This dissertation is submitted for the degree of Doctor of Philosophy

Clustering approaches for incongruent phylogenies

Kevin Charles Gori

20th July 2016

Clare Hall College,
University of Cambridge

EMBL-EBI

DECLARATION

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text.

It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University of similar institution except as declared in the Preface and specified in the text.

It does not exceed the prescribed word limit for the relevant Degree Committee.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Phylogenetic inference from multiple loci can potentially result in inferring a more accurate tree than using a single locus. However, if the loci are incongruent, due to events such as incomplete lineage sorting or horizontal gene transfer, it can be misleading to describe the entire data set by a single tree.

In this thesis I describe processes that may lead to incongruence, and existing methods for analysing incongruent data. Many of these approaches are mechanistic in nature: that is, they aim to explain the incongruence with reference to particular evolutionary mechanisms. These approaches tend to be computationally challenging, and may not be robust in the presence of unmodelled evolutionary processes. I present a "process-agnostic" clustering approach to dealing with incongruence that is not predicated on the incongruence being caused by any particular mechanism.

I present a large-scale simulation study of combinations of inter-tree distances and clustering methods to infer sets of loci with shared underlying trees, and find that the best-performing combinations are inter-tree distances that explicitly account for branch lengths and topology, followed by spectral clustering or Ward's clustering method. I show that a statistical stopping criterion for determining the number of clusters present in a dataset, devised for this project, strongly outperforms the silhouette criterion, a general-purpose heuristic.

I illustrate the usefulness of this clustering approach by applying it to several real-world data sets: identifying orthology-calling errors in a yeast genomic dataset, and implicating incomplete lineage sorting as an incongruence-causing process in a dataset from *Chiastocheta* seed parasite flies.

Further developments that can be made to the clustering approach, building on the work in the initial chapters, are covered next. These concern ways to incorporate estimation error and uncertainty into the procedure, and involve developments in visualisation and algorithmics.

Finally, there is a round-up of the publications and projects that I contributed to during the course of my PhD that are not of direct relevance to the main narrative of my thesis.

The methods described are available in an open source software library called treeCl (`http://git.io/treeCl`).

1

INTRODUCTION

1.1 PHYLOGENETICS

The fundamental concept of evolutionary biology is that all forms of life on Earth are descended from a common ancestor, through a branching pattern of evolution. This idea originated in the 19th century, and was famously expressed in Darwin's 'On the Origin of Species' (Darwin 1859). The relationships among forms can be depicted, at least as a good approximation, as a tree (figure 1.1). In evolutionary biology such trees are called phylogenies, and the methods for inferring them phylogenetics.

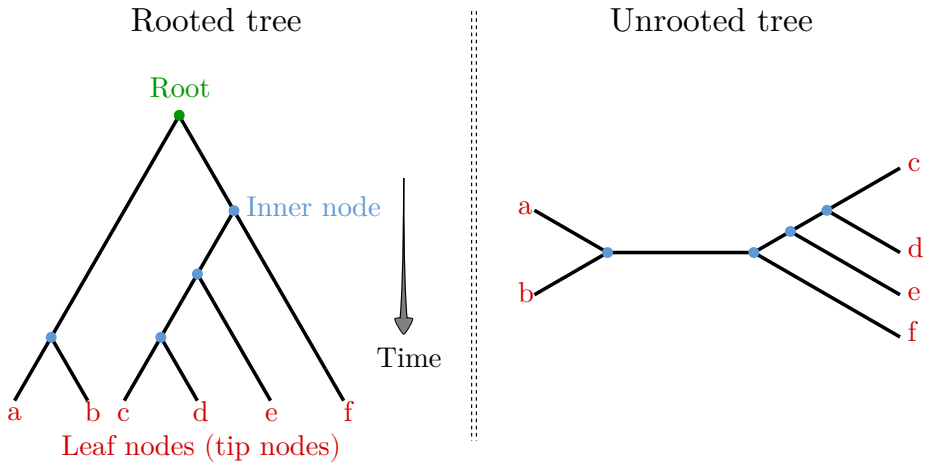


Figure 1.1. *Basic tree concepts. Both diagrams above show phylogenies depicting the relationships between six taxa, a–f. On the left is a rooted tree, which is aligned to a time axis. Nodes appearing higher up on the diagram are older than those lower down, with the root node being the very oldest. Branch lengths show the amount of divergence between nodes. Inner nodes represent ancestral forms, which are usually unobserved, while leaf, or tip, nodes show present day data. On the right is an unrooted tree depicting the same relationships among taxa. Unrooted trees depict the amount of divergence between taxa, but not their temporal relationships.*

Phylogenetic relationships are inferred using comparative methods: observations are made of features that are held in common by a set of different species, and a tree is fitted to account for their similarities and differences. Commonly, the features chosen for comparison are morphological or physiological characteristics. Modern molecular biological techniques that allow sequencing of protein and DNA have enabled phylogenetic inference through the comparison of genetic material.

1.1.1 MOLECULAR PHYLOGENETICS

Molecular sequence evolution is characterised by mutations, occurring over the course of generations, in the DNA sequences carried by organisms. There are three main types of mutation relevant to phylogenetic inference: changes of individual bases of DNA (substitutions), deletions of segments of DNA, and insertion of new segments (Nei and Kumar 2000). A mutation within a population can experience one of two fates: all individuals carrying it die out and the mutation is lost, or it spreads to all individuals in the population. In the second outcome the mutation has become *fixed*. Phylogenetics is primarily concerned with fixed mutations.

One way to model substitutions is to use probabilistic models to describe how characters states may change over time. Such models allow sequences evolution to be described in terms of probability, and for probabilistic inference to be used to build phylogenies. In this thesis, the focus is on maximum likelihood as the method of phylogenetic inference (see section 1.1.5), rather than alternatives such as Bayesian inference, distance estimation, or parsimony.

Insertions and deletions (indels) are accommodated in phylogenetics by aligning sequences prior to analysis. This essential initial step in tree inference identifies evolutionarily related segments of sequences by inferring the placement of missing stretches due to indels.

1.1.2 SEQUENCE ALIGNMENT

A sequence alignment is an arrangement of sequence information as a matrix. Rows contain sequences of characters derived from a particular taxon. Homologous characters are arranged in vertical alignment. The purpose of sequence alignment is to infer the homologous relationships between characters. There are several definitions of homology, but for all discussions of homology in this thesis I use the evolutionary definition: sequences are homologous if they are descended from a common ancestor. A discussion of different concepts of homology, as well as the aims and means of assessment of sequence alignment can be found in a book chapter I co-wrote during my PhD (Iantorno et al. 2014, see also chapter 5 and appendix A). The columns of an alignment, also termed 'sites' or 'characters', are the data points used to estimate phylogenies.

The simplest example of sequence alignment is in aligning a pair of sequences. The optimal alignment of a pair of sequences can be found using dynamic programming algorithms such as Needleman–Wunsch, for global alignment of entire sequences (Needleman and Wunsch 1970), or Smith–Waterman, for local alignment of subsequences (Smith and Waterman 1981). An application of pairwise alignment relevant to phylogenetics is in preparing a matrix of evolutionary distances between pairs of sequences, which can then be used to infer the phylogenetic tree using distance based methods such as least squares estimation (Fitch and Margoliash 1967; Cavalli-Sforza and Edwards 1967), or neighbour-joining (Saitou and Nei 1987; Gascuel 1997).

