrix X and the distance or response vector d in the following way:

$$\tilde{d} = \begin{pmatrix} d \\ \breve{d} \end{pmatrix}, \qquad \tilde{X} = \begin{pmatrix} X \\ \breve{X} \end{pmatrix} \tag{5.6}$$

The ridge regression uses this trick and expands the model in the following way:

$$\tilde{d} = \begin{pmatrix} d \\ 0 \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_k \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \tilde{X} = \begin{pmatrix} X \\ \lambda^{\frac{1}{2}} I \end{pmatrix} = \begin{pmatrix} x_{11} & & \cdots & & x_{1l} \\ \vdots & & & & \vdots \\ x_{k1} & & \cdots & & x_{kl} \\ \lambda^{\frac{1}{2}} & 0 & \cdots & & 0 \\ 0 & \lambda^{\frac{1}{2}} & & & \\ \vdots & & \ddots & & \vdots \\ & & & \lambda^{\frac{1}{2}} & 0 \\ 0 & & \cdots & 0 & \lambda^{\frac{1}{2}} \end{pmatrix} \tag{5.7}$$

where I is an identity matrix of appropriate dimension. We find that the matrix $\tilde{X}$ has now full rank l. For $\lambda > 0$ we can now estimate all splits simultaneously, but these are biased towards 0. Inserting $\tilde{X}$ and $\tilde{d}$ leads to the classical derivation of the ridge estimator of Hoerl and Kennard (1970).

$$\hat{\beta}_{ridge} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{d} = (X^T X + \lambda I)^{-1} X^T d \tag{5.8}$$

The ridge estimator minimises the following penalized sum of squares criterion

$$\min_{\beta} (d - X\beta)^T (d - X\beta) + \lambda \|\beta\|_2^2 \tag{5.9}$$

The ridge estimator has two main advantages. The first is that it is ensured that $(X^T X)$ is non-singular even if X has more columns than rows, and secondly the variance of the ridge estimator is lower than that of a least-squares solution. However, the ridge estimator is biased, forcing the coefficients towards zero. It has been shown that there always exist some values of λ for which the mean square error of the ridge estimator is smaller than that for the unbiased least-squares estimator (Rao and Toutenburg, 1995) (see figure 5.1 for a schematic presentation). But the higher the value of λ, the higher the bias of the estimates towards zero, although the parameters will stay different from zero.

5.1.3 The LASSO

The LASSO (Tibshirani, 1996; Osborne et al., 2000) is closely related to the ridge regression. Both methods differ only in the penalising term, whereas the ridge regression uses an L_2 constraint on the parameters (see equation 5.9), the LASSO has an L_1 constraint:

$$\min_{\beta}(d - X\beta)^T(d - X\beta) + \gamma \left\|\beta\right\|_1 \tag{5.10}$$

where γ is a penalty parameter that shrinks the parameters β towards zero. The behaviour of the shrinkage is clearer if we use an equivalent presentation of equation (5.10)

$$\min_{\beta}(d - X\beta)^T(d - X\beta) \text{ respect to } \left\|\beta\right\|_1 \leq k, k \geq 0 \tag{5.11}$$

where a value of k can always be found which corresponds to a certain value of γ in equation (5.10). If k is large enough we get the (non-penalised) least-squares estimate, for smaller k some of the β may become zero and these can be omitted from the model. In contrast to the ridge regression where parameters may be small but not zero, here parameters are often exactly zero. This means that we can use the LASSO to perform variable selection.

Zou and Hastie (2005) introduced a generalisation of both the ridge regression and the LASSO they called the *elastic net*.

$$\min_{\beta}(d - X\beta)^T(d - X\beta) + \lambda \left\|\beta\right\|_2^2 \text{ respect to } \left\|\lambda\right\|_1 \leq k, k \geq 0 \tag{5.12}$$

It combines the advantages of the L_1 and L_2 constraints; data with many parameters can be used like the ridge regression and also have automatic variable selection in form of the LASSO. Zou and Hastie (2005) used the elastic net to the select genes from microarray data.

Efron et al. (2004) presented an algorithm they called Least Angle Regression (LARS) which is an efficient way to describe the path of the variable selection under an L_1 constraint. This algorithm just evaluates the points where a new variable

enters the model (see figure 5.3) based on the correlation between the columns of the design matrix and the residuals. The covariates need to be scaled for this algorithm. This avoids computing many solutions to find an optimal solution. To get a whole path like in figure 5.3, we then only have to evaluate the model each time we include a new parameter. Using this technique can speed up the algorithm dramatically (Efron et al., 2004).

5.2 Methods

5.2.1 Constructing phylogenetic network using the LASSO

I have implemented a version of the elastic net to estimate edge weights of a phylogenetic network given some distance data. The implementation differs from Zou and Hastie (2005) in that it allows for characteristic features of phylogenetic analysis. The parameter values for the edge weights are restricted to non-negative values and I do not impose any constraints on the external edges leading to the tips. Negative edges cannot be interpreted meaningfully, and also can lead to artifacts. If two parameters are highly correlated it can happen that when one becomes negative, the other becomes positive to cancel the effect out. Adding no constraints prevents the network from becoming degenerate.

Algorithm 5.1 describes the elastic net as it is implemented in phangorn.

We chose the ridge penalty λ to be small (10^{-3}) to ensures that no singularities occur during the fitting process. The recomputing of the weights in step 3 without the LASSO penalty leads to less biased estimates of the split weights. The splits which correspond to external edges are estimated without constraints. This ensures that they are always selected into the model.

The design matrix X has $\binom{n}{2}$ rows, and up to $2^{n-1} - 1$ columns - if we assume all possible splits. However with prior knowledge about the organism can reduce the number of possible splits dramatically, for example when we know that a set of species belong to monophyletic group or the out-group. With constraints of this kind this method becomes applicable for larger data sets.

Algorithm 5.1 Algorithm to construct splits network using the

1. `Input: a vector of pairwise distances and design matrix containing all allowed splits, similar to table 5.1`

2. `Minimise`
$$\sum_{i-1}^{\binom{n}{2}} w_i(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\|\beta\|_2^2$$
`under the constraints` $\|\beta\|_1 \leq k, k \geq 0, \beta_j \geq 0, j = 1, \ldots, p$ `and weights` w_i `for example the inverse of the variances of the distances (see table5.3). Edges leading towards the taxa not under the penalty constraints, but must also be non-negative.`

3. `Recompute splits weights for the subset of splits which has weight with non-negative (weighted) least-squares for all splits greater 0, without the LASSO constraint.`

4. `Return splits and weights of the network`

In contrast to the NeighborNet (Bryant and Moulton, 2004), this method will not return a circular split system, so has not the advantage that it always can be visualised as a planar graph.

5.2.2 Distance Hadamard

The second method to construct split networks is based on the distance Hadamard method and uses relationship with the LARS algorithm.

Hendy and Penny (1993) introduced the distance Hadamard, a method which uses a Hadamard matrix as a design matrix (see section 1.4.2). Here also the distance matrix is expanded, here to a quadratic matrix 5.1.

$$-2d = Hq \tag{5.13}$$

where H, is the Hadamard matrix, is of dimension $2^{n-1} \times 2^{n-1}$, and additional distances are added to fill the distance vector d. The Hadamard matrix has a slightly different coding of the variables. All elements in the design matrix H correspond to an element in X and are transformed by 1-2x_{ij}. Formula 5.13 represents a saturated model, therefore

no error term appears. We will use the substitution $y = -2d$.

Our target is again to find a subset of positive split weights q which minimises the following LASSO constraint:

$$(y - Hq)^t(y - Hq) + \lambda\|q\|_1, \lambda \geq 0, q_i \geq 0, \quad i = 2, \ldots, 2^{n-1} \tag{5.14}$$

The will show that we can use the the relationship to LARS algorithms Efron et al. (2004) to get the solutions. To apply the LARS algorithm, the response vector y has to be centered to have mean 0

$$\tilde{y} = y - \mu, \qquad \mu = \frac{1}{m}\sum_{i=1}^{m} y_i, \quad m = 2^{n-1} \tag{5.15}$$

The LARS requires further that all the covariates have been standardized. From the construction of the Sylvester matrix H (see formula 1.4) it follows directly that all, but the first column, have mean 0 and equal standard deviation. We can interpret the first column, which consists only of ones, as an intercept and with centering the response vector (equation 5.15) we set this intercept to zero.

The LARS algorithm introduces the covariable with the highest correlation with the response vector $\tilde{y}$ first into the model. Subsequent estimates are chosen that way until the correlation with the next variable is equal. Here we make use of the fact that the columns are orthogonal, as follows

$$Cor(\tilde{y}, H_j) = Cor(\tilde{y} - \epsilon H_i, H_j) \tag{5.16}$$

Let $e_1, \ldots, e_{2^{n-1}}$ be the ordered list of all edge weights from a distance Hadamard. LARS first includes the longest edge into the model. Then at position $e_1 - e_2$ the second longest edge is included. All edges which are included into the model always grow equally (see figure 5.5), as their correlation with response $\tilde{y}$ is identical.

When using the distance Hadamard we can use this argument and just include splits which are larger than a arbitrary threshold or to a certain number of splits, making it useful as an explorative tool. Furthermore we can make use of the fast Hadamard conjugation which avoids storing the Hadamard matrix H.

5.2.3 Choosing the number of splits

If we increase the penalty term γ we are likely increase the number of splits which enter the network.

There a several criteria to find the optimal number of splits. The first is to find the network which minimises Mallows' C_p (Mallows, 1973):

$$C_p = \frac{RSS}{\sigma^2} - (n - 2p) \tag{5.17}$$

where RSS denotes the residual sum of squares, n is the number of observations and p is the number of parameters. When the networks we estimate for different penalties are nested we can use test used in step-wise regression (Miller, 2002, p. 43). Often this will be often the case, but is not guaranteed - when starting with a star tree and slowly increasing the LASSO penalty θ to add additional splits to the network.

5.3 Results

I will demonstrate the elastic net and distance Hadamard for phylogentic networks on the concatenated Yeast dataset (Rokas et al., 2003) which is described in section 1.5.

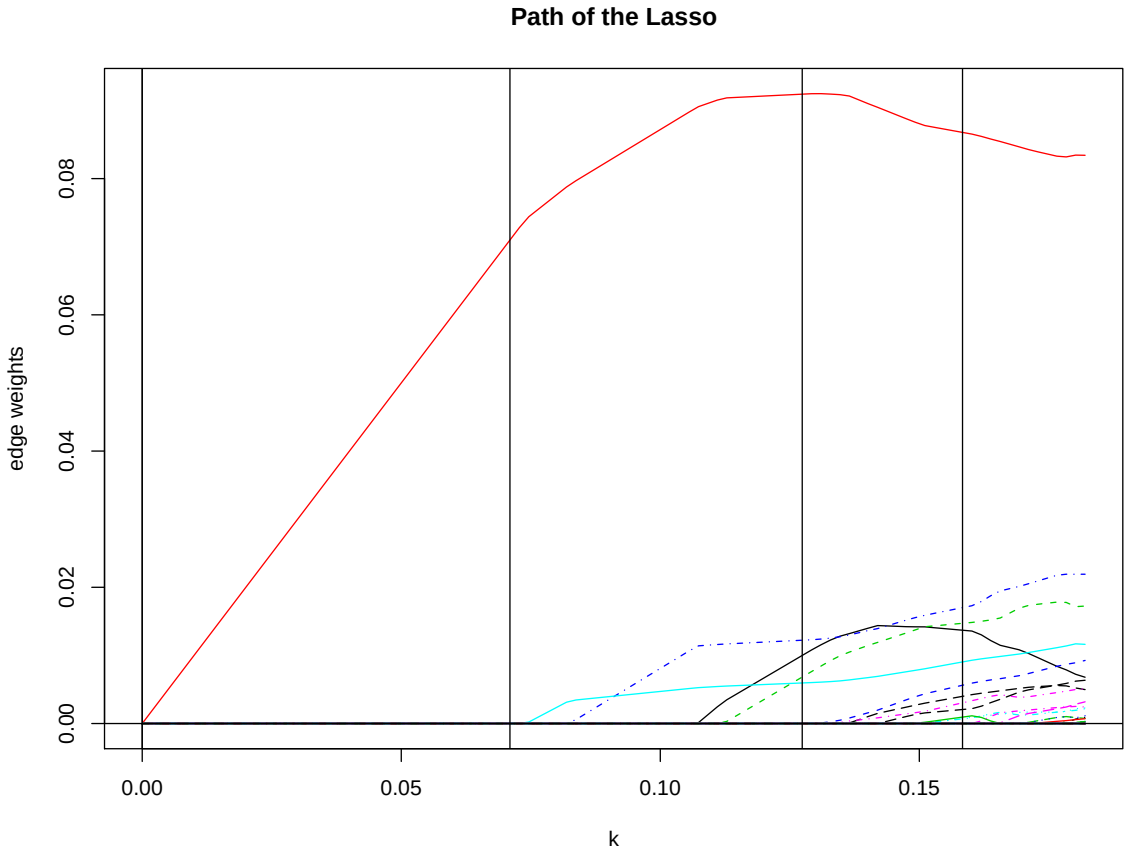


Figure 5.3: A plot of the edge weights of the internal edges depending on the threshold k in equation (5.11). On the right side 28 parameters are included into the model, that is the number of pairwise distances for this dataset with 8 taxa. This is the maximum number of parameters one could include using the LASSO. Using the elastic net it is possible to include more edges, however the number of pairwise distances is already in most applications more edges than one can visualise and interpret easily. The vertical lines indicate where the plots of figure 5.4 where chosen.

Figure 5.3 shows the path of the edge weights included in the network. Only the internal edges are shown, as the external edges are not constrained by the LASSO penalty. On the right hand side of the path 28 parameters are included. That cor-

responds to the number of pairwise distances (28) between 8 taxa. Since we also use a ridge penalty we could add more parameters to the model, however the parameters would start to fluctuate due to the strong correlation between them. The vertical lines in figure 5.3 indicate 4 particular penalties for which networks are plotted in figure 5.4. The external edges are not restricted, so for a value of $k = 0$, the network is a star tree (see figure 5.4 a)). The higher the value of k, the more variables enter the model.

In figure 5.5 the paths for the edge weights when applying the LARS to the Hadamard conjugation are shown. In contrast to figure 5.3 there are no crossings of the paths of the edge weights.

Mallows' C_p has been computed for networks estimated with elastic net and distance Hadamard approach for networks where the number of splits varied from 8 (star tree) up to 20 (see 5.4). The optimal number of splits according to Mallows' C_p are networks

	Elastic Net		Distance Hadamard	
splits	RSS	C_p	RSS	C_p
8	0.07650	7765.318	1.35311	18822.002
9	0.00611	611.083	0.18146	2429.207
10	0.00462	462.208	0.10950	1424.196
11	0.00280	278.738	0.05354	643.212
12	0.00153	151.582	0.02188	202.213
13	0.00059	57.594	0.01657	129.835
14	0.00056	57.005	0.01131	58.298
15	0.00049	51.374	0.00858	22.040
16	0.00026	30.056	0.00813	17.759
17	0.00024	30.205	0.00799	17.781
18	0.00010	18.204	0.00785	17.804
19	0.00009	19.469	0.00777	18.679
20	0.00008	20.000	0.00772	20.000

Table 5.4: RSS and Mallows' C_p the elastic net and distance Hadamard network ranging from 8 to 20 splits. The networks optimising Mallows' C_p are plotted in figure 5.6.

with 16 splits for the distance Hadamard and 18 splits for the elastic net method. These networks are plotted in figure 5.6 and they share 13 of the 16 or 18 splits. However the residual sum of squares (RSS) is very low, so the estimate for the variance in the denominator of equation (5.17) may be unreliable. The number of splits indicated

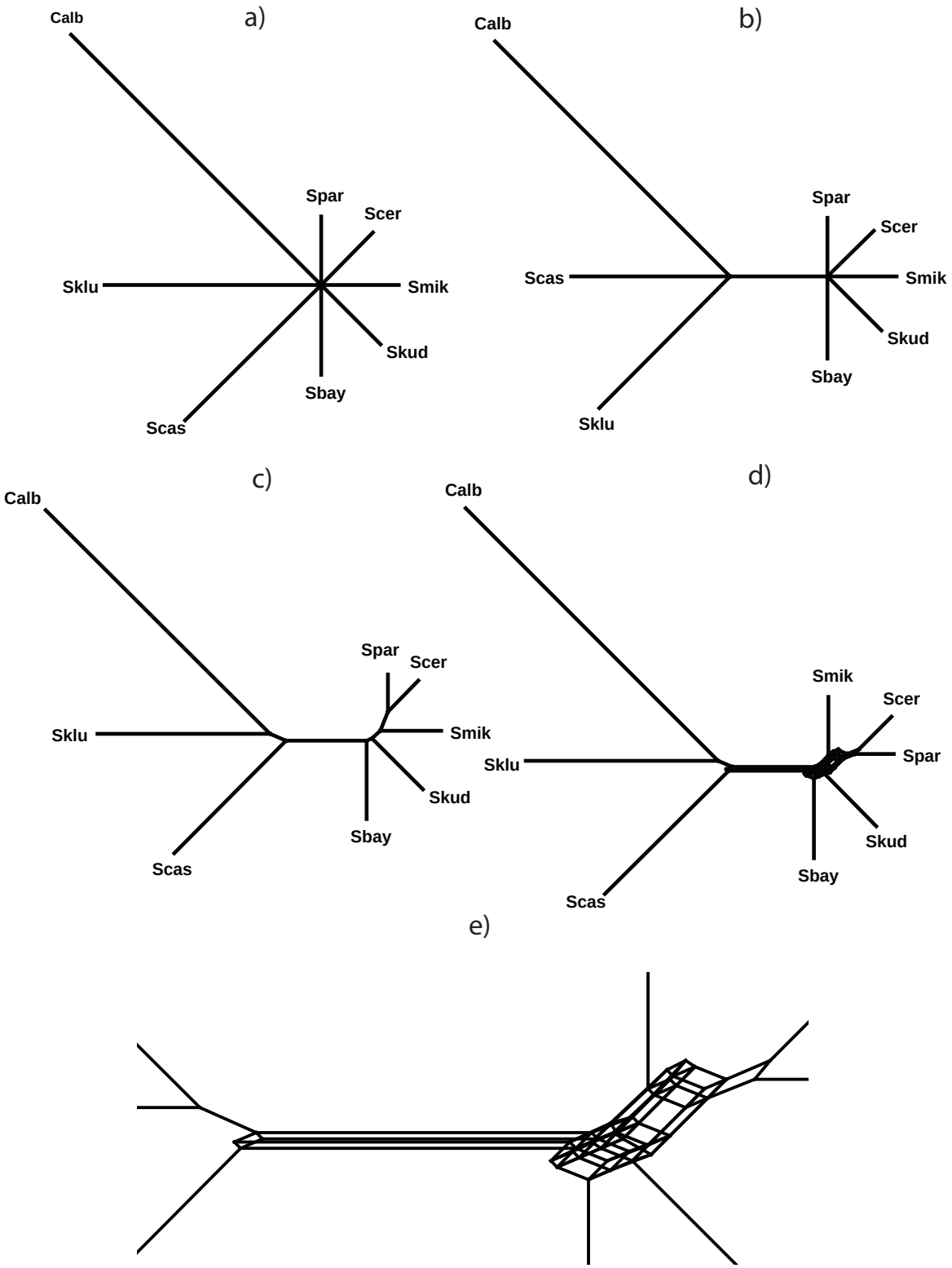


Figure 5.4: Trees and networks for different penalties for the Yeast dataset (Rokas et al., 2003). a) shows the star tree, given the maximal penalty to the internal edge weight parameter. b) to d) show how the relaxation of the penalty leads to additional edges in the network. e) gives a detailed view of the network of d).