Aligning multiple sequences is more difficult. Although dynamic programming techniques used on pairs of sequences can be extended to arbitrary numbers of sequences (Needleman and Wunsch 1970), the computational effort required grows exponentially with the number of sequences, so exact alignment is not practical for more than three sequences (Murata, Richardson, and Sussman 1985). For larger numbers of sequences heuristic methods are used. The most widely used heuristic is progressive alignment (Ho-

geweg and Hesper 1984), in which successive sequences are aligned to a growing multiple alignment. Examples of this approach include Clustal (Thompson, Higgins, and Gibson 1994), Muscle (Edgar 2004), and Mafft (Katoh et al. 2005).

1.1.3 HOMOLOGY INFERENCE

Molecular phylogenetic inference only makes sense if the sequences it is applied to are homologous, i.e. related by evolutionary descent. Therefore, the sequences need to be chosen carefully to make sure that this is the case. There are different categories of homology. Orthology is the case when sequences are related entirely through speciation events in their evolutionary history. Contrasting this is paralogy, which is a relationship defined by duplications within a genome, such as in the evolution of gene families. Orthologous sequences should be used when the purpose of analysis is to uncover the species tree uniting a set of organisms (Fitch 1970). Developing methods to infer homology of different types is an active field of research (Kuzniar et al. 2008; Sonnhammer et al. 2014; Altenhoff et al. 2016). Orthology inference methods fall into two main categories: graph-based methods, and tree-based methods (Kuzniar et al. 2008).

Graph-based methods represent genes as nodes in a graph, connected by edges weighted by the estimated evolutionary distances between the pair of genes. By restricting the graph so edges only exist between the closest hits in each genome, and applying some clustering, cliques of orthologs can be identified. Examples of graph-based orthology inference methods are Ortho-MCL, InParanoid and OMA (Li, Stoeckert, and Roos 2003; Ostlund et al. 2010; Altenhoff et al. 2011).

Tree-based methods are based on reconciliation of gene trees with species trees. Gene trees may have different topologies to species trees due to heterogeneous evolutionary forces acting on particular genes. The name given to the mismatch between gene and species trees is *incongruence*. Discussion of the reasons for, consequences of, and methods for dealing with incongruence form the main subject of this thesis, and I will return to this in

section 1.2 and onwards. Reconciliation introduces a minimum number of duplications and losses to explain the incongruence between gene trees and species trees. Orthologous and paralogous relationships are then inferred based on whether sequences are related to each other through speciation or duplication nodes on the reconciled tree. Examples of tree-based methods include LOFT and Ensembl Compara Gene Trees (Heijden et al. 2007; Vilella et al. 2009).

Work in Orthology Inference

As a side project during my PhD, I used OMA to infer orthologous sequences among several model organisms, which I then used to infer a species tree, and to assess the suitability of ferrets and mice as models for respiratory disease in humans, all as part of the ferret genome project (Peng et al. 2014, see also chapter 5 and appendix B). In addition, in another side project, I developed the OMA FamilyAnalyzer software tool, which uses inferred orthology and paralogy relationships to display gene family evolution events on a tree (Altenhoff et al. 2015, see also chapter 5 and appendix C).

1.1.4 STATISTICAL MODELS OF SEQUENCE EVOLUTION

Substitutions can be accounted for by using a probabilistic model of evolution. A suitable type of model for this is the continuous-time Markov chain, which describes changes between discrete states over periods of time. Using these models, each site in the alignment is assumed to be evolving independently. This evolution is described by a Markov chain, with the characters at the site (one of four nucleotides for DNA data, one of twenty amino acids for protein data) being the states of the chain (Yang 2014). Markov chains satisfy the *Markovian property*, which means they have no memory: they are conditionally independent of the past. This means that the upcoming state of the chain is influenced only by the current state, and is not affected by the history of states that led up to the current one. A Markov chain is described by a matrix of instantaneous rates of change between states, $\boldsymbol{Q}$, in which q_{ij} describes the instantaneous rate of change from state i to state j.

The rows of $\boldsymbol{Q}$ are required to sum to zero; the diagonal entries, q_{ii}, ensure this. This is illustrated by the $\boldsymbol{Q}$ matrix of the simplest DNA model, JC69 (Jukes and Cantor 1969):

$$\boldsymbol{Q} = \{q_{ij}\} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{bmatrix} -3\lambda & \lambda & \lambda & \lambda \\ \lambda & -3\lambda & \lambda & \lambda \\ \lambda & \lambda & -3\lambda & \lambda \\ \lambda & \lambda & \lambda & -3\lambda \end{bmatrix} \end{array} \tag{1.1}$$

JC69 has a single parameter, the substitution rate, λ. More complex DNA models have $\boldsymbol{Q}$ consisting of multiple parameters that can be fitted to data. A particularly parameter-rich model is the general time-reversible (GTR) model (Tavaré 1986). The GTR model is illustrated in (1.2): $\boldsymbol{\pi}$ are the equilibrium frequencies of model states; a–f are the substitution rates. Diagonal entries are omitted for brevity (they are constrained such that the rows sum to zero).

$$\boldsymbol{Q}_{GTR} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{bmatrix} . & a\pi_C & b\pi_A & c\pi_G \\ a\pi_T & . & d\pi_A & e\pi_G \\ b\pi_T & d\pi_C & . & f\pi_G \\ c\pi_T & e\pi_C & f\pi_A & . \end{bmatrix} \end{array} \tag{1.2}$$

JC69 can be considered a special case of GTR, in which $\pi_T = \pi_C = \pi_A = \pi_G$ and $a = b = c = d = e = f$. Several other 'named' models can be obtained similarly, by constraining certain parameters of GTR to be equal. Examples include the K80, F81, and HKY85 models (Kimura 1980; Felsenstein 1981; Hasegawa, Kishino, and Yano 1985).

$\boldsymbol{Q}$ is used to calculate the transition probabilities, $p_{ij}(t)$, that state i will become state j over time period t. The p_{ij} are collected in matrix $\boldsymbol{P}(t)$:

$$\boldsymbol{P}(t) = \exp(\boldsymbol{Q}t) \tag{1.3}$$

Protein models

Protein models are constructed analogously to DNA models, but with twenty states in the model rather than four. Protein models have many more parameters than DNA models, so it is not typical to fit the parameters during analysis. Instead, pre-prepared empirical $\boldsymbol{Q}$ matrices are used. Empirical protein matrices, for example LG and WAG (Le and Gascuel 2008; Whelan and Goldman 2001), contain parameter estimates that have been fitted to very large data sets, and fixed. New data is analysed using these fixed values.

Reversibility

For a Markov chain to be time-reversible (often abbreviated to 'reversible') it must satisfy $\pi_i q_{ij} = \pi_j q_{ji}$ for all $i \neq j$. If this condition is met then the substitution process will look the same whether it is modelled forwards or backwards in time. One consequence of this is that trees built using these models are unrooted, as the model has no capacity to infer the placement of the root (except under the assumption of a strict *molecular clock*—see below). However, this limitation is outweighed by reversible models being mathematically more convenient to work with. The majority of substitution models implemented in popular programs such as PhyML and RAxML are time-reversible (Guindon et al. 2010; Stamatakis 2014). All substitution models considered in this thesis will be time-reversible.