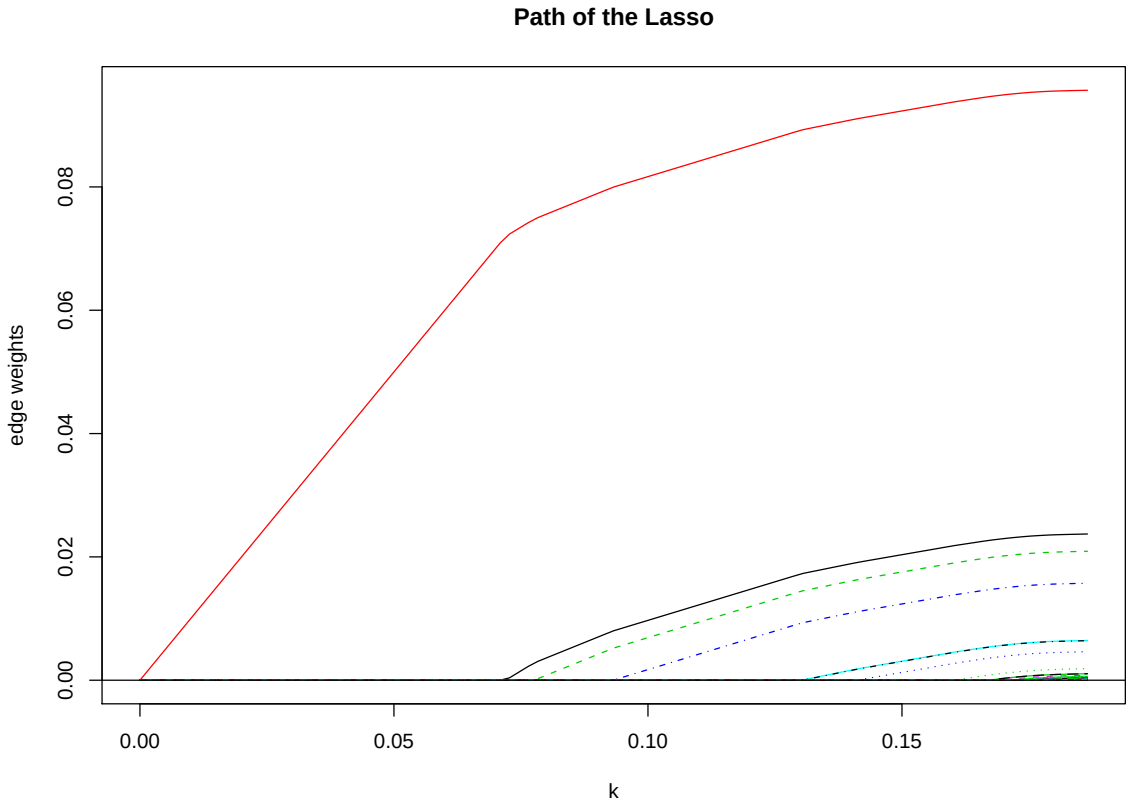


Figure 5.5: The paths of the edge weights for the distance Hadamard. As all parameters are uncorrelated none of the paths crosses.

by Mallows' C_p statistic is for both networks lower than the 21 splits created with NeighborNet.

We can compare the network produced by the elastic net and the distance Hadamard method with the more traditional Neighbor-Net (Bryant and Moulton, 2004) (see figure 5.7). Most noticeble is that the elastic net network contains several splits which cannot represented as planar graph.

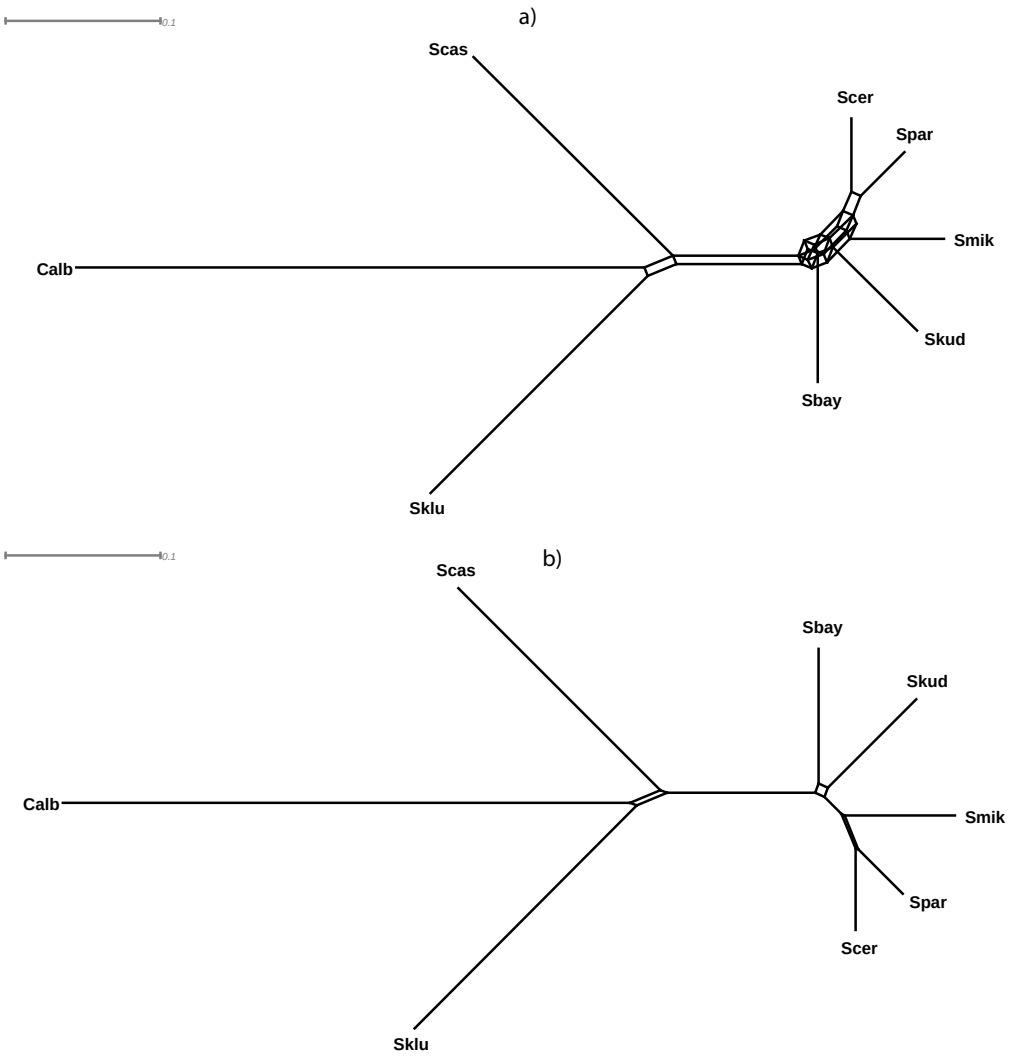


Figure 5.6: Splits graphs estimated using the a) elastic net with 18 splits and b) distance Hadamard with 16 splits as indicated by the Mallows' C_p in table 5.4.

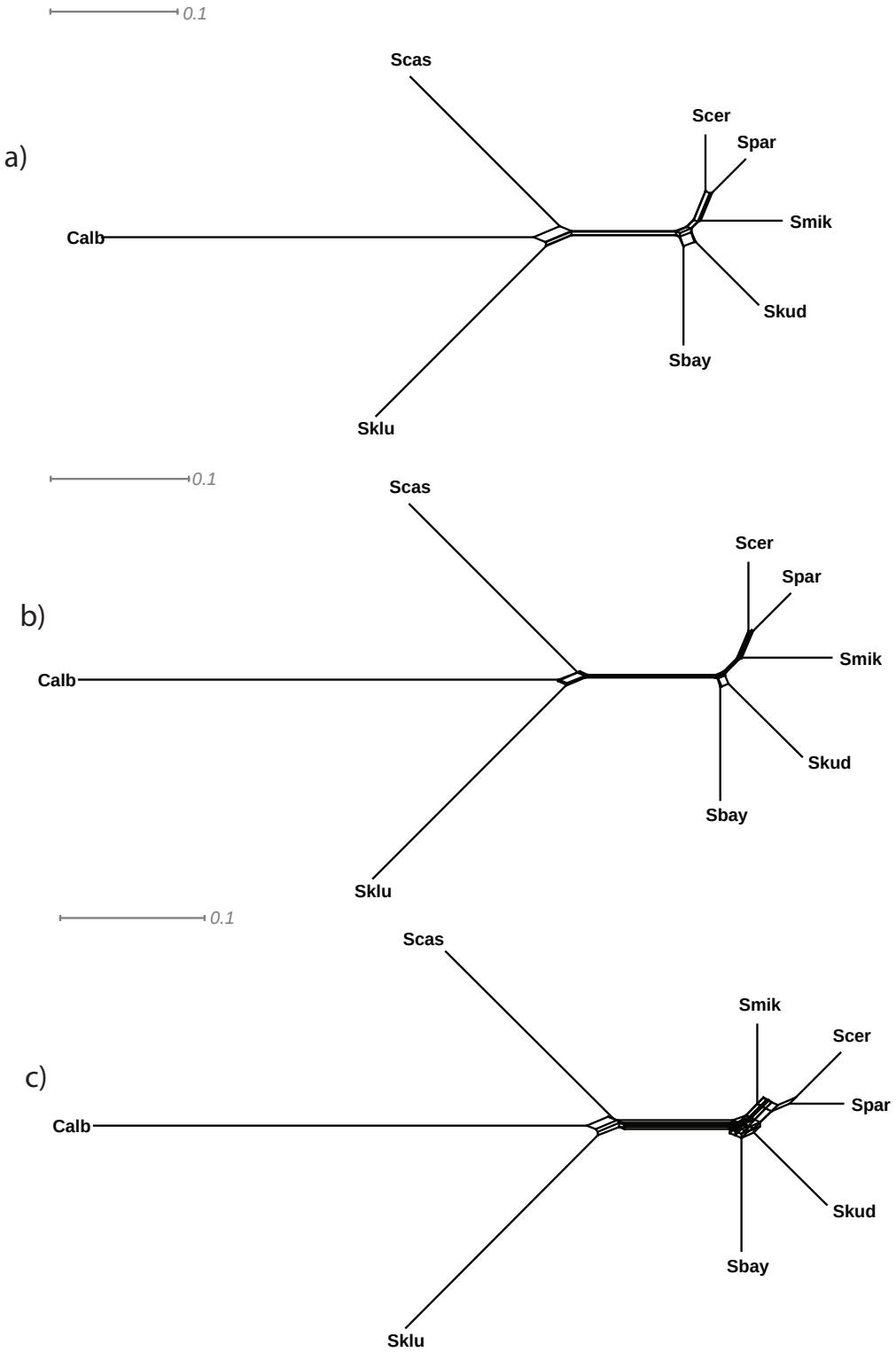


Figure 5.7: Comparison of three splits graphs with 21 splits: a) NeighborNet, b) Distance Hadamard and c) with penalized likelihood.

5.4 Conclusions

I introduced two applications of penalised likelihood, one based on the elastic net and the other on LARS, to estimate split networks. To minimise the bias we use the LASSO for parameter selection and estimate afterwards the parameters without enforcing a penalty on the parameter. Strategies like elastic net and LARS are an interesting alternative for searching the space to estimate phylogenetic networks.

These methods are not restricted in the number of splits or from initial addition of splits, nor do they depend on the edges already included into the network (as for example in Neighbor-Net). It allows the user to select the number of edges, which means the method can be used as explorative tool to describe conflict within phylogenetic data. Most other algorithms return a network with a fixed number of edges. With the elastic net, one can study the continuum from very sparse to highly over-parametrized models. Otherwise the number of edges to include into the model can be guided using Mallows' C_p statistics for least-squares, as an analogous procedure to the AIC in the maximum likelihood framework.

The elastic net method is restricted by the size of the design matrix. Instead of evaluating all possible splits, one could pass a collection of splits, for example from a bootstrap run, to visualise the network. This would allow the application of the LASSO and elastic net to far bigger networks with many more taxa.

It is straight forward to extend the ridge estimator towards weighted least-squares. However the weighting scheme as it is used in WPGMA and fastME depends on the structure of the tree. Thus adding an edge changes the weights; a continuous path as shown in figure 5.3 can not easily be achieved, making it questionable whether a path continuation method like LARS could be applied. Using weights which describe the variances of the distances and stay constant can be included.

There are other potential uses of ridge regression and LASSO in phylogenetics. In in the following chapter I will will apply penalized maximum likelihood with partition models described in chapter 4.

Chapter 6

Penalized ML for phylogenetic partitions

This chapter continues with themes from chapter 4 and 5 in particular, namely partition models from chapter 4 and PML from chapter 5.

6.1 Methods

Now we will use PML to study partition models. Assume we are given a partition model with m genes, where the trees in each partition , which show the same topology, but have different parameter estimates for the edge weights

$$l^{PML}(x, \theta, \lambda) = \sum_{i=1}^{m} l(x_i, \theta_i) - \lambda g(\theta), \quad 0 \leq \lambda \leq \infty \tag{6.1}$$

where $\theta^T = (\theta_1, \theta_2, \ldots, \theta_m)$ is a vector of parameters, θ_i is the parameter vector of the i-th gene, and $\lambda g(\theta)$ is a penalty term, which forces parameters to be similar. This representation is equivalent to optimising the likelihood $\sum_{i=1}^{m} l(x_i, \theta_i)$ with respect to $g(\theta) \leq \epsilon$, where ϵ defines the penalty in a similar way as λ in equation (6.1) (Hastie et al., 2001, p. 59).

For the penalty function a norm of the estimates $\|\theta\|_k$ is often chosen. Depending on the form of the penalty term, we distinguish between ridge estimates (Hoerl and Kennard, 1970)

$$l^{ridge}(\theta, x, \lambda) = l(\theta, x) - \frac{1}{2}\lambda\|\theta\|_2 \tag{6.2}$$

and the LASSO (Tibshirani, 1996)

$$l^{LASSO}(\theta, x, \lambda) = l(\theta, x) - \lambda\|\theta\|_1 \tag{6.3}$$

We will now illustrate the use of penalised likelihood in more detail in the following example.

6.1.1 Example

Assume we have three gene trees as shown in figure 6.1. We are interested in the differences in the edge weights between the different gene trees. We choose a special case for the penalty term (6.2)

$$l^{ridge}(\theta, x, \lambda) = \sum_{i=1}^{3} l(\theta_i, x_i) - \frac{1}{2}\lambda\theta^T K\theta \tag{6.4}$$

where K is a (symmetric) penalty matrix. We will illustrate the form of the penalty matrix in more detail.

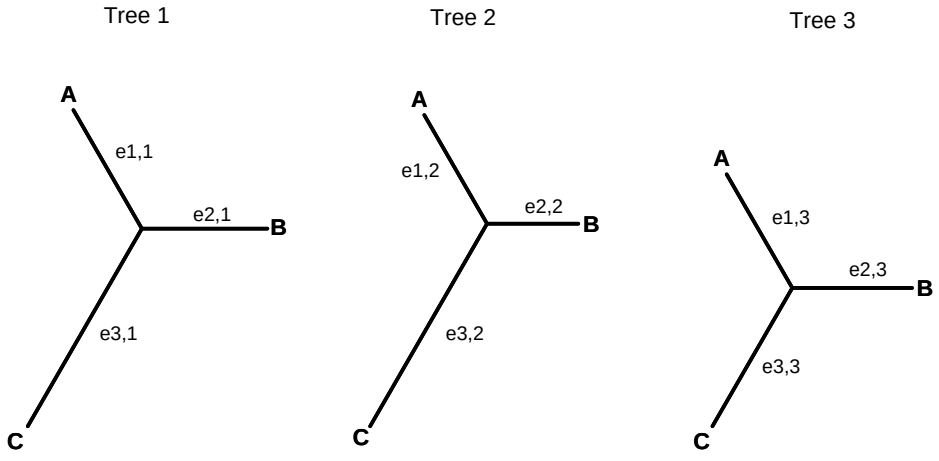


Figure 6.1: Three 3-taxon trees as an example to set up the penalty matrix of equation (6.4).

In the simple case with three edge weight parameters and three partitions as in

figure 6.1, the parameter vector θ the penalty matrix K has the form

$$\theta = \begin{pmatrix} e_{1,1} \\ e_{2,1} \\ e_{3,1} \\ e_{1,2} \\ e_{2,2} \\ e_{3,2} \\ e_{1,3} \\ e_{2,3} \\ e_{3,3} \end{pmatrix} \quad K = \begin{pmatrix} 2 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 2 \end{pmatrix} \tag{6.5}$$

The penalty is controlled by λ only, which works as a bias-variance trade-off. For $\lambda = 0$ the parameter estimates for the different partitions are free to vary without restriction. For $\lambda = \infty$ the estimates θ_1, θ_2, θ_3 are forced to be equal and the estimate is the same as for the concatenated data set. If we choose K as described above, the penalty term is equivalent to $\lambda \sum_{i=1}^{k} \sum_{j>i} \|\theta_i - \theta_j\|^2$. It is possible to specify constraints for penalties using an appropriate choice of the matrix K, as long as it is ensured that K is positive semi-definite.

For the ridge penalty all parameters are included in the model, even when they are close to zero. The advantage of choosing a LASSO penalty over the ridge penalty is that with the LASSO for increasing values of λ, variables get excluded from the model. We can define for the LASSO the partition model with the three trees similar to the ridge estimator of equation (6.4):

$$l^{lasso}(\theta, x, \lambda) = \sum_{i=1}^{3} l(\theta_i, x_i) - \lambda \|\theta^T L\|_1 \tag{6.6}$$

where the penalty matrix L for the trees from figure 6.1 has the form

$$L = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.7}$$

and there exists a nice relationship between the two penalty matrices $K = L^t L$.

When it is possible to center the parameter vector θ, we can simplify the penalty matrix in equation (6.4). Centering corresponds to including an intercept for the edge parameters in the model. The parameter estimates are then the difference from the intercept:

$$\tilde{\theta} = \theta - \bar{\theta} = \begin{pmatrix} \theta_1^{e1} \\ \theta_1^{e2} \\ \theta_1^{e3} \\ \theta_2^{e1} \\ \theta_2^{e2} \\ \theta_2^{e3} \\ \theta_3^{e1} \\ \theta_3^{e2} \\ \theta_3^{e3} \end{pmatrix} - \begin{pmatrix} \bar{\theta}^{e1} \\ \bar{\theta}^{e2} \\ \bar{\theta}^{e3} \\ \bar{\theta}^{e1} \\ \bar{\theta}^{e2} \\ \bar{\theta}^{e3} \\ \bar{\theta}^{e1} \\ \bar{\theta}^{e2} \\ \bar{\theta}^{e3} \end{pmatrix} \qquad \tilde{K} = I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.8}$$

where $\bar{\theta}^i = \frac{1}{k} \sum_{j=1}^{k} \theta_j^i$, and the penalty matrix K simplifies to the identity matrix. We can center the parameter vector using the well-known relationship

$$\sum_{j=1}^{n} \sum_{i=1}^{n} (x_i - x_j)^2 = 2n \sum_{i=1}^{n} (x_i - \bar{x})^2, \qquad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{6.9}$$

The penalised likelihood (equation 6.4) then simplifies to equation (6.2)

$$l(x, \theta, \lambda) = l(x, \theta) - \frac{1}{2}\lambda\tilde{\theta}^T\tilde{\theta} \tag{6.10}$$

$$= l(x, \theta) - \frac{1}{2}\lambda\|\theta\|_2 \tag{6.11}$$

which is a generalisation of the ridge estimator (Hoerl and Kennard (1970)). The original ridge estimator was derived for least-squares, whereas we are using a more general likelihood framework. The motivation behind the ridge estimator was to find solutions when the coefficients in a regression model were highly correlated. We know that the expectation of the negative Hessian matrix (equation 6.14) is semi-definite, which follows directly from the symmetry of the matrix. Adding a positive value on the diagonal makes the matrix positive definite and therefore the inversion in equation (6.14) is always possible. The positive side effect is that the inversion will be numerically more stable and the convergence will be faster as the correlation between parameters is decreased. A further advantage is that the (co-)variances of the ridge estimator θ^{PML} (equation 6.14) are smaller than for the maximum likelihood estimate. However the ridge penalty shrinks all the parameters towards zero - when the estimates are centered - for growing values of λ, which we can see in figure 6.5. Having a high value of λ can introduce a severe bias to the estimates.

6.1.2 Moments of penalised likelihood estimates

The score function and the Hessian matrix for the ridge estimate (formula 6.4) are easy to derive

$$s^{\text{PML}}(\theta) = \frac{\partial l^{\text{PML}}(\theta)}{\partial\theta} = \frac{\partial l(\theta)}{\partial\theta} - \lambda K\theta = s(\theta) - \lambda K\theta \tag{6.12}$$

$$\mathcal{H}^{\text{PML}}(\theta) = \frac{\partial^2 l^{\text{PML}}(\theta)}{\partial\theta\partial\theta} = \frac{\partial^2 l(\theta)}{\partial\theta\partial\theta} - \lambda K = \mathcal{H}(\theta) - \lambda K \tag{6.13}$$

This allows us to optimise θ^{pml} using a Newton-Raphson or similar algorithm. The moments for the LASSO are computationally more complex (Osborne et al., 2000;

Tibshirani, 1996). We can derive the estimator for the variance using the relationship

$$\mathrm{Var}(\theta^{\mathrm{PML}}) = \mathrm{E}\left(-\frac{\partial^2 l^{\mathrm{PML}}(\theta)}{\partial\theta\partial\theta}\right)^{-1} = (-\mathcal{H}(\theta) + \lambda K)^{-1} \tag{6.14}$$

6.1.3 Optimising the penalty

When we optimise a PML model we have to optimise additionally the penalty parameter λ. To compare different likelihood models, the AIC and BIC are often used. To derive these statistics we need to compute the number of degrees of freedom of the PML.

Degrees of freedom

Penalised likelihood chooses parameters more parsimoniously than a maximum likelihood approach. An increasing penalty term in equation (6.1) forces the corresponding parameters to be more similar, and for $\lambda = \infty$ the corresponding parameters will be identical. The number of degrees of freedom for a penalised MLE are clearly lower than for an unrestricted model. Hastie and Tibshirani (1990) give an approximation of the number of independent parameters

$$df = tr\left((-\mathcal{H}(\theta) + \lambda K)^{-1}(-\mathcal{H}(\theta))\right) \tag{6.15}$$

which we will refer to as the effective number of degrees of freedom. Equation (6.15) is a generalisation of the estimate for the effective degrees of freedom of the ridge estimate

$$df(\lambda) = tr(X(X^tX + \lambda I)^{-1}X^t) \tag{6.16}$$

$$= tr((X^tX + \lambda I)^{-1}X^tX) \tag{6.17}$$

as given in Hastie et al. (2001, p. 63). Equation (6.16) can be interpreted as the trace of the projection matrix of a ridge estimator. It is easy to see that for $\lambda = 0$, the number of parameters is equal to the number of parameters in the unrestricted model.

For the LASSO, the degrees of freedom are easier to derive, as we can take the number of non-zero elements of $\hat{\theta}$ as an estimate (Zou et al., 2007).

We can use this approximation for the degrees of freedom to estimate the AIC or

BIC and estimate the optimal value for λ (figure 6.3). The AIC and BIC are defined as follows

$$AIC = -2l(\theta, x) + 2 \cdot df \tag{6.18}$$

$$BIC = -2l(\theta, x) + \ln(n) \cdot df \tag{6.19}$$

where df is the number of degrees of freedom and n is the sample size, which in the phylogenetic context is the number of sites in the alignment.

Alternatively to the AIC or BIC, the penalty can be optimised using a k-fold cross-validation (CV)(Hastie et al., 2001, p. 214). In k-fold cross-validation, the sites of the data are divided randomly into k subsets. Always one of the k subsets is used for validation, and the remaining k-1 subsets are used as training data. The parameters of the PML are optimised on the training data and we compute the likelihood for the estimates on the validation data. The process is then repeated k times, in our case 10 times, so that each of the k subsets of the sites is used once for validation.

6.2 Results

We now consider two datasets, each consisting of four genes (YGL253W, YMR015C, YDR443C, YDR484W and YDR531W, YDR465C, YJR117W, YDL148C) of the yeast dataset. Trees for the individual genes and the concatenated data are plotted in figure 6.2 for each dataset. The edge weights of the gene trees for the first dataset differ considerable, whereas the gene trees for the second are more similar. We estimate a partitioning model for each dataset, with a ridge penalty on the differences of corresponding edge weight estimates.

We infer the PML estimates for different penalties λ, ranging from 10^{-3} to 10^{3}. When parameters are correlated or many parameters are used to specify the model, a small penalty λ often results in a big decrease of the effective degrees of freedom (figure 6.3). This can lead to a strong influence on the BIC and to a lesser extent the AIC. It is easier to interpret the models using the effective degrees of freedom so we will use this interpretation in the following.

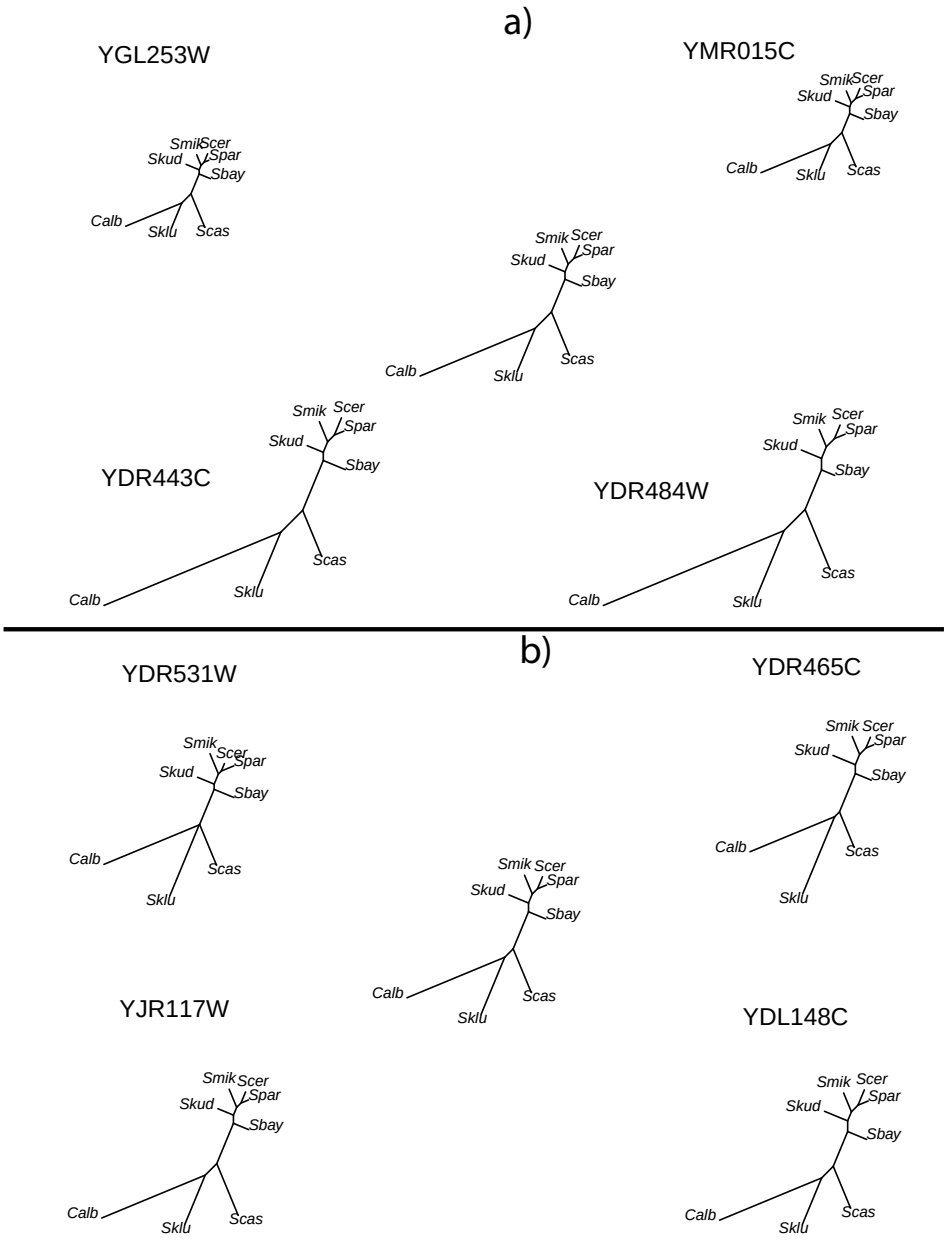


Figure 6.2: In a) the 4 gene trees of of the first partition model are shown, and the tree in the centre has edge weights of the concatenation of the 4 genes. b) shows the trees for the second partition model. The gene trees in a) differ considerably in their edge weights, whereas the edge weights of the gene trees in b) are more similar.

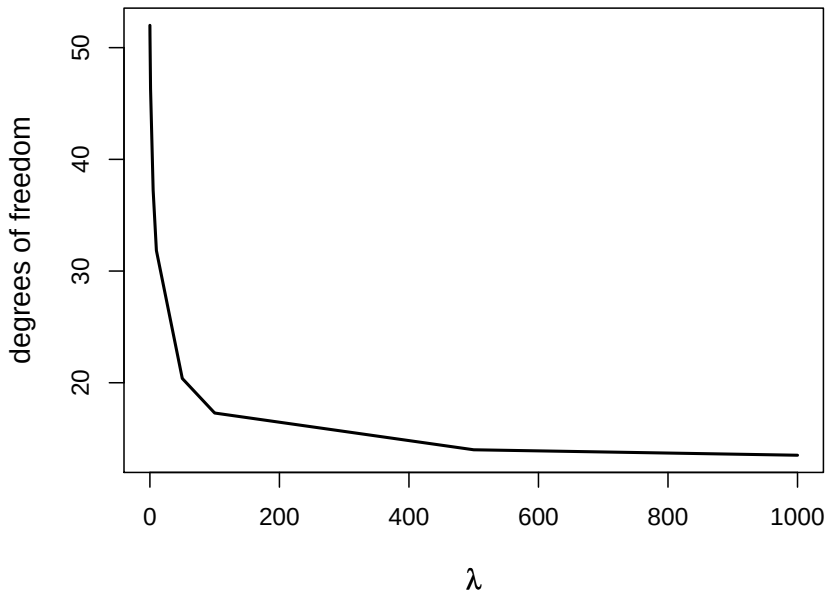


Figure 6.3: The number degrees of freedom, estimated using equation (6.15), are a strict monotone decreasing function of the penalty λ. The degrees of freedom are easier to interpret than the values of λ, which range between 0 and ∞. We will therefore often show plots of other parameters in dependence of the number degrees of freedom instead of λ (e.g. see figure 6.4).

The problem is to find the value of λ which minimises the AIC, BIC or the k-fold cross-validation criterion. Each of the four gene trees have 13 edges, the number of degrees of freedom of the model can vary between $52 = (4 \times 13)$, when there is no penalty added, and 13 for a high penalty, which is equivalent to a concatenation of the four genes. The optimal number of these parameters varies strongly between the AIC and BIC (figure 6.4) and between the two datasets. On the other hand the optimal values for the AIC and the k-fold CV are very similar for each of the datasets. Figure 6.4 shows the likelihood, the AIC, and the BIC depending on the number of degrees of freedom.

For the first dataset the AIC and CV both suggest models with penalty and the effective number of degrees of freedom of between 40 and 50. A high number of pa-

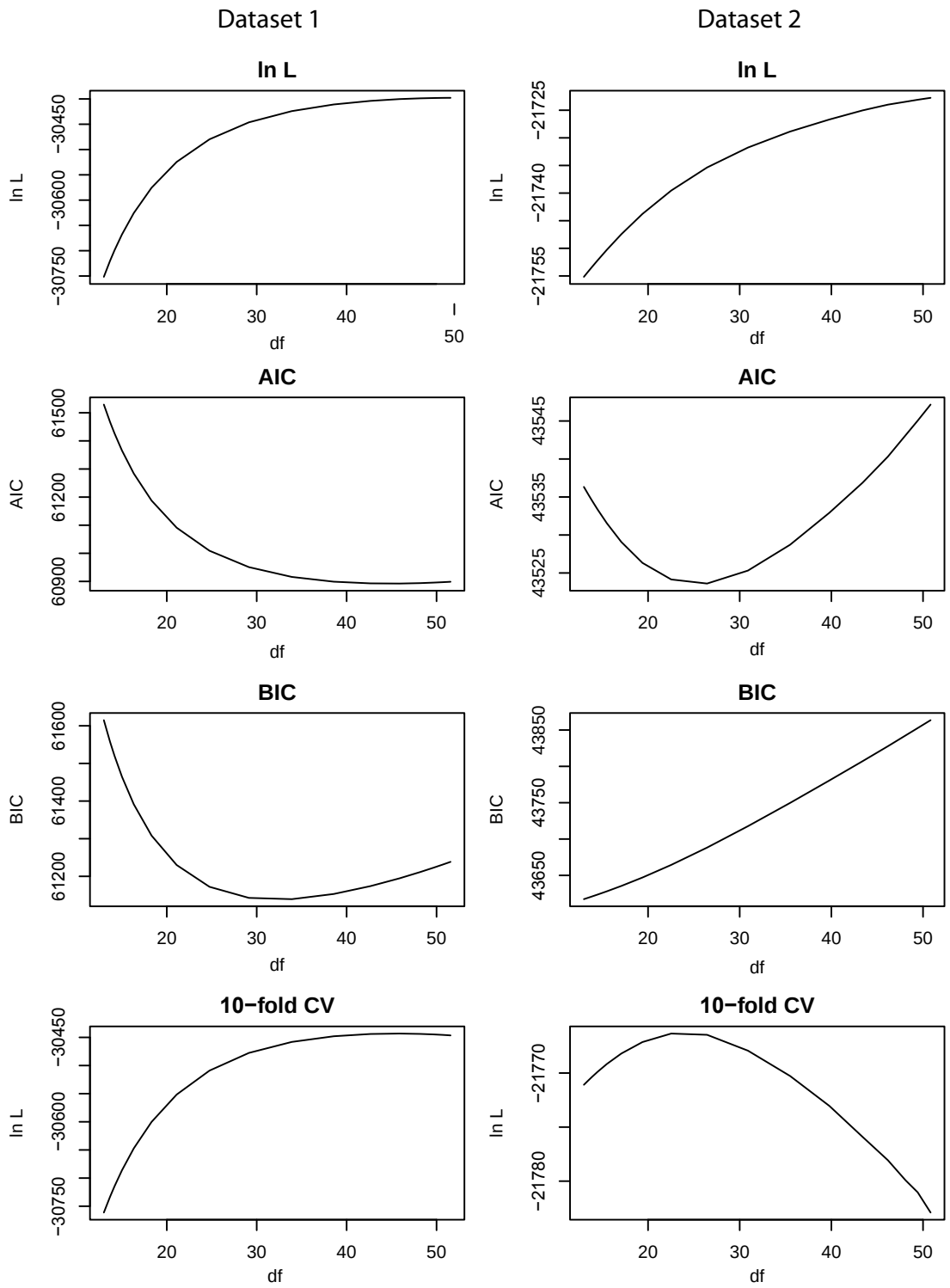


Figure 6.4: The log-likelihood a), Akaikes (AIC) b), Bayesian/Schwarz's c) Information Criteria and the k-fold cross-validation as a function of the approximate number of parameters. The likelihood always increases with the number of parameters. The optimal model according to the BIC criterion is for both datasets more parsimonious than the one optimal according to the AIC or cross-validation.