Molecular clock

The strict molecular clock is the assumption that there is a constant substitution rate acting over the entire tree. Under this assumption, all tips are equidistant from the root, which allows its position to be found. Such clock-like trees are termed *ultrametric*. Real molecular data generally does not

conform to the strict clock assumption, except when sampled from closely related taxa (Reis, Donoghue, and Yang 2016). For example, the substitution rate among rodents is much higher than in primates (Wu and Li 1985), which leads the mammalian phylogeny to not be ultrametric.

1.1.5 LIKELIHOOD

The terms 'probability' and 'likelihood' are used interchangeably in everyday speech. In technical mathematical discussions they refer to two related, but different, ideas. A probability model is a structure that relates data, X, to a hypothesis θ, by the probability function $P(X|\theta)$: the probability of obtaining results X given the model parameters θ, according to the probability model. This is defined for any X that is a member of the set of possible results, and sums to 1 over the domain of X. The distinction between probability and likelihood is in the treatment of the two quantities X and θ. In probability, X is a variable and θ is constant, whereas in likelihood, θ is a variable, and X is constant. The likelihood, $L(\theta|X)$, of hypothesis θ given data X is proportional to $P(X|\theta)$ (Edwards 1992). Likelihood can be used for parameter inference, which involves finding values of θ that maximise the likelihood. This is termed 'maximum likelihood inference', or simply 'maximum likelihood'.

When maximum likelihood is used for phylogenetic inference, the data are the observed character states in columns of sequence alignments, and the parameters are those belonging to the substitution model (section 1.1.4), the branch lengths, and also the tree topology. Some parties argue that the tree topology is a (discrete) parameter of the model, to be optimised like any other parameter (Yang, Goldman, and Friday 1995); others consider the topology to be part of the model, with topology optimisation being a type of model selection (Yang and Rannala 2012). This rather academic distinction generally has no bearing on practical analysis. However, I will address a case in which the ambiguity in the role of the tree within the model is problematic when discussing model selection in section 2.5.3.

Likelihood calculation on a tree

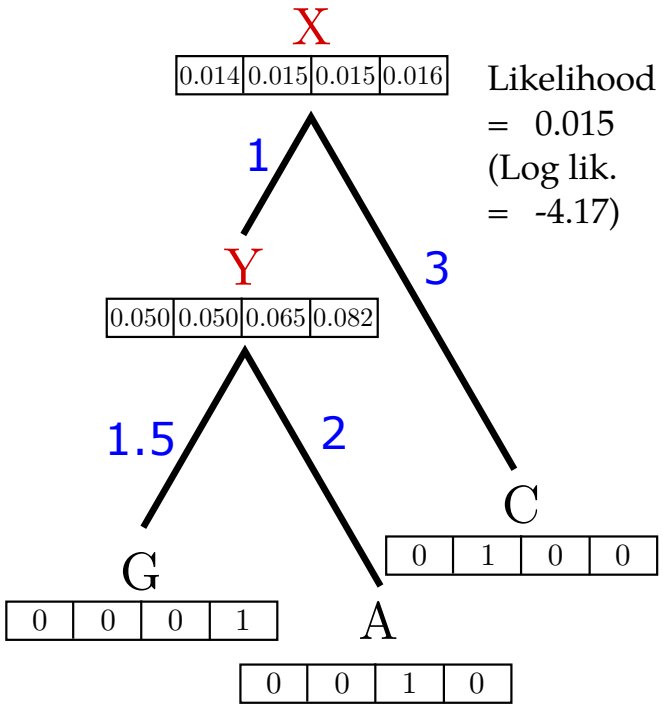


Figure 1.2. *Calculating likelihood on a tree. The diagram shows a tree of three tips relating a single site. One taxon carries the nucleotide 'G', one 'A' and the third 'C'. Blue numbers are branch lengths. The arrays at each node show the conditional likelihood vectors used in Felsenstein's pruning algorithm. The nucleotide states at the internal nodes, 'X' and 'Y', are unknown, so the likelihood is marginalised over all sixteen possible combinations of states. This example uses transition probabilities obtaied from the JC69 model of substitution.*

Figure 1.2 shows a tree relating three taxa, for a single site of an alignment. One taxon carries the nucleotide 'G', one 'A' and the third 'C'. There are two inner nodes, 'X' and 'Y'. The ancestral nucleotide states at these nodes are unobserved and unknown. Each of the inner nodes can take one of four nucleotide states, so there are sixteen possible combinations. The likelihood of this tree can be obtained by summing over all sixteen combinations, as in (1.4): x_i refers to the character state at node i, $L(\theta|\boldsymbol{x})$ is the likelihood of the tree given the site data $\boldsymbol{x} = \{G, A, C\}$, and the transition probabilities

of state x_i becoming state x_j over the branch length t_j ($p_{x_i x_j}(t_j)$) are derived from the substitution model, which in this example is JC69.

$$L(\theta|\boldsymbol{x}) = \sum_{x_X} \sum_{x_Y} [\pi_{x_X} p_{x_X x_Y}(1) p_{x_Y G}(1.5) p_{x_Y A}(2) p_{x_X C}(3)] \tag{1.4}$$

If the model of substitution is reversible, the placement of the root does not affect the likelihood of the tree.

Pruning algorithm

Felsenstein (1973) pioneered an efficient method—the famous 'pruning' algorithm—to calculate the likelihood for a tree topology. This is a dynamic programming algorithm in which, for each site, *conditional likelihood vectors* are calculated at each node of the tree (figure 1.2). These conditional likelihoods depend only on the node's descendants, so can be computed in a single, post-order, traversal of the tree. The conditional likelihood at a leaf is set directly from the data: if the data is a 'T', for example, then the likelihood of observing 'T' is 1, and the likelihood of observing one of 'C', 'A' or 'G' is 0. If the site is a gap then the site is treated as unknown data, and the likelihood of seeing 'T', 'C', 'A' or 'G' are all set to 1. The conditional likelihoods at internal nodes are calculated from their descendants. For example, the conditional likelihood at node i ($L_i(x_i)$) is obtained from its descendants j and k by the following equation (Yang 2014):

$$L_i(x_i) = \left[\sum_{x_j} p_{x_i x_j}(t_j) L_j(x_j)\right] \times \left[\sum_{x_k} p_{x_i x_k}(t_k) L_k(x_k)\right] \tag{1.5}$$

The likelihood of the tree for a single site is obtained by a weighted average of the conditional likelihoods at the root (L_0), weighted by the nucleotide frequencies π_{x_0}:

$$L(\theta|\boldsymbol{x}) = \sum_{x_0} \pi_{x_0} L_0(x_0) \tag{1.6}$$

Due to the assumption of independence of sites, the overall likelihood is the product of the sitewise likelihoods, and consequently the log-likelihood is

the sum of the sitewise log-likelihoods. Using log-likelihoods in the computation helps prevent numerical errors that can occur when computers store very small numbers. The pruning algorithm offers substantial computational savings over the first method described above, which sums the likelihood over all possible combinations of ancestral states. The pruning algorithm enables the likelihood to be computed in time proportional to the number of nodes in the tree, which grows linearly with the number of taxa, rather than time proportional to the number of combinations of inner node states, which grows exponentially. The likelihood framework enables substitution parameters, branch lengths and the topology to be optimised. Maximum likelihood parameter estimates are obtained using numerical optimisation techniques (Yang 2014), while the maximum likelihood tree topology is usually found by used tree search heuristics (Whelan 2007).

Tree search

Finding the optimal tree topology is difficult because of the sheer number of possible trees. The number of distinct tree topologies grows explosively with the number of taxa, which rules out exhaustive search for more than a handful of taxa (see table 1.1). Instead, the topology is searched for using heuristic tree search methods. A starting tree is selected, either constructed randomly or inferred from the data using a fast method. Tree search proceeds by making small rearrangements to this tree to obtain 'neighbouring' trees, which are tested to see if they offer an improved fit to the data. This continues until no better tree topology can be found. Commonly used rearrangements include nearest neighbour interchange (NNI) and subtree prune and regraft (SPR) (Lakner et al. 2008; Guindon and Gascuel 2003; see also figures 1.3 and 1.4).

1.2 INCONGRUENCE

One of the chief applications of molecular phylogenetics is to infer the species tree depicting the relationships of a group of organisms. These trees are inferred by comparative analysis of a sample of the organisms' genetic

Taxa	Number of unrooted topologies
3	1
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025
11	34459425
12	654729075

Table 1.1. *How the number of tree topologies increases with taxa. The number of unrooted topologies for n taxa is* $(2n-5)!!$.

sequences. Over the years the size of the genetic sequence sample typically available for any given organism has grown, due to fast-paced technological improvements in sequencing. In the early days of phylogenetics, trees would be inferred based on the sequence of a single locus (Fitch and Margoliash 1967; Neyman 1971; Kashyap and Subas 1974); however, contemporary studies may use thousands of loci (e.g. Salichos and Rokas 2013; Jarvis et al. 2014). Such 'phylogenomic' approaches were touted as an end to incongruence (Rokas et al. 2003), which, put simply, is the tendency for phylogenetic trees inferred from separate sources of information to differ. Trees inferred from single locus alignments often differ due to the stochastic variation inherent to inferences made from limited amounts of information; multilocus phylogenetics was expected to overcome that uncertainty by sheer volume of data. However, incongruence persists even in the phylogenomic world (Jeffroy et al. 2006), and ever larger datasets are not sufficient to resolve it (Philippe et al. 2011). It seems that incongruence is a natural consequence of the evolutionary processes that shape sequence variation over time, and as such it should be explicitly accounted for when inferring evolutionary relationships.

Molecular phylogenetic methods infer the evolutionary history of homologous sequences. Due to the modern abundance of sequencing data, trees are increasingly inferred by jointly analysing sequences from multiple loci. This is expected to deliver better-resolved and less-biased inferences by averaging out uncertainty over a greater amount of data (Pamilo and Nei 1988; Delsuc, Brinkmann, and Philippe 2005).

There are a number of methods for multilocus phylogenetic analysis (see reviews by Bininda-Emonds, Gittleman, and Steel 2002; Queiroz and Gatesy 2007; Liu et al. 2009). These proceed by inferring the single evolutionary tree that best fits the entire dataset, effectively 'averaging' the signal from multiple loci. The presumption here is that these loci have experienced the same evolutionary history. However, this is not necessarily the case (see section 1.3 for details). When a dataset comprises multiple loci, the trees derived from individual loci have the capacity to be incongruent with each other (Jeffroy et al. 2006). A key question here is whether the incongruence is meaningful—whether it indicates a real underlying difference in the evolution of distinct genomic loci—or it is due to sample variation. Building a single summary tree from multiple loci implicitly assumes the latter: that each locus is a noisy estimate of the same underlying tree. If our main concern is uncovering the species tree that relates a set of organisms, then incongruence is an inconvenience that we would prefer to integrate out, as resolution of deep phylogenies may not be possible without minimising the impact of incongruence on species tree inference (Nosenko et al. 2013).

Alternatively, aside from (or in addition to) inferring the species tree, we may want to understand the histories of the parts of the genome that have been formed under different evolutionary trajectories. With this aim in mind, I focus on isolating clusters of incongruent loci for further investigation, under an alternative approach to studying incongruence—that of identifying 'cluster trees', distinct from species trees.

1.3 PROCESSES LEADING TO INCONGRUENCE

There are a variety of processes that can lead to different regions of a genome having different histories (Leigh, Lapointe, et al. 2011): horizontal gene transfer (HGT), hybridisation, incomplete lineage sorting (ILS), recombination, and differential gene duplication and loss (GDL), among others. If we believe such processes have occurred, then we should expect that the trees derived from different loci could be incongruent with one another. Consequently, "summary" trees inferred from the entire dataset may be only partially representative or, in the worst case, not representative of the evolution of any locus. Because this is a systematic error, rather than noise, we cannot expect it to be reduced by adding more data (Philippe et al. 2011). If we believe there is real heterogeneity in the evolutionary process that produced the genomes, and incongruence is an indication of this, then we should look for ways of partitioning multilocus data into groups that are related by the same history (Bull et al. 1993; Huelsenbeck et al. 1994; Cunningham 1997; Waddell, Kishino, and Ota 2000).

1.3.1 HORIZONTAL GENE TRANSFER

Horizontal gene transfer (HGT) is a term that describes the transfer of genomic DNA through means other than parent–child transfer (Ravenhall et al. 2015). It is widely observed in bacteria (Beiko, Harlow, and Ragan 2005; Whidden, Zeh, and Beiko 2014), as well as in plants (Rice et al. 2013; Xi et al. 2013), in which it may occur through "organelle-capture" mechanisms distinguishable from introgressive hybridisation (Stegemann et al. 2012; see also section 1.3.3). HGT is also frequently observed among the eukarya that live within the plant ecosphere (Paganini et al. 2012), and among yeasts (Wu, Buljic, and Hao 2015). There is limited evidence for HGT occurring among vertebrates (Crisp et al. 2015). HGT creates a characteristic subtree prune and regraft change to the shape of a phylogenetic tree (figure 1.3). SPR also has wide use as a tree search heuristic in tree optimisation algorithms (Lakner et al. 2008).

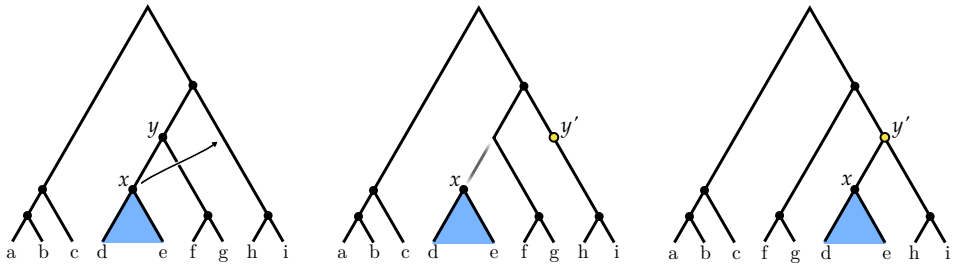


Figure 1.3. *Subtree prune and regraft. The subtree at* x *is to be pruned from the tree and regrafted at another edge. A new node,* y′ *is inserted at the regraft point. The edge between* x *and* y *and the node* y *are deleted, and a new edge is added between* x *and* y′*.*

1.3.2 INCOMPLETE LINEAGE SORTING

Lineage sorting refers to the transmission of genetic polymorphisms among dividing populations. Lineage sorting can be complete if a gene is polymorphic in the ancestral population but monomorphic in each subpopulation after population division, and incomplete otherwise. Incomplete lineage sorting (ILS) is thus characterised by the maintenance of ancestral, pre-speciation polymorphisms in populations after speciation. Rapid radiations of species are particularly likely to be accompanied by ILS, as the timescale between population divisions is so short that polymorphisms do not go to fixation in the intervening time. ILS has been observed among the recently radiating great apes clade (Hobolth et al. 2011; Scally et al. 2012). ILS typically produces local changes to a phylogenetic tree, in which nodes exchange among their nearest neighbours, a process called NNI (figure 1.4). Like SPR, NNI is used as a tree search heuristic (Guindon and Gascuel 2003; Whelan 2007).