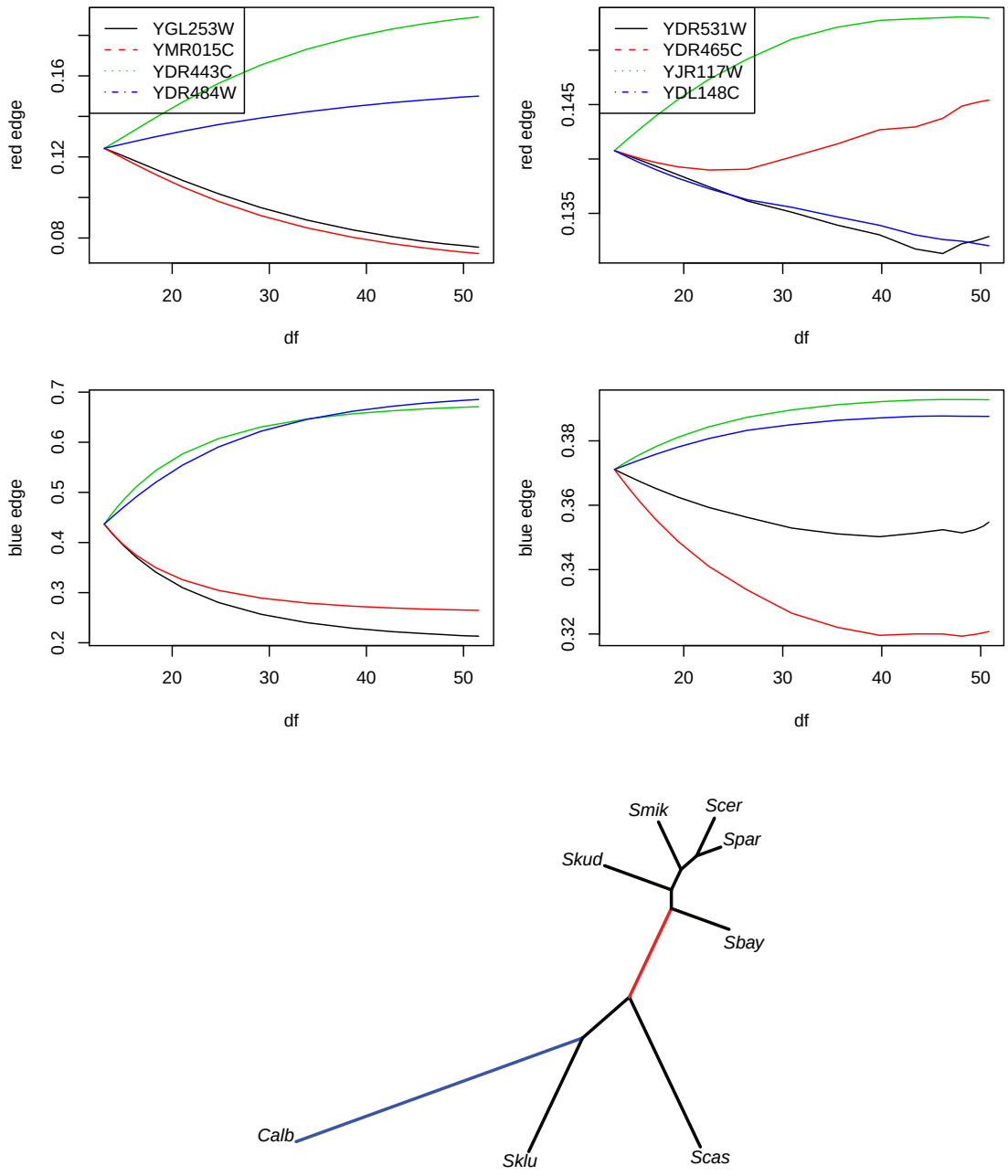


Figure 6.5: Edge weights estimates depending on the number of parameters for the two penalized likelihood models with four genes are shown for the red and blue edges as indicates in the tree below. With increasing number of parameters (decreasing penalty) the parameters diverge.

rameters is expected for this dataset as the individual gene trees differ considerably in their edge weights (see figure 6.2 a)). The BIC suggests a more parsimonious model, which would lead to about 30-35 parameters.

For the second dataset the preferred model, according to the BIC criteria, is the minimal model with only one set of edge weights. The optimum of the degrees of freedom for the AIC and CV is estimated at about twice as high, about 25 parameters.

Figure 6.5 shows the influence of the penalty on the edge weights of the gene trees for two edges.

6.3 Summary

In this chapter we used PML as a bias-variance trade-off for partition models with many parameters. The ridge estimator reduces variance to get robust estimates. Here the PML detected that the gene trees in the two datasets differed in their edge length distribution, and allocated different number of parameters to the models for the datasets.

However we add some computational costs when optimising the penalty value. By providing formulas to estimate the number of degrees of freedom, measures like the AIC and BIC can be evaluated. The optimal model suggested by AIC agrees closely with the one chosen by k-fold cross-validation.

As an alternative the LASSO penalty is advantageous in that it automatically selects variables. The advantage of the ridge regression over the LASSO is that it can be easily incorporated into existing software.

Chapter 7

Biases in hierarchical clustering

Large scale phylogenetic studies are only computationally feasible using hierarchical agglomerative clustering heuristics, typically of $O(n^3)$ for n taxa, but in special cases $O(n^2)$ (Murtagh, 1984), where n can be in the order of hundreds or thousands of taxa. Recent progress has been made even with methods of maximum likelihood, especially the using the program Garli (Zwickl, 2006) and RAxML (Stamatakis, 2006). Existing cubic methods are applied to distance or dissimilarity data, usually derived from pairwise comparisons of aligned homologous sequences. It is difficult to compare the accuracies of different methods for real data as the true tree is generally not known. For this reason simulations are a useful tool for studying the performance of different phylogenetic methods. A desirable property of any method is that inferred trees are not biased towards particular tree shapes; to test this we record different measures of tree shape and accuracy for simulated data sets.

In this study we first use a simulation study to demonstrate some biases distinguishing different heuristic phylogenetic methods. These are applied to the popular Neighbor Joining and UPGMA methods, as well as single and Complete Linkage clustering methods that are less well known to the phylogenetics community. We have restricted our survey to cubic order heuristics, so we do not consider the popular Maximum Parsimony (MP) or Maximum Likelihood (ML) methods, which cannot be computational feasible applied in large scale cases. The current clustering methods all use pairwise distances as input data. We introduce here a novel cubic order heuristic, ParsimonyNJ, which

uses the parsimony criterion for clustering selection and takes sequences as input.

We use several measures of tree shape and accuracy to compare trees across methods for each simulation run. We see that the outcomes depend on the tree generation model implemented to select the trees on which the sequences are simulated, this required us to carefully develop a methodology for generating "random" trees. We find significant variation in both the measures of tree shape and accuracy for the different clustering methods and different tree generation models. Gascuel (2000) performed a similar study focusing on unrooted trees.

7.1 Methods

7.1.1 Random tree generation

Bifurcating, tip-labeled trees were generated under two models, the Yule model and the PDA (Proportional to Distinguishable Arrangements) model (Mooers and Heard, 1997).

The Yule model starts with a single node at time zero. At any time all the extant edges have the same probability to split into two. A tree generated by the Yule model is necessarily rooted and ultrametric. Two properties of Yule trees are that the edge weights are exponentially distributed and that the number of tips grows exponentially with time. The waiting time for the next speciation event in a tree with k taxa is $\exp(k\lambda)$ distributed, where λ is a constant.

The PDA model, sometimes also called the uniform model, is a method of selecting trees at random. Every rooted tree has the same probability to be selected under this model. As the PDA model chooses unweighted trees, edge weights need to be generated separately.

While we are only interested in the unweighted tree and are not concerned about the edge weights there exist also two general models, the α-model by Ford (2005) and β-model by Aldous (1996), which include the Yule and PDA models as special cases.

Rooted or unrooted trees

Trees generated using the Yule model are rooted and ultrametric. If we want a non ultrametric (non-clocklike) tree, we can reassign edge weights by drawing at random from an exponential distribution. Trees generated using the PDA model have no edge weight information. To generate weighted non-clocklike trees we draw the edge weights from an exponential distribution, as we did for a Yule tree. We can also use this procedure to reweight Yule trees.

Assigning the edge weights to an unweighted tree in order to generate a clocklike or ultrametric tree is more difficult. We use the following steps:

1. Construct an unweighted tree using either the Yule or PDA approach.

2. Construct an ordering of the internal nodes in the tree. The root is assigned the number 1. The next node is selected from the set of un-numbered internal nodes adjacent to a numbered node, with probabilities proportional to the number of nodes in the subtree below the node. Tip nodes are not given an order number.

3. Having established an ordering of the internal nodes we then assign edge weights to the tree. Here we have the possibility to model different scenarios of the growth in the number of species, depending on the distribution we use to model the waiting times. We model an exponential growth in the number of species, as for the Yule (1925) tree, by drawing from the exponential distribution $\exp(k\lambda)$ where k is the number of lineages in the time interval of interest. If we draw the waiting times from a distribution which is independent of time we can simulate linear growth.

We construct four different types of weighted trees, with clock-like and non clock-like edge weights for both Yule and PDA trees. In comparison to a Yule tree, the edge weights for a ultrametric PDA tree are not exponentially distributed, and the variance of the edge weights is higher. As the ultrametric PDA trees have many more short edges, reconstructing is more challenging.

7.1.2 Clustering Methods

The first step of each of the clustering methods we consider here is to build a distance matrix d_{ij} from the sequence data. In some methods it is important to correct the distances for multiple changes, otherwise systematic error can mislead the analysis (Huson and Steel, 2004). This is a monotone transformation so it does not affect complete or Single Linkage methods (described below). Although computation for most clustering algorithms needs $O(n^3)$ steps, computation of the distance metric requires $O(n^2m)$, where n and m are the number of species and the length of the sequences. Thus if $m > n$, building the distance matrix may be computationally more costly than the clustering algorithm itself.

For our comparison we will use several of the agglomerative cluster algorithms (UPGMA, WPGMA, single and Complete Linkage clustering). We describe agglomerative clustering in algorithm 7.1 and more details of these algorithms can be found in Sneath and Sokal (1973) or Kaufman and Rousseeuw (1990). I also showed some relation of UPGMA and WPGMA with linear models in section 5.1.1. In phylogenetic analysis the most widely used clustering method is Neighbor Joining (NJ) algorithm (Saitou and Nei, 1987). We include NJ and some extensions (Weighbor by Bruno et al. (2000) and BioNJ by Gascuel (1997a)) in the comparison. We now describe the basic principle of agglomerative clustering (Kaufman and Rousseeuw, 1990; Sneath and Sokal, 1973). The clustering methods differ in their linkage functions (step 3) and also in the selection criteria (step 2):

Linkage functions

The four methods under consideration differ in their linkage functions. In each case the linked clusters are C_v and C_w, where $d(C_v, C_w)$ is minimal over all cluster pairs. For UPGMA,

$$d(C_v, C_w) = d_U(C_v, C_w) = \frac{1}{|C_v||C_w|} \sum_{n \in C_v} \sum_{m \in C_w} d_{n,m}; \tag{7.2}$$

Algorithm 7.1 Agglomerative clustering

1. *Initial condition*
   Each species represents a cluster. The distances between the clusters correspond to the (possibly corrected) distances or dissimilarities between the species of one cluster to those of another.

2. *Selection criterion*
   The two closest clusters C_k and C_l are merged to form a new cluster C_m:

$$d(C_k, C_l) = \min_{i \neq j} d(C_i, C_j) \tag{7.1}$$
$$C_m \longleftarrow C_k \cup C_l$$

3. *Reduction formula*
   The distance matrix is now updated.The clusters C_k and C_l get deleted, and the distances between the cluster C_m and each of the remaining clusters is computed with a *linkage function* (see below).

4. Back to (2) until all clusters are merged.

for WPGMA

$$d(C_v, C_w) = d_W(C_v, C_w) = \frac{1}{2} \sum_{n \in C_v} \sum_{m \in C_w} d_{n,m}; \tag{7.3}$$

for Single Linkage

$$d(C_v, C_w) = d_S(C_v, C_w) = \min_{n \in C_v, m \in C_w} d_{n,m} \tag{7.4}$$

and for Complete Linkage

$$d(C_v, C_w) = d_C(C_v, C_w) = \max_{n \in C_v, m \in C_w} d_{n,m}. \tag{7.5}$$

Hence we see in UPGMA , the distance between clusters C_v and C_w is the average of distances across each pair of taxa, with one taxon from each cluster. In WPGMA, the cluster distances are weighted by the size of the cluster. The distance between clusters in the Single Linkage method is the minimum distance between pairs of taxa, with one taxon from each cluster and for Complete Linkage it is the maximum pairwise distance.

It is reported (Sneath and Sokal, 1973) that trees reconstructed with Single Linkage are likely to be less balanced than those generated by UPGMA or WPGMA, and Complete Linkage is more likely to be balanced. Here we extend this result by comparing different biases among these four clustering methods, and contrast these with the NJ-methods we describe later.

A distance is said to be ultrametric, if it fulfills the following three point condition:

$$d(i, j) \leq \max\{d(i, k), d(j, k)\}. \tag{7.6}$$

Ultrametric distances can always be displayed on a weighted clock-like tree. A weaker condition is additivity, distances are said to be additive if the four point condition is true

$$d(i, j) + d(k, l) \leq \max\{d(i, k) + d(j, l), d(i, l) + d(j, k)\}. \tag{7.7}$$

Additive distances can always be displayed on a weighted tree (Buneman, 1971). Complete Linkage and Single Linkage only require that distances are ordinal, they are invariant against (positive) monotone transformations of the distances.

All the clustering methods described so far are consistent if the distance or dissim-

ilarity matrix is ultrametric, but may be inconsistent for additive distances, that is, given longer and longer sequences they might converge on the wrong phylogenetic tree.

Neighbor Joining

In contrast to the methods above, NJ (Saitou and Nei, 1987; Studier and Keppler, 1988) is consistent for additive distances. At each step the two clusters with the minimal net divergence are merged. Consider a tree where two sister clusters are separated from the rest of the taxa by an edge, but there are no other internal edges. The net divergence is the length of this edge as determined by a least squares best fit.

BioNJ (Gascuel, 1997a) and Weighbor (Bruno et al., 2000) are extensions of the NJ algorithm using a biological model to give more accurate distance estimates when reducing the distance matrix.

We introduce here a new heuristic clustering method that uses the NJ selection criterion on distances derived using a parsimony framework.

Parsimony Neighbor Joining (PNJ)

The Hamming distance between sequences i and j is the minimal number of substitutions required to change i to j. For PNJ we need to generalize the idea of a character state to a set of character states. $C_i[k]$ denotes the set of possible states of sequence i at site k. At site k, if the intersection $C_i[k] \cap C_j[k]$ is not empty, (but $C_i[k] \neq C_j[k]$) no substitution is required. The PNJ algorithm is outlined in table 7.2. The main difference between NJ and PNJ is in step 3 of algorithm 7.2 where we compute an ancestral sequence which is then used for updating the distance matrix. I developed a implementation of PNJ, which is part of the R-package `phangorn`.

7.1.3 Tree Comparison Measures

To compare the trees estimated by the different methods we used a measure of accuracy and several measures of tree shape.

Algorithm 7.2 Parsimony Neighbor Joining

1. Compute the pairwise distances D_{ij} between all the taxa using the Hamming distances from the alignment.
2. Choose clusters i and j for which
$$D_{ij} - u_i - u_j, \qquad u_i = \sum_{j:j\neq i}^{m} \frac{D_{ij}}{m-2}$$
is minimised.
3. Compute an 'ancestral' sequence for cluster i and j. For every site k in the alignment we assign the characters of the intersection $C_i[k] \cap C_j[k]$, provided it is non-empty, otherwise we assign the union $C_i[k] \cup C_j[k]$.
4. Update the distance matrix D by deleting the entries belonging to the clusters i and j, and computing the distance between the 'ancestral' sequence and each remaining sequence.
5. Back to (2) until all clusters are merged.

Partition metric

As our measure of accuracy we used the partition metric (Robinson and Foulds, 1981), also known as Robinson-Foulds metric or symmetric distance, which counts the number of partitions, i.e. the number of edges, where two tree topologies differ. In the simulation study we know the tree on which we generated our sequences on and compare this to the trees recovered. In contrast to other measures of accuracies (NNI, SPR and TBR) the partition metric can be computed efficiently (Semple and Steel, 2003).

We now have a look at several methods for comparing the tree shape. For a overview see Agapow and Purvis (2002) or Blum and Francois (2005).

Counting Cherries

The number of cherries is an easily calculated statistic based on the tree shape. A cherry is a pair of adjacent tips (McKenzie and Steel, 2000; Blum and Francois, 2005). McKenzie and Steel (2000) derive formulae for the mean and variance of the numbers

of cherries on a tree with n leaves for the Yule process

$$E(C_n) = \frac{n}{3} \qquad Var(C_n) = \frac{2n}{45} \tag{7.8}$$

and the uniform model

$$E(C_n) = \frac{n}{4} \qquad Var(C_n) = \frac{n}{16} \tag{7.9}$$

Colless and Sackin

Two frequently used measures of tree shape that apply to rooted trees are the Colless statistic and the Sackin statistics. Blum et al. (2006) showed analytical that both indices are highly correlated with $\rho \sim 0.98$, so we restrict the further analysis to the Sackin index.

The Sackin index sums the number of edges between each tip and the root.

$$S_n = \sum_{i=1}^{n} N_i$$

where N_i is the path length between a leaf i and the root. Under a Yule model the expectation and variance of S_n are

$$E(S_n) = 2n \sum_{i=2}^{n} \frac{1}{i} = 2nH_n - 2n \sim 2n \ln(n) + 2n\gamma - 2n + o(1) \tag{7.10}$$

$$Var(S_n) = 7 - \frac{\pi^2}{2} n^2 \tag{7.11}$$

where H_n is the n-th harmonic number and γ is Euler's constant. This index is equivalent to the external path length, which is of interest in computer science in the analysis of search algorithms (Neininger, 2002).

Path length

Recall that the path length is defined as the number of edges between two leaves (chapter 1.2.1). The expectation for the path length is derived for a Yule tree (Steel and McKenzie, 2001). For a given tree we can estimate the mean of the pairwise path lengths. The expectation for the path length (Steel and McKenzie, 2001) on a Yule

tree with n leaves is the following:

$$d_n = 4H_n - 4 - 4(1 - \frac{2H_n - 2}{n-1}) \sim 2\mu_n - 4 = 4H_n - 8 \sim 4\ln(n) + 4\gamma - 8 + O(1)$$

The expected sum of all pairwise path lengths D_n is therefore $\frac{n(n-1)}{2} d_n$.

So the mean of the

$$E(D_n) \quad = \quad 2n^2 H_n - 4n^2 \sim 2n^2 \ln(n) + o(n^2) \tag{7.12}$$

We derived an empirical cumulative distribution of D_n using Monte Carlo simulation (see figure 7.1).

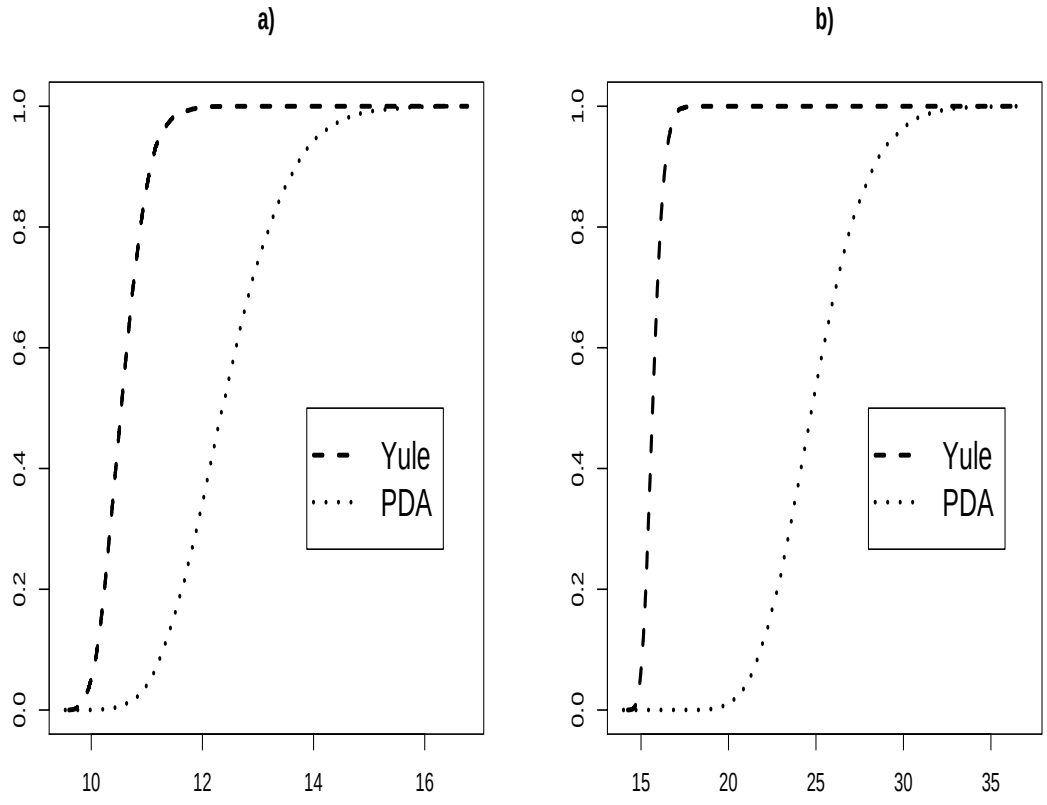


Figure 7.1: The empirical cumulative distribution function for the path length, derived from 10000 simulated trees sampled according to the Yule and PDA methods with a) 50 taxa and b) 200 taxa.

Several studies compared the power of tree shape statistics (Kirkpatrick and Slatkin, 1993; Agapow and Purvis, 2002) to distinguish trees generated by different models, but with knowledge of the covariance between the different shape statistics it is also possible to construct more powerful statistics from the joint distribution of these tree shape statistics (see figure 7.2) by combining statistics which are weakly correlated. For a similar approach see Matsen (2006). Some caution is required with the path length statistic as is it not asymptotically normally distributed.

7.2 Results

7.2.1 Simulation Study

For each of the four tree generating models described in section 7.1.1 we generated 50 random trees with 200 taxa. We normalized so that the expected number of substitutions per site was 0.25 .

We then used seq-gen version 1.3.2 (Rambaut and Grassly, 1997) to generate sequence alignments of length 2000 for each tree, using the Jukes Cantor substitution model. A distance matrix was constructed for each sequence alignment using the Jukes Cantor model to correct for multiple changes (performed using Paup*). The NJ and BioNJ trees were estimated by Paup*, the Weighbor with its own software (Bruno et al., 2000), and the trees for the other cluster methods and PNJ were reconstructed with R.

NJ, BioNJ, Weighbor and PNJ are most accurate as measured by the Robinson-Foulds distance (see figure 7.3). The differences between these NJ-methods are small in comparison to the cluster methods. This is a result we expected, see also Hollich et al. (2005). The method of generating the underlying tree plays a significant role in how well the methods work. If the distances are not clocklike all the methods which rely on ultrametric distances (UPGMA, WPGMA, single and Complete Linkage) perform badly (figure 7.3).

All methods have problems reconstructing the uniform ultrametric trees, as they contain many very short internal edges (Gascuel and McKenzie, 2004). When com-

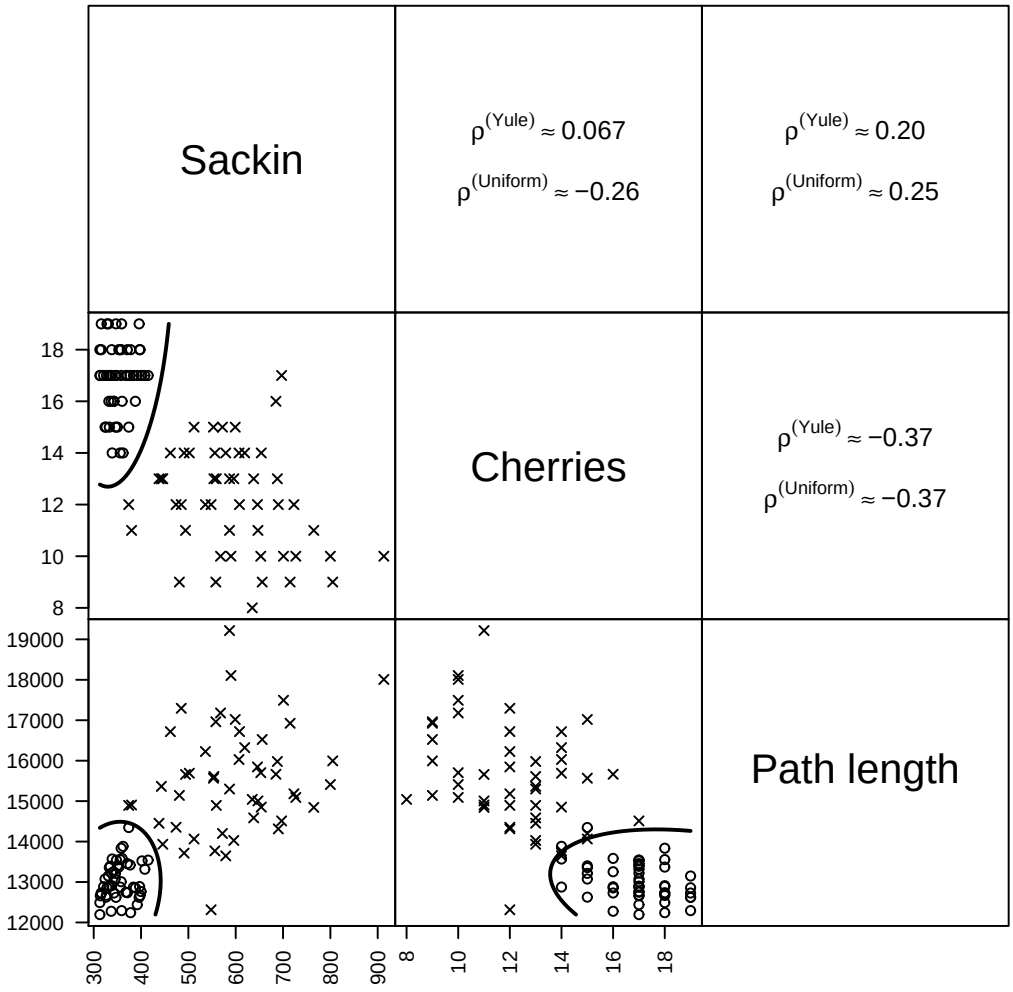


Figure 7.2: In the lower triangle are the scatter plots for different tree measures for 50 random uniform (x) and Yule (○) trees. The lines approximate the border between the two groups, estimated with quadratic discrimination analysis, which assumes that the tree shape statistics are bivariate normal distributed. In the upper triangle the empirical correlations for the methods are given.

paring only the NJ methods, PNJ is less accurate especially when the distances are not ultrametric see figure 7.4. BioNJ and Weighbor are sometimes slightly better in terms of the mean and have a lower variance than NJ. The same results hold for the parsimony score, the other measure of accuracy we used (figure 7.5, 7.6).

The tree shape statistics (figure 7.7) differ from the measures of accuracy. We can see that all the NJ methods are unbiased in the number of cherries. The other linkage methods follow a pattern, Single Linkage is less balanced (fewer cherries) than UPGMA and WPGMA and these are slightly less balanced than Complete Linkage. The variance is higher if the underlying tree is non ultrametric, because of course all the estimates are further away from the tree (see figure 7.3). If trees are generated using the uniform model the number of cherries is much lower.

7.2.2 Case Studies

For the first case study we use a dataset of 461 different flowering plants species from Nickrent et al. (2002). Distances were calculated from an alignment of 4635 characters from three genes (18S, rbcL, atpB). We chose a dataset with a high number of taxa to make differences in tree shape between the different clustering methods more apparent. Table 7.1 shows the results for the different cluster methods.

Method	Parsimony Score	number of cherries	p-value (Yule)	p-value (uniform)
NJ	37465	151	0.556	< 0.001
BioNJ	37403	152	0.713	< 0.001
Weighbor	37403	152	0.713	< 0.001
PNJ	38058	146	0.090	< 0.001
UPGMA	39751	134	< 0.001	0.001
WPGMA	38960	144	0.033	< 0.001
Single Linkage	46631	119	< 0.001	0.529
Complete Linkage	39114	147	0.141	< 0.001

Table 7.1: Parsimony score and numbers of cherries for the trees generated by the different methods for the Nickrent et al. (2002) data set. The p-values are given for testing the null-hypothesis that trees originate from a Yule process and a uniform model.

NJ, BioNJ and Weighbor perform well according to the parsimony score, and so

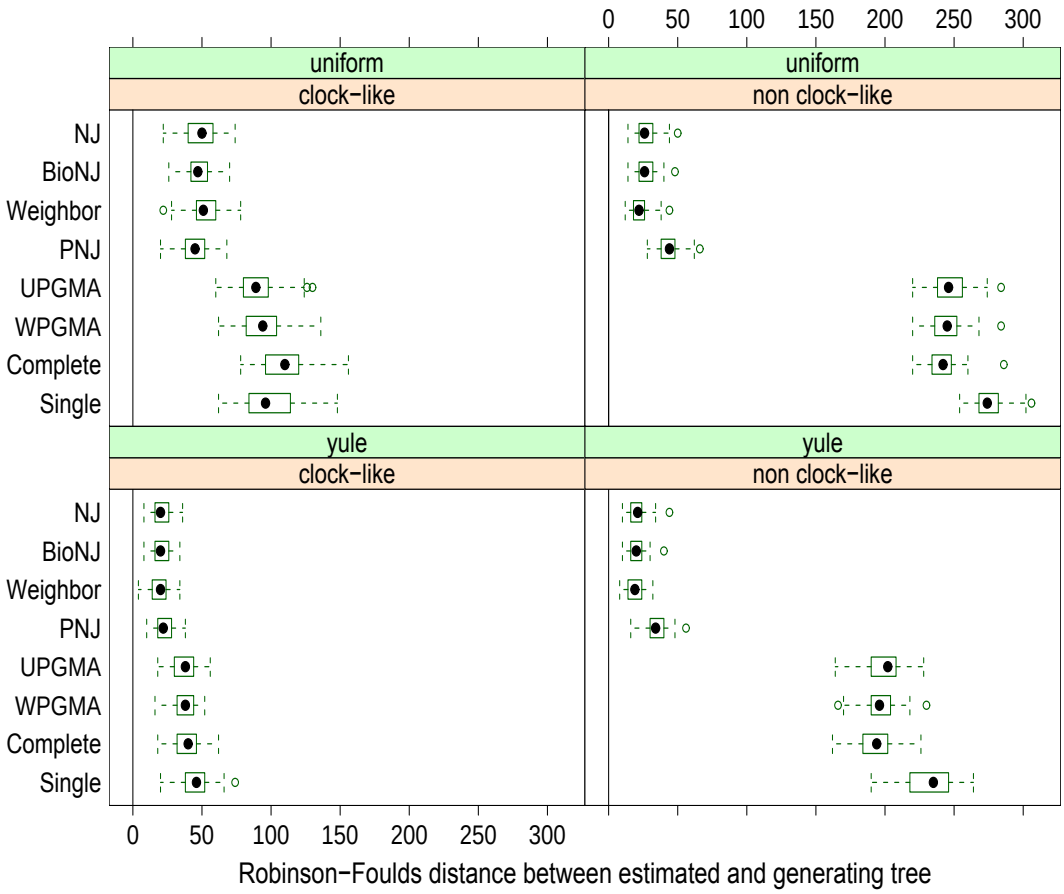


Figure 7.3: Robinson-Foulds distances between the generating tree and the different estimates represented by a boxplot (Tukey, 1977). An estimated tree which recovers the underlying generating tree would have distance zero. The length of the box spans from the first to the third quartile, the dot within the box represents the median. The whiskers represent a range (similar to a 95% quantile) and outliers are represented by unfilled circles.

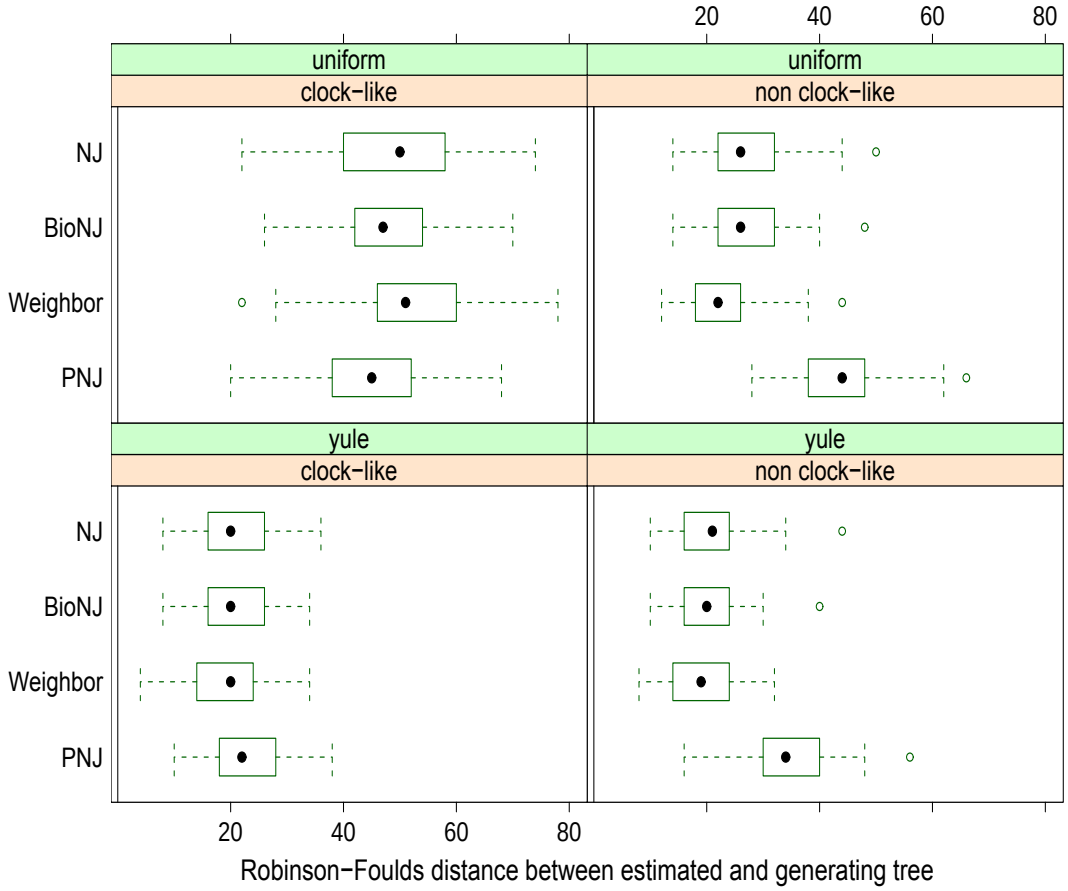


Figure 7.4: Robinson-Foulds distances between the generating tree and the different estimates of the NJ methods. If the underlying tree is not clock-like PNJ performs less well than the other methods. The differences between these methods are small in comparison to the cluster methods. This is a result we expected, see also Hollich et al. (2005). All methods have problems reconstructing the uniform ultrametric trees, because of the many very short internal edges (Gascuel and McKenzie, 2004).