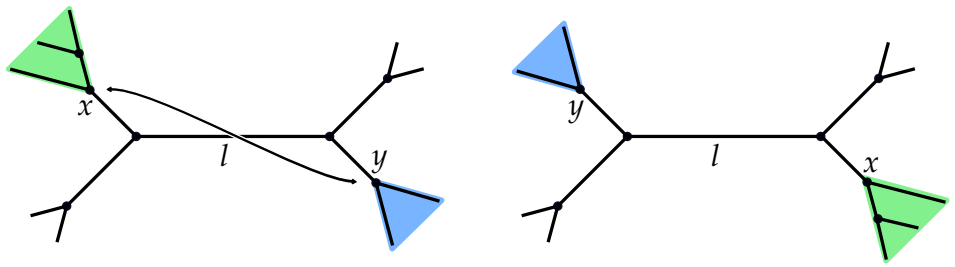


Figure 1.4. *Nearest neighbour interchange. The subtrees at* x *and* y*, which are separated by branch* l*, are interchanged.*

1.3.3 HYBRIDISATION

Hybridisation occurs when individuals from different species mate and produce offspring. In many cases the offspring are infertile, but in some cases they can back-cross with one of the parent species. In this way genes from the population of one species may be brought into the population of another, a process called introgression (figure 1.5). Introgressive hybridisation has been used as a model to explain incongruence in cases where, due to the population dynamics and timescales considered, ILS is not a good fit (Köhler and Deein 2010). In general it is difficult to distinguish between ILS and hybridisation; methods that do so may assume a molecular clock (Sang and Zhong 2000), or a coalescent model (Kubatko 2009).

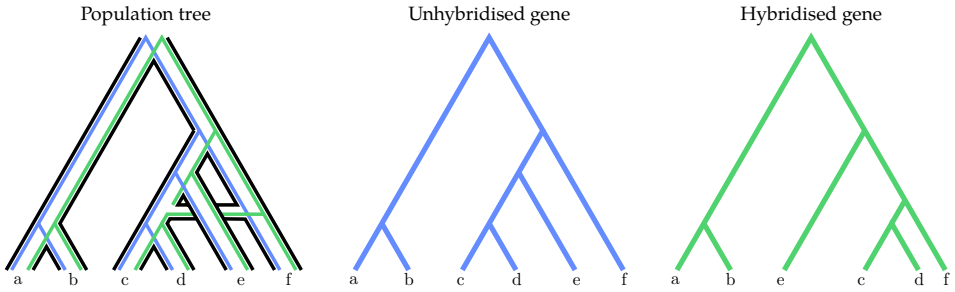


Figure 1.5. *Hybridisation. Ancestral lineages hybridise and exchange genetic information (horizontal section of left-most tree). Not all loci are affected. In this example, the tree defined by black bars represents the evolution of a population. Within the population, the blue lines represent the evolution of a locus unaffected by hybridisation, and the green lines represent a locus that was affected by hybridisation. The resulting trees for these loci are shown to the right, in the same colours.*

1.3.4 DIFFERENTIAL GENE DUPLICATION AND LOSS

Genes often occur as members of gene families, which arise when genetic loci are duplicated within a genome, so that genomes contain multiple related paralogous genes derived from a common ancestor. Initially, the duplicated copies will likely overlap substantially in function, although they may differ due to their genetic location in regulation and tissue expression levels. The overlap in function introduces redundancy, which allows a level of freedom to evolve to develop in one or both of the copies, and they may undergo rapid evolution due to the relaxation of constraints. The new forms may be fixed in the population, e.g. through neofunctionalisation or

subfunctionalisation (Force et al. 1999), or they may be lost. Subsequent differential gene losses between populations can be problematic in phylogenetic reconstruction, due to the difficulty of distinguishing between paralogs and orthologs (Dessimoz et al. 2006; see also figure 1.6 for an example). In this way, differential gene duplication and loss (GDL) contributes to phylogenetic incongruence (Scannell et al. 2006).

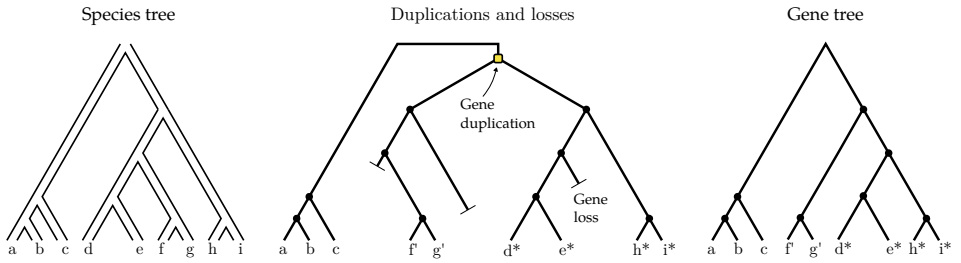


Figure 1.6. *Differential gene duplication and loss. After a gene duplication followed by a sequence of losses, the gene tree topology (right) is incongruent with the species topology (left). Clades marked with an asterisk are paralogous to clades marked with a prime.*

1.4 MODELLING INCONGRUENCE

Many methods dealing with incongruence make explicit assumptions about its biological basis. Such "mechanistic" approaches have been developed to model HGT (Hallett and Lagergren 2001; Dessimoz, Margadant, and Gonnet 2008; Abby et al. 2010), ILS (Rannala and Yang 2003; Heled and Drummond 2010), recombination (Kosakovsky Pond et al. 2006), GDL (Chen, Durand, and Farach-Colton 2000; Boussau et al. 2013; Wu et al. 2013), and combinations of processes such as combined ILS/GDL models (Bansal, Burleigh, and Eulenstein 2010; Szöllősi and Daubin 2012), or HGT/GDL models (Doyon et al. 2010; David and Alm 2011; Nguyen et al. 2012). However, mechanistic approaches can be computationally prohibitive and may not be robust to other, unmodelled, sources of incongruence. As we have seen in section 1.3, these are numerous.

I focus my attention on an alternative class of methods that I describe as "process-agnostic". These aim to detect the existence and extent of any significant incongruence within a dataset, without making any assumptions

about its biological basis, in the hope of avoiding introducing problems due to model misspecification. Existing process-agnostic approaches take the form of either statistical tests of incongruence (Planet 2006; Leigh, Lapointe, et al. 2011), or clustering approaches that partition datasets into groups that are cohesive and self-similar (Ané et al. 2007; Nye 2008; Leigh, Schliep, et al. 2011).

Nye's Tree of Trees (2008) summarises the phylogenetic similarities among genes as another tree, termed a meta-tree, where the tips correspond to the different, incongruent trees derived from multilocus data, and internal nodes represent the consensus of their child trees. The length of the edge between two nodes is analogous to the distance between the trees represented by each node, with inter-tree distances calculated using the Robinson-Foulds metric (Robinson and Foulds 1981; see section 2.2 for the details of this metric). The meta-tree is inferred using an algorithm similar to neighbour-joining (Saitou and Nei 1987). One application of this approach has been in qualitatively assessing the strength of incongruent phylogenetic signal in multilocus data (Nye 2008): Tree of Trees is applied to bootstrap replicates of loci. If the replicates for one locus, or set of loci, together form a defined clade in the tree of trees, then this is evidence that they evolved according to a distinct tree. Alternatively, if the tree of trees is starlike, and bootstrap replicates are not monophyletic according to the locus they are derived from, this represents a lack of evidence for incongruence. Arguably it is inappropriate to represent the pattern of incongruence in trees as a tree, because this implies an ancestor-descendant relationship between nodes. It is hard to see how this could be realistic. Potential limitations of Tree of Trees are that it uses a greedy clustering approach, and its reliance on only the Robinson-Foulds tree distance metric. These both limit the extent to which it explores all the possible partitions of the data.

Similarly, Conclustador (Leigh, Schliep, et al. 2011) uses inter-tree distances as a basis for clustering. Trees are compared using a novel Euclidean distance among bipartitions weighted by bootstrap support, with clustering performed by either a version of the k-means algorithm (MacQueen 1967),

or a hybrid spectral clustering (Kaufman and Rousseeuw 1987; Zelnik-Manor and Perona 2004) and hierarchical clustering method. The number of clusters is selected using the CH index (Caliński and Harabasz 1974) or the eigengap heuristic (Luxburg 2007) for k-means and for spectral clustering, respectively. A conceptually similar method is PhyBin (Newton and Newton 2013), which can either identify genes with topologically identical trees or perform hierarchical clustering on the Robinson-Foulds distance matrix between every tree. These methods again explore only a limited way of partitioning the data, and the use of heuristic methods for selecting the number of clusters may not be effective.

Statistical binning (Mirarab et al. 2014) uses a graph-based algorithm to divide a set of genes into a number of approximately equal-sized bins of phylogenetically compatible genes (Warnow 1994). This has been used as a preprocessing step to coalescent species tree estimation, to reduce the run time of analyses, and increase its accuracy in the presence of ILS (Mirarab et al. 2014). This approach conforms to the 'integrating out' approach to incongruence, and has not been used to reveal anything about the loci that exhibit incongruence. Instead the purpose of binning is to reduce the computational burden of analysing vast datasets: instead of analysing every gene together, the input of a coalescent analysis are reduced to the set of summary trees from each bin.

Whidden and Matsen (2015) describe a method for computing the pairwise approximate SPR distances amongst a set of trees. This is used to infer clusters of trees within the credible set of trees produced by Bayesian tree building programs such as MrBayes (Ronquist et al. 2012) or PhyloBayes (Lartillot et al. 2013). The primary aim is to map the landscape of tree-space in a Monte-Carlo scenario, but the approach could conceivably also be used to assess incongruence within a set of gene trees. The network of trees can be visualised using Cytoscape (Shannon et al. 2003), or by multidimensional scaling (Torgerson 1952). Such visualisation techniques are discussed further in Chapter 4.

BUCKy (Larget et al. 2010) uses a Bayesian probabilistic framework to estimate a gene-to-tree map that assigns each gene to one of the $(2n-3)!!$ possible unrooted trees of n taxa (Felsenstein 2004). A Dirichlet process prior (Ferguson 1973; Antoniak 1974) is used to determine the total number of distinct trees represented by the gene-to-tree map. As an exhaustive treatment of tree-topologies is required, this is only of practical use for problems of modest size. For instance, the authors describe examining a dataset of 106 genes from 8 species.

These existing methods have in common that they each adopt a specific clustering procedure. There are, however, many potential distance measures and clustering algorithms, and we know almost nothing about their relative performance in identifying genes that share common evolutionary histories under plausible biological scenarios. For instance, the Robinson-Foulds distance used in Tree of Trees ignores any difference in branch lengths among trees, yet these might provide useful information in the context of ILS; the Dirichlet process prior in BUCKy tends to result in uneven cluster sizes (Larget et al. 2010), yet this might be suboptimal in the context of recombination. Furthermore, the problem of determining the optimal number of clusters remains poorly understood, with methods providing no or only generic solutions.

1.5 THESIS GOALS AND ORGANISATION

Incongruence, in which different sources of evidence lead to the inference of different evolutionary trees, is a well-known problem in molecular phylogenetics. It was expected that multi-locus phylogenetics, or phylogenomics, would reduce the amount of incongruence observed through sheer amount of data, but that has not been the case. Incongruence is a reflection not only of stochastic variation, but also has an evolutionary basis through numerous processes. In the course of evolutionary influence, incongruence can be dealt with mechanistically, in which a model explicitly accounts for one or more incongruence-causing processes of evolution, or it can be approached in a "process-agnostic" way, in which no particular mechanism is assumed

to account for all the incongruence observed. In this thesis, I choose to take a process-agnostic approach based on clustering.

The thesis is organised as follows: Chapter 2 describes this process-agnostic approach, and tests its performance on extensive simulated data. Chapter 3 covers the application of Chapter 2's methods to empirical data: sets of orthologous sequences derived from ascomycetous yeasts and from *Chiastocheta* genus globeflower flies. Chapter 4 discusses further developments of the method and their application to the visualisation of the evolutionary relationships between large numbers of loci, and to optimising the clustering process. My conclusions are presented in Chapter 6.

In the next chapter I will describe a general approach to clustering multilocus data based on phylogenetic congruence. I will look at some specific tree distance metrics and clustering methods that can be used to implement this approach, and I also introduce a procedure based on likelihood ratio testing to infer the optimal number of clusters. I will present a number of results obtained from applying these methods to simulated data.

2

CLUSTERING BY CONGRUENCE

This chapter describes an approach for clustering loci into congruent clusters, by minimising within-cluster incongruence compared to between-cluster incongruence. This process is intended to be agnostic to the underlying process of evolution, being applicable to data whatever the evolutionary processes are that have shaped them. The details of the clustering approach are introduced in section 2.1.

This approach requires the estimation of distances between the evolutionary trees for each locus (which are estimated independently). Several distance metrics have been proposed to measure the distance between two trees, and I present a summary of the more prominent ones in section 2.2. The chapter continues with a synopsis of some clustering methods that can be applied to the distances, which are described in section 2.3.

I tested my clustering approach through extensive simulation studies. Initially, my intention with these studies was simply to verify that this clustering approach can work, but once I had demonstrated this to my own satisfaction I extended their scope. I tested different combinations of three distance metrics and seven clustering methods, in order to see if any of them were particularly effective, or ineffective, to guide the development of the method. This simulation design and its results are presented in section 2.4.

I also devised a statistical method based on likelihood ratio testing that can be used as a stopping criterion to determine the appropriate number of clusters to divide a dataset into. The stopping criterion uses a likelihood-based measure of cluster evaluation, which is introduced in section 2.5. In the same section I describe the details of the stopping criterion in two vari-

ants, which are based on parametric bootstrapping and permutation, respectively. I used simulation to test the performance of both variants of the stopping criterion. I contrast their performance with a general purpose method for determining the number of clusters (section 2.5).

I end the chapter with an analysis of the effect of missing data on the clustering method, by extending the simulation studies to generate sequence datasets in which a proportion of locus sequences are entirely missing from the multiple sequence alignments. This is more extreme than indels causing certain sites to be absent. I show that the method is robust to a large amount of missing data, aiding its suitability for use on real biological data, which frequently has missing data. The application of the clustering approach to real data is covered in Chapter 3.