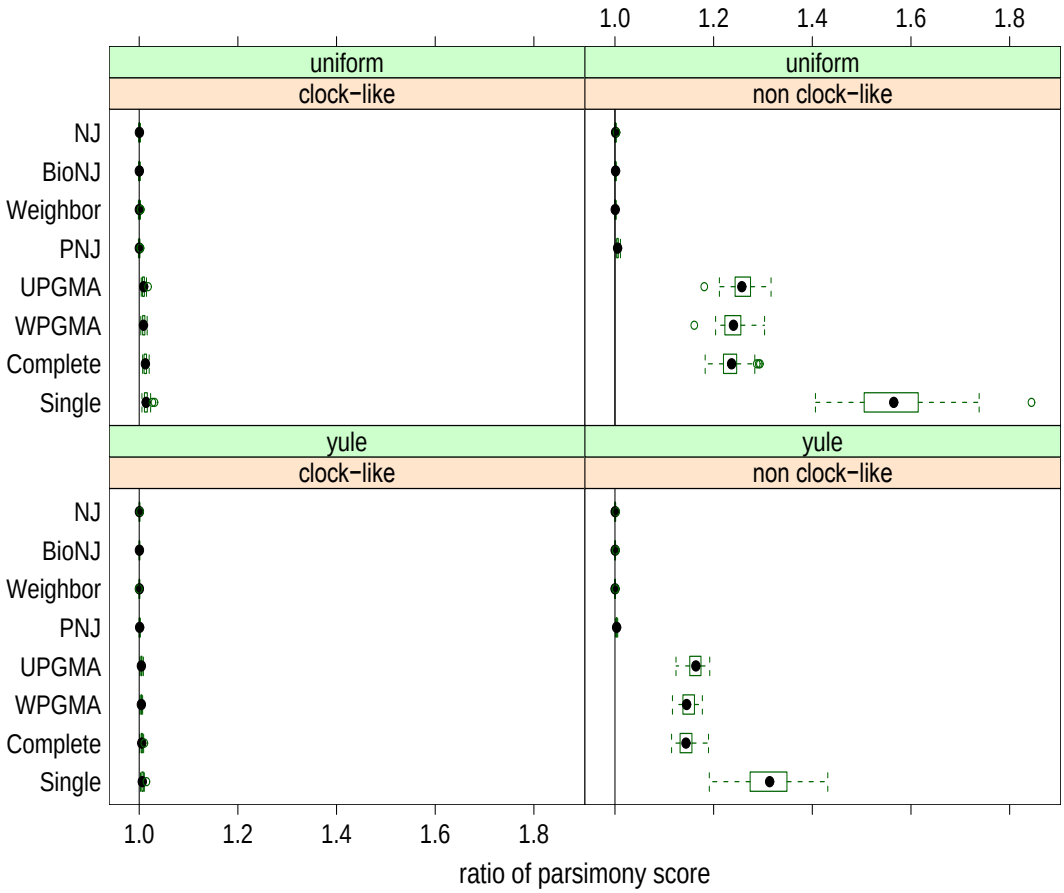


Figure 7.5: Ratio of the parsimony score between the generating tree and the cluster methods. Most classical clustering methods are not robust, if the distances are not ultrametric. In the next graph we will only compare the methods using the selection criterion similar to Neighbor Joining.

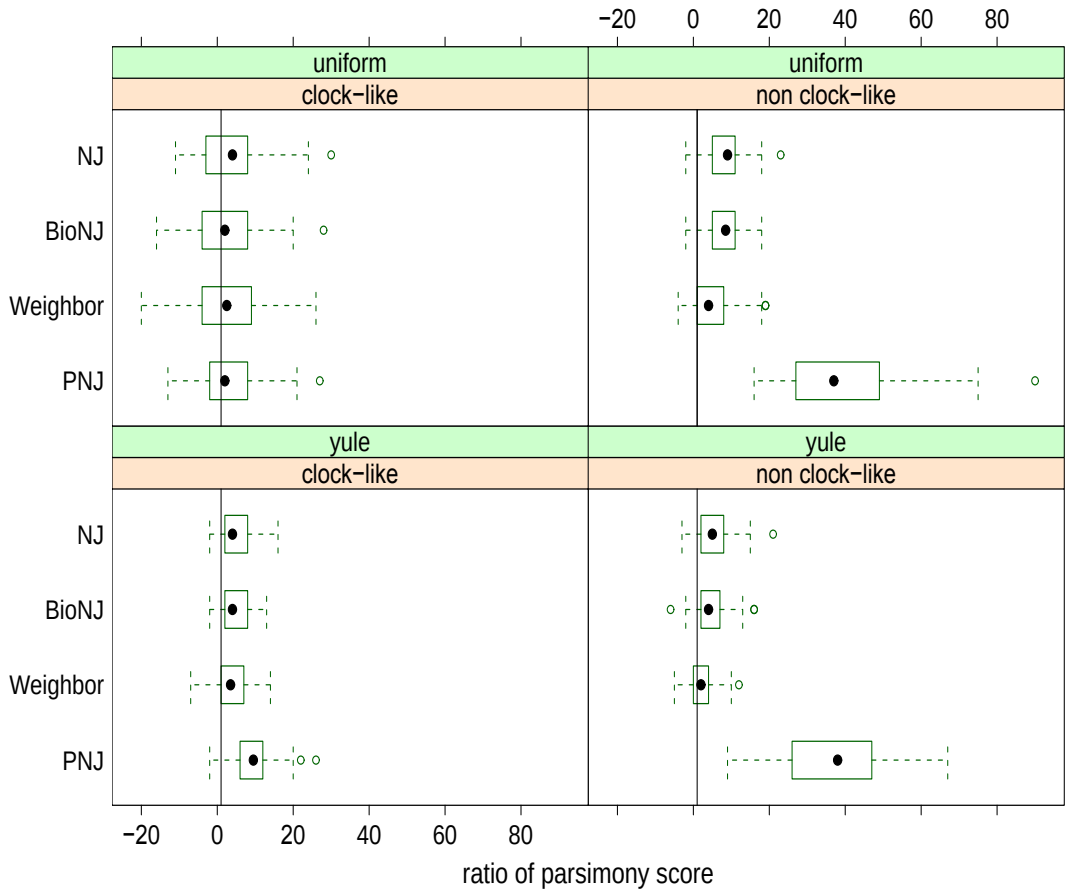


Figure 7.6: Ratio of the parsimony score between the generating tree and the Neighbor joining methods. For the ultrametric uniform case all the methods find trees with parsimony scores close to that of the generating tree. Note that those methods can overfit the data in the sense that they produce trees with lower parsimony score than the generating tree. For non clock-like trees the PNJ tree is biased, as parsimony does not account for multiple changes.

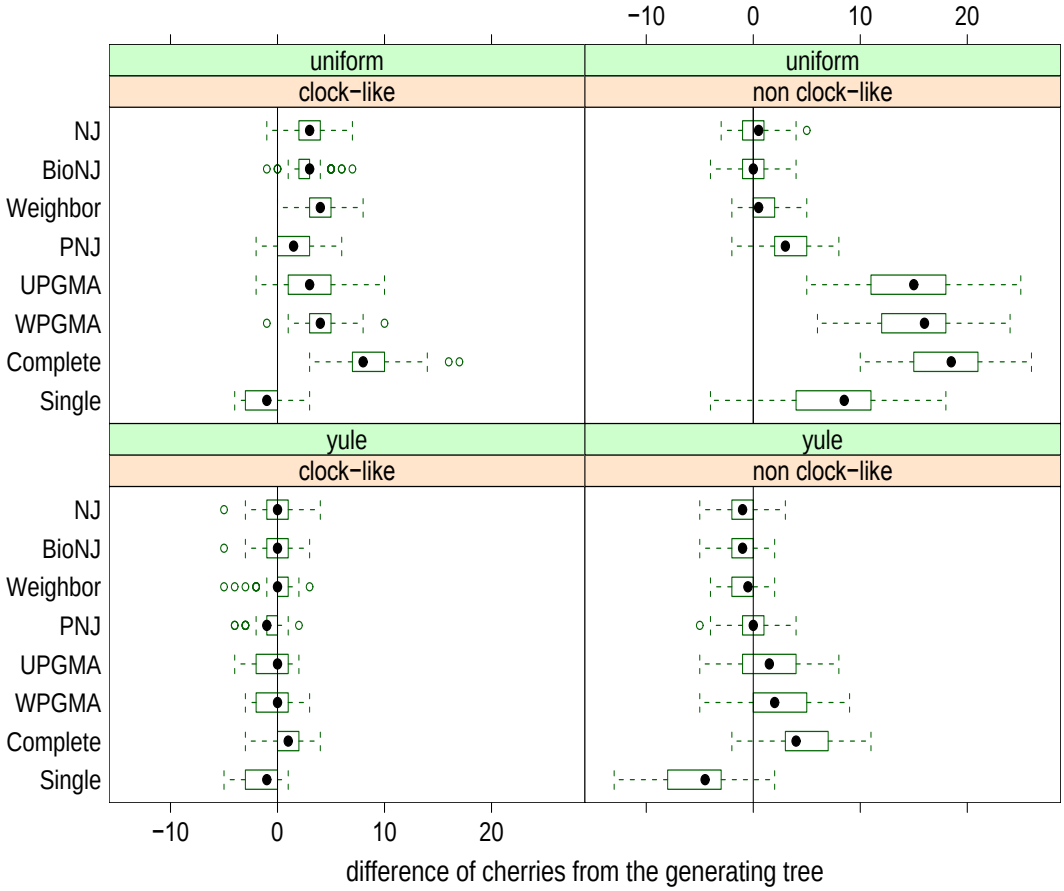


Figure 7.7: Differences in the number of cherries. The reference indicated by a vertical line is the number of cherries in the generating tree. NJ, BioNJ and Weighbor perform well even if the distances are not ultrametric. UPGMA and the other clustering methods tend to have more cherries than the generating tree if the distance matrix is not ultrametric.

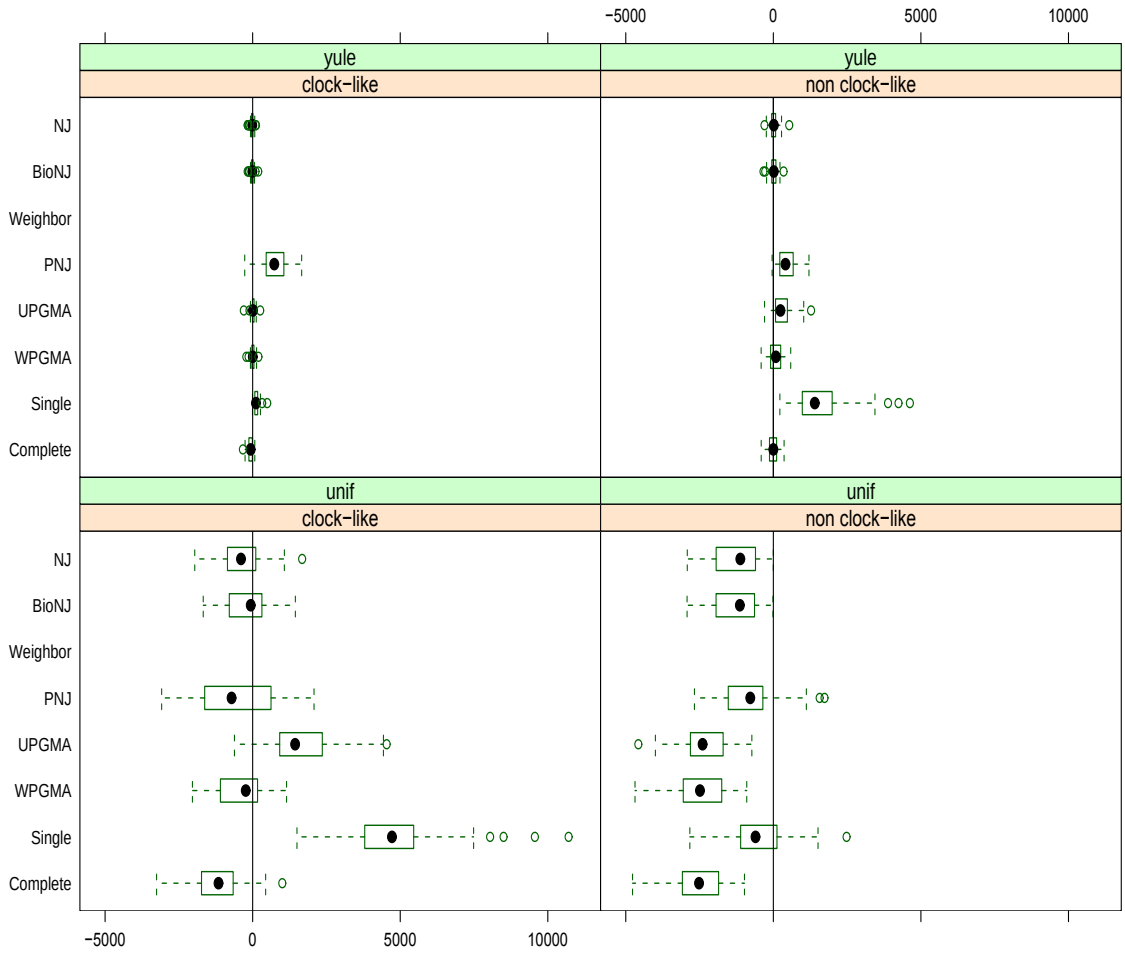


Figure 7.8: Differences in the Sackin index between the different methods and the generating tree. Positive values indicate a less balanced tree than the reference tree. Single linkage produces unbalanced trees. The Sackin index for Weighbor is not evaluated as there is no information on which cluster was joined last to determine the position of the root.

does PNJ to a lesser extent. The number of cherries is close to the number we expect from data generated under a Yule process (153.7). The two methods which perform worst according to the parsimony score have a very low number of cherries (UPGMA and Single Linkage). The Single Linkage tree is the only one which does not reject the hypothesis that the true tree comes from a PDA model (115.3).

The second dataset was comprised of human mitochondrial DNA from 311 complete sequences and 16569 characters (Ingman and Gyllensten, 2006). The average pairwise distance between two sequences is 50 substitutions, in contrast the Nickrent data with 468.

Here the PNJ method worked better than the other NJ methods. This is interesting as Steel and Penny (2004) have shown that when sequences are densely sampled maximum parsimony is a maximum likelihood estimator. It suggests that PNJ could be a useful cubic order method for such data.

Once again the Single Linkage tree has significantly fewer cherries than the trees produced by other methods and the number is outside the range deemed plausible under the Yule model.

Method	Parsimony Score	number of cherries	p-value (Yule)	p-value (uniform)	Path length
NJ	3113	104	0.93	< 0.01	18.74
BioNJ	3113	104	0.93	< 0.01	18.91
Weighbor	3137	108	0.24	< 0.01	17.92
PNJ	3112	104	0.93	< 0.01	20.29
UPGMA	3297	99	0.21	< 0.01	28.26
WPGMA	3236	103	0.87	< 0.01	23.45
Single Linkage	3540	84	< 0.01	0.18	38.47
Complete Linkage	3233	105	0.72	< 0.01	19.35

Table 7.2: Parsimony score and numbers of cherries for the trees generated by the different methods for the human mitochondrial DNA data.

7.3 Discussion

Overall the NJ methods had better accuracy (Robinson-Foulds), lower parsimony scores, and were less biased in tree shape than the clustering methods.

This was not just for non-ultrametric distances where it was expected, but also for ultrametric distances. Additional tree rearrangements like NNI, SPR or TBR could be used to improve those methods where an optimality criterion is defined (for example WLS for WPGMA or UPGMA). Our result may be specific to simulations of phylogenetic data, where the variance of the distance estimates (between two leaves) is not independently identically distributed (Felsenstein, 2004), but rather dependent on the edge weight. This effect favours NJ methods over clustering methods (Gascuel and McKenzie, 2004).

We do not expect to recover the underlying tree perfectly and need to know if a specific clustering method introduces a bias. Most of the tree shape statistics we used measure the balance of the tree, comparing the sizes of subtrees. These methods need trees to be rooted, but when the root is not known, a misspecification of the root can affect the statistics. Comparative studies of tree shape should be careful to account for how the tree is rooted and also the influence of an outgroup. The cherry statistics or path length can be calculated for unrooted trees, and are independent of the placement of the root, but is defined only for bifurcating trees.

As expected we found that Single Linkage produced less balanced trees than WPGMA and UPGMA and these are less balanced than Complete Linkage trees; these deviations are well known (Sneath and Sokal, 1973; Kaufman and Rousseeuw, 1990).

For the NJ methods we could not find such a distinct pattern. PNJ can be biased towards unbalanced trees, because this method does not account for multiple changes and also how the 'ancestral' sequences are computed, but it performs well if there are only few expected substitutions per site. The shape statistics for PNJ are very similar to the other NJ methods and there seems to be no systematic error, but the estimators for BioNJ in general have a smaller variance than NJ itself. Weighbor often performs slightly better than NJ, but not always and is more time intensive (ca. 200 seconds

versus less than 3 seconds for the simulations with 200 taxa) in comparison to the other methods.