The aim of this chapter is to provide an appraisal of the congruence-based clustering approach. In particular, to investigate which particular clustering implementations are the most effective, what the most important considerations are when assessing the differences between evolutionary histories of loci, and whether an appropriate number of different histories can be estimated, sufficient to describe the disparate evolutionary paths that underlie genomic data.

2.1 CLUSTERING PROTOCOL

The clustering approach investigated here takes a set of sequence alignments, and partitions them into disjoint subsets. The intention is that, following partitioning, each cluster should contain loci that have minimal incongruence, and putatively share a common phylogenetic history. The assignment of loci into clusters is done on the basis of how similar their evolutionary trees are, and is agnostic to any particular evolutionary mechanisms that may be causing incongruence.

Throughout the thesis I will describe such division of a dataset as a partition, and the resulting subsets as clusters. Furthermore, each alignment will be referred to using the general term 'locus', as the nature of the se-

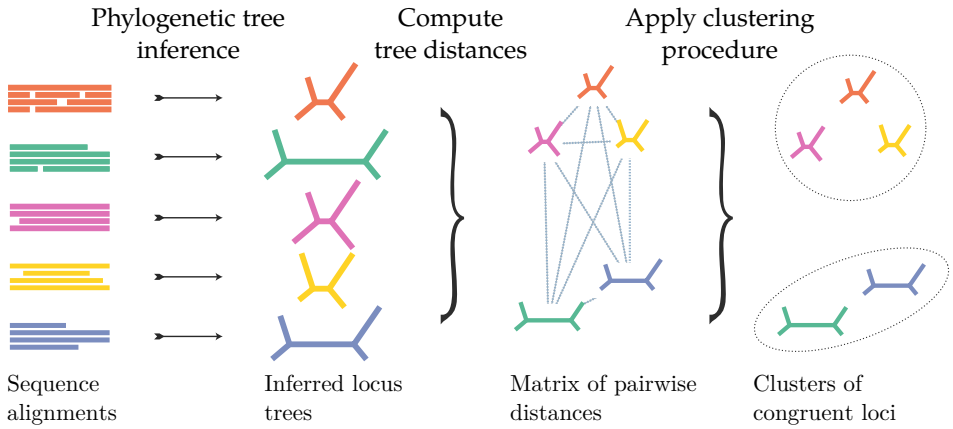


Figure 2.1. *Overview of the clustering process. From left to right: input alignments are read; trees are inferred independently from the alignments; inter-tree distances are computed and used as the basis for clustering. Further procedures are used to re-estimate one tree for each cluster and to choose the optimal number of clusters.*

quences (for example protein-coding gene, intron, pseudogene, or another category) is not of primary importance.

The approach is a three step pipeline (see figure 2.1 for a graphical representation):

1. Infer a single phylogenetic tree for each input sequence alignment.
2. Gauge the level of evolutionary similarity among loci by measuring distances between pairs of trees; for a set of n trees this requires $n(n-1)/2$ computations.
3. Apply a clustering algorithm to the distances to generate a set of clusters. This requires that the number of clusters be a fixed value. Tests introduced in section 2.5 provide a reasoned approach to choosing this number.

2.1.1 SOFTWARE IMPLEMENTATION

I have implemented the pipeline in a python package, treeCl[*]. This package handles all of the tasks required for the analyses presented in this thesis, such as inferring evolutionary trees from sequence alignments, calculating

[*] `http://git.io/treeCl`

inter-tree distances, clustering, concatenating alignments and computing likelihoods used to score partitions. As well as being able to run on a single core, it also can distribute tree estimation and phylogenetic distance calculations for parallel execution on a single machine or on a computer cluster.

Input data are several sets of DNA or protein sequences obtained from a number of taxa (in principle the data can be a mixed set of DNA and protein sequences, however in all the results presented in this thesis the data were of a homogeneous type). Each set of sequences is derived from a particular genetic locus that exists in all (or a subset) of the taxa. It is assumed that these have been filtered so that they contain orthologous sequences, and aligned, prior to analysis. The initial step is to infer a tree for each locus. Any method that produces trees with branch lengths can be used. In my work I have used maximum likelihood tree estimation, in which trees are inferred under probabilistic models of sequence evolution (Felsenstein 2004). The inferences were carried out using RAxML (Stamatakis 2014) and PhyML (Guindon et al. 2010); other likelihood calculations were performed using the Phylogenetic Likelihood Library (Flouri et al. 2015), Bio++ (Guéguen et al. 2013) and my own custom code in treeCl.

2.2 TREE DISTANCE METRICS

The following is a selection of metrics that have been proposed to compare tree structures, including the ones that were used in the clustering investigation.

2.2.1 ROBINSON-FOULDS DISTANCE

The Robinson-Foulds distance (Robinson and Foulds 1981) is a tree topology-sensitive measure of dissimilarity between trees. It can be thought of in two equivalent ways: either as the number of operations required to transform one tree into another, where the two operations considered are (a) contracting a branch to zero length and merging its nodes, and (b) expanding a node (see figure 2.2); or as the number of splits that occur in only one tree (the symmetric difference of the sets of splits defined by each

tree) — each edge in the tree defines a split, in that removal of the edge would divide the leaves of the tree into two subsets. The distance is a metric (Robinson and Foulds 1981), and can be computed in linear time (Day 1985).

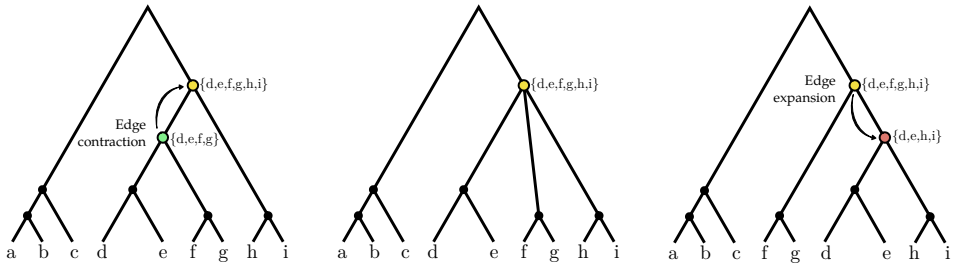


Figure 2.2. *Robinson-Foulds operations. Two Robinson-Foulds operations are required to convert between the trees from figure 1.3: one contraction (left, middle above), and one expansion (middle, right). The set of descendants of internal nodes are illustrated in braces.*

2.2.2 EUCLIDEAN DISTANCE

The "branch-score" distance (Kuhner and Felsenstein 1994) is an application of the Euclidean distance to trees. A pair of trees is represented as two vectors (one for each tree), each of which contains an entry for each split present in either tree. The entries of each vector are the branch lengths for each split in each tree, which is set to zero if the split doesn't exist in the tree. The branch score distance — hereafter referred to as the Euclidean distance to reflect the nomenclature used by Sukumaran and Holder (2010) — is then the Euclidean distance between the vectors: the square root of the sum of the squares of the element-wise difference between the vectors.

2.2.3 WEIGHTED ROBINSON-FOULDS DISTANCE

There also exists a variant of Robinson and Foulds' distance that is sensitive to branch lengths, in which the quantities summed are the absolute differences between branch lengths (Robinson and Foulds 1979). Both this and the Euclidean distance are instances of the general Minkowski distance which is the k-th root of the sum of the absolute differences raised to the k-th power, with weighted Robinson-Foulds having k=1 and the Euclidean distance having k=2 (Kuhner and Felsenstein 1994). This is analogous to

the difference between the Manhattan (city-block) and Euclidean distances in Euclidean spaces.