When the number of substitutions between the sequences is low, for example comparing human mitochondrial genes, the differences between the methods are smaller. In this case PNJ performs well.

The shape statistics (cherries, path length) indicate that Weighbor produces slightly more balanced trees than NJ or BioNJ, but the differences are small. On the other hand the Sackin (and also Colless) index tends in the opposite direction. This seems to be an artefact of the rooting. One should be aware that the shape statistics can be influenced by phylogenetic reconstruction methods if one observes many short branches.

Appendix A

The R-package phangorn

Many of the the algorithms I described I have implemented in the statistical programming language R. R allows a fast prototyping of new ideas, but allows where speed is required to call native C/C++ and FORTRAN routines. Furthermore the language allows user interactions, in the respect that once an estimate is produced it is possible to allow further computations with this object, for example result can be plotted.

Most of the functions I describe here are made available in the package phangorn which is available from the CRAN `http://www.cran.r-project.org/web/packages/phangorn/index.html`. It depends on the package `ape` by (Paradis et al., 2004), which provides many functions for reading, writing, plotting and manipulating phylogenetic trees and the package `quadprog`, which supplies routines to solve quadratic programming problems. There is a vast amount of literature about the R language now, for a general introduction see for example Dalgaard (2002). There are many materials available on the website `http:\www.r-project.org`, and an introduction with a focus on phylogenetics is Paradis (2006).

This appendix is not an introduction into R, but should enable the reader to reproduce results represented in the chapters above. The examples are generated using Sweave (Leisch, 2002). For readability of the source code most messages, output or plots from the program which are not essential for the understanding of the algorithms are omitted. Also time consuming loops or bootstrapping are left out in this appendix, so that all the examples should run in several minutes. The results were tested under

`R version 2.8.0`, `phangorn 0.0-4`, `ape 2.2-2` and `quadprog 1.4-11`.

I will describe now some of the functionality which is used in different chapters. More information and examples of the implemented function are available in the help pages of the function, e.g. `?pml` will give more information about the function `pml`. Lines with comments start with a hash `#`.

A.1 Mixture models

The following code produces some of the mixture models of section 2.1.4. We create data based on the exact expected site frequencies. The advantage of this over simulated data from seq-gen is that we can verify if the optimum is reached as we know the exact maximum.

```
> library(phangorn)
> # read in trees
> tree1 <- read.tree(text = "((t1:0.3,t2:0.3):0.1,(t3:0.3,t4:0.3)
      :0.1,t5:0.5);")
> tree2 <- read.tree(text = "((t1:0.3, t3:0.3):0.1,(t2:0.3,t4:0.3)
      :0.1,t5:0.5);")
> # create dataset
> X <- allSitePattern(5)
> fit1 <- pml(tree1,X)
> fit2 <- pml(tree2,X)
> # give site pattern the exact weights of 60:40 mixture
> # based on the expected site frequencies
> weights <- 6000*exp(fit1$site) + 4000*exp(fit2$site)
> attr(X, "weight") <- weights
> # fit and present ML and mixture estimates to the mixture data
> fit1 <- update(fit1, data=X)
> fit2 <- update(fit2, data=X)
> (fitM <- pmlMix(~edge+nni, fit2 , m=2))
> (fit1 <- optim.pml(fit1))
> (fit2 <- optim.pml(fit2))
> par(mfrow = c(2,2))
> plot(fit1, main="Tree1 100%", "u")
> plot(fit2, main="Tree1 100%", "u")
> plot(fitM$fits[[1]], main=paste("Mix: Tree1 ",
      round(fitM$omega[1],4),"%", sep=""), "u")
> plot(fitM$fits[[2]], main=paste("Mix: Tree2 ",
      round(fitM$omega[2],4),"%", sep=""), "u")
```

```
> logLik(fit1)
> logLik(fit2)
> logLik(fitM)
```

The posterior of the estimated mixture and also the edge length may differ from the input trees.

A.2 Multiple optima

The following code is used to generate the example in section 3.3.1. It produces a object `trees` which contains 10 phylogenies all with the same topology as `tree1`, but with two different sets of edge lengths.

```
> X <- allSitePattern(4)
> tree1 = read.tree(text = "((t1:0.05,t2:0.05):0.5,t3:0.05,t4:0.05);")
> tree2 = read.tree(text = "((t1:0.05,t3:0.05):0.3,t2:0.05,t4:0.05);")
> fit1 <- pml(tree1,X)
> fit2 <- pml(tree2,X)
> weights = 1000*exp(fit1$site) + 1000*exp(fit2$site)
> attr(X, "weight") = weights
> trees = list()
> likelihood = numeric(10)
> set.seed(1)
> tree = tree1
> for(i in 1:10){
     tree$edge.length = -.25*log(runif(5))
     fit = pml(tree, X)
     fit = optim.pml(fit)
     trees[[i]] = fit$tree
     likelihood[i] = logLik(fit)
  }
> class(trees) = "multiPhylo"
> plot(trees,"u")
```

A.3 Partition models

I will now present how to cluster genes using the stochastic partitioning algorithm described in section 4.1.1 implemented in the function `pmlCluster`. We first compute

an ML estimate for the concatenated yeast dataset (Rokas et al., 2003) with a Γ+I model. The different clusters are allowed to differ in their evolutionary rates, but the edge lengths are optimised for all genes, the remaining parameters are taken from the initial concatenated model. In section 4.2.1 I used a model which uses a more extensive parameter optimisation.

```
> data(yeast)
> dm <- dist.logDet(yeast)
> tree <- NJ(dm)
> weight <- xtabs(~ index+genes, attr(yeast, "index"))
> fit <- pml(tree, yeast, inv=.2, k=4)
> # optimise GTR (gamma + I) for concatenated data
> fit <- optim.pml(fit, TRUE, TRUE, TRUE, TRUE, TRUE)
> set.seed(123)
> fitCluster <- pmlCluster(edge~rate, fit, weight=weight, p=4)
```

Next we visualise the relationship between the genes based on the multidimensional scaling based on the distance Hadamard as described in section 4.2.3:

```
> data(chloroplast)
> weight <- xtabs(~ index+genes, attr(chloroplast, "index"))
> l <- dim(weight)[2]
> tmp <- chloroplast
> res = matrix(NA, 262143, l)
> colnames(res) <- colnames(weight)
> for(i in 1:l){
        attr(tmp, "weight") = weight[, i]
        dm <- dist.ml(tmp, "JTT")
        res[,i] = distanceHadamard(dm)[-1,2]
  }
> distGenes <- dist(t(res))
> colnames(res) <- colnames(weight)
> mds <- cmdscale(distGenes, k=2)
> plot(mds, type="n", xlim=c(-.5,1.3), main="MDS", xlab=
      "1st principal component", ylab="2nd principal component")
> text(mds, labels=rownames(mds))
```

A.4 Distance methods and penalized likelihood

The UPGMA and UPGMA method are implemented in R in the function `hclust` with the option the method set to ”average” and ”mcquitty”. The function `upgma` is just a

wrapper around the function `hclust` creating an tree object that can be used in the phangorn or ape package. FastME is implemented in the ape package. We read in a dataset with 47 different species and compute trees using UPGMA, WPGMA and FastME. Then we compute the design matrices to these trees and verify the equivalence of the edge weights from the (weighted) linear model fits with the trees from the UPGMA, WPGMA and fastME procedures.

```
> library(phangorn)
> data(Laurasiatherian)
> dm <- dist.logDet(Laurasiatherian)
> names <- attr(dm, "Labels")
> treeUPGMA <- upgma(dm)
> treeWPGMA <- upgma(dm, method = "mcquitty")
> treeFME <- fastme.bal(dm)
> X.UPGMA <- designTree(treeUPGMA, "rooted")
> fitUPGMA <- lm(dm ~ X.UPGMA - 1)
> summary(fitUPGMA)
> all.equal(drop(attr(X.UPGMA, "contrast") %*% fitUPGMA$coef),
      treeUPGMA$edge.length)
> X.WPGMA <- designTree(treeWPGMA, "rooted")
> weights <- 2^-rowSums(designTree(treeWPGMA))
> fitWPGMA <- lm(dm ~ X.WPGMA - 1, weights = weights)
> summary(fitWPGMA)
> all.equal(drop(attr(X.WPGMA, "contrast") %*% fitWPGMA$coef),
      treeWPGMA$edge.length)
> X.FME <- designTree(treeFME)
> weights <- 2^-rowSums(X.FME)
> y = as.matrix(dm)[treeFME$tip, treeFME$tip]
> y = y[lower.tri(y)]
> fitFME <- lm(y ~ X.FME - 1, weights = weights)
> summary(fitFME)
> all.equal(as.numeric(fitFME$coef), treeFME$edge.length)
```

Now we show two possibilities to compute splits graph. The first is based on the Hadamard conjugation, the second uses an algorithm which is similar to the elastic net. I implemented the network splits network method using a quadratic programming algorithm to satisfy the non-negativity and penalty constraints.

```
> data(yeast)
> dm = dist.logDet(yeast)
```

```
> fitDH = distanceHadamard(dm)
> # choose only edges greater than 0.02
> fitDH = fitDH[fitDH$edges>.002,]
> write.nexus.splits(fitDH, file="tmp.nxs")
> fitNLAR <- splitsNetwork(dm, lambda=.001, gamma= .1)
> write.nexus.splits(fitNLAR, file="tmp.nxs")
```

The nexus file "tmp.nxs" contains the information about the splits graph and can be visualised with Spectronet (Huber et al., 2002) or Splitstree (Huson and Bryant, 2006).

Bibliography

Agapow, P.-M. and Purvis, A. (**2002**). "Power of eight tree shape statistics to detect nonrandom diversification: a comparison by simulation of two models of cladogenesis." *Systematic Biology*, 51(6): 866–872.

Agresti, A. (**2002**). *Categorical data analysis*. Wiley, New York.

Akaike, H. (**1974**). "A New Look at the Staistical Model Identification." *IEEE Transactions on Automatic Control*, 19(6): 716–723.

Aldous, D. J. (**1996**). "Probability distribution on Cladograms." In D. J. Aldous and R. Permantle, editors, "Random Discrete Structures," Springer, Berlin.

Aldous, D. J. (**2001**). "Stochastic Models and Descriptive Statistics for Phylogenetic Trees, from Yule to Today." *Statistical Science*, 16(1): 23–34.

Allman, E. S., Ane, C., and Rhodes, J. A. (**2008**). "Identifiability of a Markovian model of molecular evolution with gamma-distributed rates." *Advances in Applied Probability*, 40(1): 229–249.

Allman, E. S. and Rhodes, J. A. (**2003**). "Phylogenetic invariants for the general Markov model of sequence mutation." *Mathematical Biosciences*, 186(2): 113–144.

Allman, E. S. and Rhodes, J. A. (**2006**a). "The identifiability of tree topology for phylogenetic models, including covarion and mixture models." *Journal of Computational Biology*, 13(5): 1101–1113.

Allman, E. S. and Rhodes, J. A. (**2006**b). "Phylogenetic invariants for stationary base composition." *Journal of Symbolic Computation*, 41(2): 138–150.

Allman, E. S. and Rhodes, J. A. (**2007**). "Identifying evolutionary trees and substitution parameters for the general Markov model with invariable sites." *arXiv*, :0812.5005v1 [q-bio.PE].

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (**2000**). "Gene Ontology: tool for the unification of biology." *Nature Genetics*, 25: 25–29.

Atteson, K. (**1999**). "The performance of neighbor-joining methods of phylogenetic reconstruction." *Algorithmica*, 25(2-3): 251–278.

Bar-Hen, A., Mariadassou, M., Poursat, M.-A., and Vandenkoornhuyse, P. (**2008**). "Influence Function for Robust Phylogenetic Reconstructions." *Molecular Biology and Evolution*.

Ben-Hur, A., Elisseeff, A., and Guyon, I. (**2002**). "A stability based method for discovering structure in clustered data." *Pacific Symposium on Biocomputing*, 7: 6–17.

Benjamini, Y. and Yekutieli, D. (**2001**). "The control of the false discovery rate in multiple hypothesis testing under dependency." *Annals of Statistics*.

Bevan, R. B., Bryant, D., and Lang, B. F. (**2007**). "Accounting for gene rate heterogeneity in phylogenetic inference." *Systematic Biology*, 56(2): 194–205.

Blum, M. G. B. and Francois, O. (**2005**). "On statistical tests of phylogenetic tree imbalance: The Sackin and other indices revisited." *Mathematical Biosciences*, 195(2): 141–153.

Blum, M. G. B., Francois, O., and Janson, S. (**2006**). "The mean, variance

and limiting distribution of two statistics sensitive to phylogenetic tree balance." *to appear in: Annals of Applied Probability*.

Bruno, W. J., Socci, N. D., and Halpern, A. L. (**2000**). "Weighted neighbor joining: A likelihood-based approach to distance-based phylogeny reconstruction." *Molecular Biology and Evolution*, 17(1): 189–197.

Bryant, D. (**2005**). "On the uniqueness of the selection criterion in Neighbor-Joining." *Journal of Classification*, 22(1): 3–15.

Bryant, D. and Moulton, V. (**2004**). "Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks." *Molecular Biology and Evolution*, 21(2): 255–265.

Bulmer, M. (**1991**). "Use of the Method of Generalized Least-Squares in Reconstructing Phylogenies from Sequence Data." *Molecular Biology and Evolution*, 8(6): 868–883.

Buneman, P. (**1971**). "The recovery of trees from measures of dissimilarity." In "Mathematics in the Archaeological and Historical Sciences," pages 387–395. Edinbourgh University Press.

Cavender, J. A. (**1978**). "Taxonomy with confidence." *Mathematical Biosciences*, 40(3-4): 271–280.

Chang, J. T. (**1996**). "Full reconstruction of Markov models on evolutionary trees: Identifiability and consistency." *Mathematical Biosciences*, 137(1): 51–73.

Chor, B., Hendy, M. D., Holland, B. R., and Penny, D. (**2000**). "Multiple maxima of likelihood in phylogenetic trees: an analytic approach." *Molecular Biology and Evolution*, 17(10): 1529–1541.

Dalgaard, P. (**2002**). *Introductiory statistics with R*. Statistics and Computing. Springer, New York.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (**1977**). "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society B*, 39(1): 1–38.

Desper, R. and Gascuel, O. (**2004**). "Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting." *Molecular Biology and Evolution*, 21(3): 587–598.

Desper, R. and Gascuel, O. (**2005**). "The minimum evolution distance-based approach to phylogenetic inference." In O. Gascuel, editor, "mathematics of evolution and phylogeny," Oxford University Press, Oxford.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (**2004**). "Least angle regression." *Annals of Statistics*, 32(2): 407–499.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (**1998**). "Cluster analysis and display of genome-wide expression patterns." *Proceedings of the National Academy of Sciences of the United States of America*, 95(25): 14 863–14 868.

Fahrmeir, L. and Tutz, G. (**1996**). *Multivariate statistische Verfahren*. de Gruyter, Berlin.

Fahrmeir, L. and Tutz, G. (**1997**). *Multivariate statistical modelling based on generalized linear models*. Springer, New York.

Felsenstein, J. (**1981**). "Evolutionary trees from DNA sequences: a maxumum likelihood approach." *Journal of Molecular Evolution*, 17: 368–376.

Felsenstein, J. (**1985**). "Confidence-Limits on Phylogenies - an Approach Using the Bootstrap." *Evolution*, 39(4): 783–791.

Felsenstein, J. (**2004**). *Inferring Phylogenies*. Sinauer Associates, Sunderland.

Fitch, W. M. and Margoliash, E. (**1967**). "Construction of Phylogenetic Trees." *Science*, 155: 279–284.

Ford, D. J. (**2005**). "Probabilities on cladograms: introdution of the alpha model." *arXiv:math.PR/0511246v1*.

Fowlkes, E. B. and Mallows, C. L. (**1983**). "A Method for Comparing Two Hierarchical Clusterings." *Journal of the American Statistical Association*, 78: 553–569.

Fukami, K. and Tateno, Y. (**1989**). "On the maximum likelihood method for estimating molecular trees: uniqueness of the likelihood point." *Journal of Molecular Evolution*, 28: 460–464.

Gamerman, D. (**1997**). *Markov chain Monte Carlo : stochastic simulation for Bayesian inference*. Chapman & Hall, London.

Garthwaite, P. H., Jolliffe, I. T., and Jones, B. (**2002**). *Statistical inference*. OUP, Oxford.

Gascuel, O. (**1997**a). "BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data." *Molecular Biology and Evolution*, 14(7): 685–695.

Gascuel, O. (**1997**b). "Concerning the NJ algorithm and its unweighted version, UNJ." In B. Mirkin, F. McMorris, F. Roberts, and A. Rzhetsky, editors, "Mathematical Hierarchies and Biology," pages 149–170. Providence.

Gascuel, O. (**2000**). "Evidence for a relationship between algorithmic scheme and shape of inferred trees." In W. Gaul, O. Opitz, and M. Schader, editors, "Data Analysis, Scientific Modeling and Practical Applications," pages 157–168. Springer, Berlin.

Gascuel, O. and McKenzie, A. (**2004**). "Performance analysis of hierarchical clustering algorithms." *Journal of Classification*, 21(1): 3–18.

Gascuel, O. and Steel, M. (**2006**). "Neighbor-joining revealed." *Molecular Biology and Evolution*, 23(11): 1997–2000.

Gatesy, J., DeSalle, R., and Wahlberg, N. (**2007**). "How many genes should a systematist sample? Conflicting insights from a phylogenomic matrix characterized by replicated incongruence." *Systematic Biology*, 56(2): 355–363.

Gentleman, R., Scholtens, D., Ding, B., Carey, V., and Huber, W. (**2005**). "Case Studies Using Graphs on Biological Data." In R. Gentleman, V. Carey, W. Huber, A. Irizarry, and S. Dudoit, editors, "Bioinformatics and Computational Biology Solutions Using R and Bioconductor," Springer, New York.

Goldfarb, D. and Idnani, A. . (**1982**). "Dual and Primal-Dual Methods for Solving Strictly Convex Quadratic Programs." In J. Hennart, editor, "Numerical Analysis," pages 226–239. Springer, Berlin.

Goldfarb, D. and Idnani, A. (**1983**). "A numerically stable dual method for solving strictly convex quadratic programs." *Mathematical Programming*, 27: 1–33.

Goldman, N., Anderson, J., and Rodrigo, A. (**2000**). "Likelihood-based tests of topologies in phylogenetics." *Systematic Biology*, 49: 652–670.

Gower, J. C. (**1966**). "Some distance properties of latent root and vector methods used in multivariate analysis." *Biometrika*, 53: 325–328.

Gregory, T. R. (**2008**). "Understanding Evolutionary Trees." *Evolution: Education and Outreach*, 1(2): 121–137.

Hartigan, J. A. and Wong, M. A. (**1979**). "A K-means clustering algorithm." *Applied Statistics*, 28: 100–108.

Hastie, T. and Tibshirani, R. (**1990**). *Generalized Additive Models*. Chapman and Hall, London.

Hastie, T., Tibshirani, R., and Friedman, J. (**2001**). *The Elements of Statistical Learning*. Springer, New York.

Hendy, M. (**1989**). "The relationship between simple evolutionary tree models and observable sequence data." *Systematic Zoology*, 38(4): 310–321.

Hendy , M. (**2005**). "Hadamard conjugation: an analytical tool for phylogenetics." In O. Gascuel, editor, "mathematics of evolution and phylogeny," Oxford University Press, Oxford.

Hendy, M., Penny, D., and Steel, M. (**1994**). "A Discrete Fourier analysis for evolutionary trees." *Proceedings of the National Academy of Sciences of the United States of America*, 91: 3339–3343.

Hendy, M. D. and Penny, D. (**1993**). "Spectral Analysis of Phylogenetic Data." *Journal of Classification*, 10(1): 5–24.

Hoerl, A. E. and Kennard, R. W. (**1970**). "Ridge Regression - Biased Estimation for Nonorthogonal Problems." *Technometrics*, 12(1): 55–67.

Hollich, V., Milchert, L., Arvestad, L., and Sonnhammer, E. L. L. (**2005**). "Assessment of protein distance measures and tree-building methods for phylogenetic tree reconstruction." *Molecular Biology and Evolution*, 22(11): 2257–2264.

Hornik, K. (**2005**). "A CLUE for CLUster Ensembles." *Journal of Statistical Software*, 14(12).

Huber, K. T., Langton, M., Penny, D., Moulton, V., and Hendy, M. (**2002**). "Spectronet: A package for computing spectra and median networks." *Applied Bioinformatics*, 1(3): 159–161.

Huelsenbeck, J. P. and Ronquist, F. (**2001**). "MrBayes: Bayesian inference of phylogenetic trees." *Bioinformatics*, 17: 754–755.

Huson, D. H. and Bryant, D. (**2006**). "Application of Phylogenetic Networks in Evolutionary Studies." *Molecular Biology and Evolution*, 23(2): 254–267.

Huson, D. H. and Steel, M. (**2004**). "Distances that perfectly mislead." *Systematic Biology*, 53(2): 327–332.

Ihaka, R. and Gentleman, R. (**1996**). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, 5(3): 299–314.

Ingman, M. and Gyllensten, U. (**2006**). "mtDB: Human Mitochondrial Genome Database, a resource for population genetics and medical sciences." *Nucleic Acids Research*, 34: D749–D751.

Jukes, T. H. and Cantor, C. R. (**1969**). "Evolution of protein molecules." In H. N. Munro, editor, "Mammalian Protein Metabolism," pages 21–132. Academic Press, New York.

Kaufman, L. and Rousseeuw, P. J. (**1990**). *Finding groups in data : an introduction to cluster analysis*. Wiley, New York.

Kelchner, S. A. and Thomas, M. A. (**2006**). "Model use in phylogenetics: nine key questions." *Trends in Ecology and Evolution*, 22(2): 87–94.

Kimura, M. (**1980**). "A Simple Method for Estimating Evolutionary Rates of Base Substitutions through Comparative Studies of Nucleotide-Sequences." *Journal of Molecular Evolution*, 16(2): 111–120.

Kimura, M. (**1981**). "Estimation of Evolutionary Distances between Homologous Nucleotide-Sequences." *Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences*, 78(1): 454–458.

Kirkpatrick, M. and Slatkin, M. (**1993**). "Searching for evolutionary patterns in the shape of a phylogenetic tree." *Evolution*, 47(4): 1171–1181.

Kishino, H. and Hasegawa, M. (**1989**). "Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea." *Journal of Molecular Evolution*, 29: 170–179.

Kolaczkowski, B. and Thornton, J. W. (**2004**). "Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous." *Nature*, 431(7011): 980–984.

Kolaczkowski, B. and Thornton, J. W. (**2008**). "A Mixed Branch Length Model of Heterotachy Improves Phylogenetic Accuracy." *Molecular Biology and Evolution*, 25(6): 1054–1066.

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

Leisch, F. (**2002**). "Sweave: Dynamic generation of statistical reports using literate data analysis." In W. Härdle and B. Rönz, editors, "Compstat 2002 - Proceedings in Computational Statistics," pages 575–580. Physika Verlag, Heidelberg.

Li, C. H., Lu, G. Q., and Orti, G. (**2008**). "Optimal data partitioning and a test case for ray-finned fishes (Actinopterygii) based on ten nuclear loci." *Systematic Biology*, 57(4): 519–539.

Lindsay, B. G. (**1995**). *Mixture models: theory, geometry and applications*. Institute of Mathematical Statistics American Statistical Association.

Lockhart, P., Novis, P., Milligan, B. G., Riden, J., Rambaut, A., and Larkum, T. (**2006**). "Heterotachy and tree building: A case study with plastids and eubacteria." *Molecular Biology and Evolution*, 23(1): 40–45.

Makelainen, T., Schmidt, K., and Styan, G. P. H. (**1981**). "On the Existence and Uniqueness of the Maximum Likelihood Estimate of a Vector-Valued Parameter in Fixed-Size Samples." *The Annals of Statistics*, 9(4): 758–767.

Mallows, C. (**1973**). "Some comments on C_p." *Technometrics*, 15: 661–675.

Mantel, N. (**1967**). "The detection of disease clustering and a generalized regression approach." *Cancer Research*, 27: 209–220.

Matsen, F. A. (**2006**). "A geometric approach to tree shape statistics." *Systematic Biology*, 55(4): 652 – 661.

Matsen, F. A. and Steel, M. (**2007**). "Phylogenetic mixtures on a single tree can mimic a tree of another topology." *Systematic Biology*, 56(5): 767–775.

McKenzie, A. and Steel, M. (**2000**). "Distributions of cherries for two models of trees." *Mathematical Biosciences*, 164(1): 81–92.

Miller, A. (**2002**). *Subset Selection in Regression*. Chapman & Hall, Boca Raton.

Mooers, A. O., Harmon, L. J., Blum, M. G. B., Wong, D. H. J., and Heard, S. B. (**2007**). "Some models of phylogenetic tree shape." In M. Steel and O. Gascuel, editors, "Reconstructing Evolution," Oxford, Oxford.

Mooers, A. O. and Heard, S. B. (**1997**). "Inferring evolutionary process from phylogenetic tree shape." *Quarterly Review of Biology*, 72(1): 31–54.

Mossel, E. and Vigoda, E. (**2005**). "Phylogenetic MCMC algorithms are misleading on mixtures of trees." *Science*, 309(5744): 2207–2209.

Mossel, E. and Vigoda, E. (**2006**). "Limitations of Markov chain Monte Carlo algorithms for Bayesian inference of phylogeny." *Annals of Applied Probability*, 16(4): 2215–2234.

Murtagh, F. (**1984**). "Complexities of hierarchic clustering algorithms: state of the art." *Computational Statistics Quarterly*, 1: 101–113.

Neininger, R. (**2002**). "The Wiener index of random trees." *Combinatorics Probability and Computing*, 11(6): 587–597.

Neyman, J. (**1971**). "Molecular studies of evolution: a source of novel statistical problems." In "Statistical decission theory and related topics," Academia Press, New York.

Nickrent, D. L., Blarer, A., Qiu, Y. L., Soltis, D. E., Soltis, P. S., and Zanis, M. (**2002**). "Molecular data place Hydnoraceae with Aristolochiaceae." *American Journal of Botany*, 89(11): 1809–1817.

Osborne, M., Presnell, B., and Turlach, B. (**2000**). "On the LASSO and its dual." *Journal of Computational and Graphical Statistics*, 9(2): 319–337.

Ota, R., Waddell, P. J., Hasegawa, M., Shimodaira, H., and Kishino, H. (**2000**). "Appropriate likelihood ratio tests and marginal distributions for evolutionary tree models with constraints on parameters." *Molecular Biology and Evolution*, 17(5): 798–803.

Pagel, M. and Meade, A. (**2004**). "A Phylogenetic Mixture Model for Detecting Pattern-Heterogeneity in Gene Sequence or Character-State Data." *Systematic Biology*, 53(4): 571–581.

Pagel, M. and Meade, A. (**2005**). "Mixture models in phylogenetic inference." In O. Gascuel, editor, "Mathematics of evolution and phylogeny," Oxford, New York.

Pagel, M. and Meade, A. (**2008**). "Modelling heterotachy in phylogenetic inference by reversible-jump Markov chain Monte Carlo." *Philosophical Transactions of the Royal Society B*, 363: 3955–3964.

Paradis, E. (**2006**). *Analysis of Phylogenetics and Evolution with R.* Springer, New York.

Paradis, E., Claude, J., and Strimmer, K. (**2004**). "APE: Analyses of Phylogenetics and Evolution in R language." *Bioinformatics*, 20(2): 289–290.

Penny, D. and Hendy, M. (**1986**). "Estimating the Reliability of Evolutionary Trees." *Molecular Biology and Evolution*, 3(5): 403–417.

Penny, D. and Hendy, M. D. (**1985**). "Testing Methods of Evolutionary Tree Construction." *Cladistics*, 1(3): 266–278.

Phillips, M. J., Delsuc, F., and Penny, D. (**2004**). "Genome-Scale Phylogeny and the Detection of Systematic Biases." *Molecular Biology and Evolution*, 21: 1455–1458.

Press, W., Teucholsky, S., Vetterling, W., and Flannery, B. (**1992**). *Numerical Recipes in C*. Cambridge University Press, New York, 2nd edition.

Pupko, T., Huchon, D., Cao, Y., Okada, N., and Hasegawa, M. (**2002**). "Combining multiple data sets in a likelihood analysis: Which models are the best?" *Molecular Biology and Evolution*, 19(12): 2294–2307.

R Development Core Team (**2007**). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

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

Rao, C. and Toutenburg, H. (**1995**). *Linear models: least squares and alternatives*. Springer, New York.

Renée, E. and Tillier, M. (**1994**). "Maximum likelihood with multiparameter models of substitution." *Journal of Molecular Evolution*, 39(4): 409–417.

Robinson, D. F. and Foulds, L. R. (**1981**). "Comparison of Phylogenetic Trees." *Mathematical Biosciences*, 53(1-2): 131–147.

Rogers, J. S. and Swofford, D. L. (**1999**). "Multiple local maxima for likelihoods of phylogenetic trees: a simulation study." *Molecular Biology and Evolution*, 16(8): 1079–1085.

Rokas, A., Williams, B. L., King, N., and Carroll, S. B. (**2003**). "Genome-scale approaches to resolving incongruence in molecular phylogenies." *Nature*, 425(6960): 798–804.

Ronquist, F., Larget, B., Huelsenbeck, J. P., Kadane, J. B., Simon, D., and van der Mark, P. (**2006**). "Comment on "Phylogenetic MCMC algorithms are misleading on mixtures of trees"." *Science*, 312(5772).

Saitou, N. and Nei, M. (**1987**). "The Neighbor-Joining Method - a New Method for Reconstructing Phylogenetic Trees." *Molecular Biology and Evolution*, 4(4): 406–425.

Sanderson, M. J. (**2002**). "Estimating absolute rates of molecular evolution and divergence times: A penalized likelihood approach." *Molecular Biology and Evolution*, 19(1): 101–109.

Schwarz, G. (**1978**). "Estimating the Dimension of a Model." *Annals of Statistics*, 6: 461–464.

Semple, C. and Steel, M. (**2003**). *Phylogenetics*. Oxford, Oxford.

Semple, C. and Steel, M. (**2004**). "Cyclic permutations and evolutionary trees." *Advances in Applied Mathematics*, 32(4): 669–680.

Shi, X., Gu, H., and Field, C. (**2008**). "Pattern Classification of Phylogeny Signals." *Statistical Applications in Genetics and Molecular Biology*, 7(1): Article 30.

Shimodaira, H. and Hasegawa, M. (**1999**). "Multiple comparisons of log-likelihoods with applications to phylogenetic inference." *Molecular Biology and Evolution*, 16: 1114–1116.

Simonoff, J. S. (**1996**). *Smoothing Methods in Statistics*. Springer, New York.

Sneath, P. H. and Sokal, R. R. (**1973**). *Numerical Taxonomy, the principles and practice of numerical classification*. W. H. Freeman, San Francisco.

Sokal, R. R. and Rohlf, F. J. (**1995**). *Biometry*. Freeman.

Stamatakis, A. (**2006**). "RAxML-VI-HPC: Maximum Likelihood-based Phylogenetic Analyses with Thousands of Taxa and Mixed Models." *Bioinformatics*, 22(21): 2688–2690.

Steel, M. (**2005**). "Should phylogenetic models be trying to 'fit an elephant?" *Trends in Genetics*, 21(6): 307–309.

Steel, M. and McKenzie, A. (**2001**). "Properties of phylogenetic trees generated by Yule-type speciation models." *Mathematical Biosciences*, 170(1): 91–112.

Steel, M. and Penny, D. (**2004**). "Two further links between MP and ML under the Poisson model." *Applied Mathematics Letters*, 17: 785–790.

Steel, M. A. (**1994**). "The maximum likelihood point for a phylogenetic tree is not unique." *Systematic Biology*, 43(4): 560–564.

Stefankovic, D. and Vigoda, E. (**2007**a). "Phylogeny of mixture models: Robustness of maximum likelihood and non-identifiable distributions." *Journal of Computational Biology*, 14(2): 156–189.

Stefankovic, D. and Vigoda, E. (**2007**b). "Pitfalls of heterogeneous processes for phylogenetic reconstruction." *Systematic Biology*, 56(1): 113–124.

Studier, J. A. and Keppler, K. J. (**1988**). "A Note on the Neighbor-Joining Algorithm of Saitou and Nei." *Molecular Biology and Evolution*, 5(6): 729–731.

Swofford, D. L. (**2002**). "PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods) 4.0 Beta."

Tarone, R. E. and Gruenhage, G. (**1975**). "Note on Uniqueness of Roots of Likelihood Equations for Vector-Valued Parameters." *Journal of the American Statistical Association*, 70(352): 903–904.

Tibshirani, R. (**1996**). "Regression shrinkage and selection via the Lasso." *Journal of the Royal Statistical Society Series B-Methodological*, 58(1): 267–288.

Titterington, D., Smith, A., and U.E., M. (**1985**). *Statistical analysis of fininte mixture distributions*. Wiley, Chichester.

Tuffley, C. and Steel, M. (**1998**). "Modelling the covarion hypothesis of nucleotide substitution." *Mathematical Biosciences*, 147: 63–91.

Tukey, J. W. (**1977**). *Exploratory data analysis*. Addison-Wesley.

Voelckel, C., Heenan, P. B., Janssen, B., Reichelt, M., Ford, K., Hofmann, R., and Lockhart, P. J. (**2008**). "Transcriptional and biochemical signatures of divergence in natural populations of two species of New Zealand alpine Pachycladon." *Molecular Ecology*, 17(21): 4740–4753.

Waddell, P. J. (**1995**). *Statistical methods of phylogenetic analysis: Including hadamard conjugation, LogDet transforms, and maximum likelihood*. Ph.D. thesis, Massey University.

Waddell, P. J., Kishino, H., and Ota, R. (**2007**). "Phylogenetic Methodology for Detecting Protein Interactions." *Mol Biol Evol*, 24(3): 650–659.

Waddell, P. J., Ota, R., and Penny, D. (**2008**). "Measuring Fit of Sequence Data to Phylogenetic Model: Gain of Power using Marginal Tests." *arXiv*, 0812.5005v1/q-bio.PE.

Waddell, P. J., Penny, D., Hendy, M. D., and Arnold, G. (**1994**). "The Sampling Distributions and Covariance-Matrix of Phylogenetic Spectra." *Molecular Biology and Evolution*, 11(4): 630–642.

Wu, W., Schmidt, T. R., Goodman, M., and Grossman, L. I. (**2000**). "Molecular evolution of cytochrome c oxidase subunit I in primates: Is there coevolution between mitochondrial and nuclear genomes?" *Molecular Phylogenetics and Evolution*, 17(2): 294–304.

Yang, Z. (**2000**). "Maximum Likelihood Estimation on Large Phylogenies and Analysis of Adaptive Evolution in Human Influenza Virus A." *Molecular Biology and Evolution*, 51: 423–432.

Yang, Z. (**2006**). *Computational Molecular evolution*. Oxford University Press, Oxford.

Yang, Z. H. (**1994**). "Maximum-Likelihood Phylogenetic Estimation from DNA-Sequences with Variable Rates over Sites - Approximate Methods." *Journal of Molecular Evolution*, 39(3): 306–314.

Yule, G. (**1925**). "A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S." *Philosophical Transactions of the Royal Society of London, B*, 213: 21–87.

Zou, H. and Hastie, T. (**2005**). "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society B*, 67(2): 301–320.

Zou, H., Hastie, T., and Tibshirani, R. (**2007**). "On the 'degrees of freedom' of the LASSO." *Annals of Statistics*, 35(5): 2173–2192.

Zwickl, D. J. (**2006**). *Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion*. Ph.D. thesis, University of Texas at Austin.