2.2.4 GEODESIC DISTANCE

The geodesic distance metric on phylogenetic trees restricts the distance between trees to the geometry of the space of phylogenetic trees (Billera, Holmes, and Vogtmann 2001). This space consists of a number of multi-dimensional orthants, with each edge length in the tree being a measurement along a dimension in the orthant. There is a separate orthant for each possible fully-resolved tree topology, and the orthants are mathematically "glued together" so that neighbouring orthants differ by a single NNI operation (figure 1.4). Partially resolved trees (in which some internal edges are missing, or have a length of zero) occupy a position on the boundary between orthants. The degenerate case, the star topology (all internal edges are missing or have length zero), exists at a vertex connecting all orthants. Thus, an upper bound of the geodesic distance is the length of the path from one tree, through the star topology vertex, and extending to the other tree. A lower bound is obtained when the trees share the same topology, and is then equal to the Euclidean distance. Finding the minimum geodesic path can be computed in polynomial time (Owen and Provan 2011). The geodesic distance explicitly accounts for both differences in topology and branch lengths.

2.2.5 OTHER DISTANCES

Many alternative metrics have been proposed, mostly topology-based, i.e. neglecting branch lengths.

Quartet distance

The quartet distance (Estabrook, McMorris, and Meacham 1985) is based on the idea of decomposing a tree of at least four tips into all the relationships that it specifies between all combinations of its tips taken four at a time, which are its *quartets*. The quartet distance between two trees is the proportion of the $\binom{n}{4}$ quartets that differ between two trees of n taxa. It is a

topology-sensitive measure that does not take branch lengths into account. It is more sensitive to partial similarities of trees than the Robinson-Foulds distance (Felsenstein 2004:530), and fast methods exist for its calculation (Crosby and Williams 2012).

NNI and SPR distances

The NNI distance between two trees is the minimum number of nearest neighbour interchange operations required to transform one into the other (and the SPR distance is analogous to this, using SPR operations in place of NNI operations). Determining this minimum number is an NP-complete problem, as proven in Li and Zhang (1999), who also provide a polynomial-time approximation. Approximate methods can also be used to estimate the SPR distance in polynomial time (Whidden, Beiko, and Zeh 2013; Chung, Perna, and Ané 2013; Whidden, Zeh, and Beiko 2014; Whidden and Matsen 2015).

Boot-split distance

The boot-split distance is a derivative of the Robinson-Foulds distance (also called the "split distance") in Puigbò, Garcia-Vallvé, and McInerney (2007). It accounts for the uncertainty of the trees, by giving greater weight to similarities among edges with high support values. It is calculated according to equation (2.1), in which: e is the sum of bootstrap values of shared splits; d is the sum of bootstrap values of unshared splits; a is the sum of all bootstrap values; M_e is the mean bootstrap value of equal splits; M_d is the mean bootstrap value of different splits. If there are no shared splits, e and M_e are taken to be zero, similarly with unshared splits (Puigbò, Wolf, and Koonin 2012).

$$BSD = \frac{1}{2}\left[1 - \frac{e}{a}M_e + \frac{d}{a}M_d\right] \tag{2.1}$$

Matching

'Matching' constructs a bipartite graph of the splits in the two trees being compared (Lin, Rajan, and Moret 2012). A bipartite graph is one in which its vertices belong to two disjoint sets, U and V, and each edge is a connection between U and V. A perfect matching, in terms of a bipartite graph, connects all nodes in U and V exactly once. If U and V are the sets of splits in two trees, then the matching distance between them is the minimum weight perfect matching under a weighting scheme in which the weight of an edge is the Hamming distance between vectors representing the split. The matching distance can be made equivalent to the Robinson-Foulds distance by using a weighting scheme in which the weight of an edge is zero if the connected splits are identical, and two otherwise.

Path length distance

The path length difference (Penny, Watson, and Steel 1993) computes the number of branches that separate each pair of species in any given tree. The path length difference distance between two trees is the square root of the squares of the differences between these numbers (Felsenstein 2004:531). A similar approach has been applied to rooted trees (Kendall and Colijn 2015), and can also be extended to incorporate branch lengths.

The above metrics form a selection of some of the most prominent tree distance metrics, although more metrics are available. It is not practical to investigate every distance metric in this list, so I chose to narrow my focus to a small sample of well-known metrics. The distances I chose to investigate were the Robinson-Foulds distance, the Euclidean distance and the geodesic distance, for reasons that will be discussed in section 2.4.

2.3 CLUSTERING

The distance metrics described previously are applied to every pair of trees in the input set, to produce a distance matrix, $\boldsymbol{D}$. This matrix is square and symmetrical, with off-diagonal entries $\boldsymbol{D}_{ij}$ encoding the distance ($\geqslant 0$)

between trees i and j, and diagonal entries $\boldsymbol{D_{ii}} = 0$, thus describing the distance between every pair of locus trees. A clustering algorithm can then use the information in the distance matrix to divide the input set into clusters. I have investigated seven such algorithms, detailed below. Each algorithm presumes that the required number of clusters is known in advance; new approaches for choosing the optimal number of clusters are introduced in section 2.5.

Methods that embed distances as coordinates in multidimensional spaces are described by the name of the embedding method (spectral clustering, multidimensional scaling, etc.), however the clustering itself is performed using the k-means algorithm.

2.3.1 HIERARCHICAL CLUSTERING

Agglomerative hierarchical clustering begins with every data point assigned to its own cluster. The pair of clusters that is closest together is merged into a single cluster (replacing the pair it is formed from), and the newly merged cluster's distance from all other clusters is computed according to a linkage formula. Pairs are successively merged in this fashion until all the data belong to the same cluster. This implicitly relates the elements in the dataset by a tree, which can be cut at appropriate points to obtain a partition with a specific number of clusters. There are different versions of hierarchical clustering that differ in their linkage formulae; for example, single-linkage defines the distance between two clusters to be the distance between their two closest points, complete-linkage defines it to be the distance between their two furthest points, average-linkage uses the mean of the distances between cluster members, and in Ward's method distances are updated to minimise the increase of the error sum of squares (Ward 1963).

2.3.2 COORDINATE TRANSFORMATION

Methods such as multidimensional scaling (MDS) can be used to embed a matrix of pairwise distances between points as points in Euclidean space.

Classical multidimensional scaling (CMDS), also known as principal coordinate analysis, is an analytical method for producing such an embedding. Metric multidimensional scaling instead minimises a quantity called *stress* that corresponds to the discrepancy between the distances among the embedded points and the original distances. Spectral clustering represents pairwise point similarities (as opposed to distances) as a graph. The graph is described by a Laplacian matrix; the eigenvalues of this matrix act as a set of coordinates embedding the points (Ng, Jordan, and Weiss 2002). Pairwise distances are converted to similarities using, for example, radial basis functions or the k-nearest neighbours procedure (Luxburg 2007), or by local scaling (Zelnik-Manor and Perona 2004). I implemented the local scaling variant as it does not require the choice of any scaling parameter values, but rather adjusts automatically to the data. As was noted above, the clustering step is performed using k-means, subsequently to embedding the points in Cartesian space.

2.3.3 k-means

The k-means algorithm partitions a set of data points into k clusters, with k set *a priori* (MacQueen 1967). The data points are taken to be distributed over the 'space' of the problem. It is an iterative procedure: (1) to initialise the algorithm, k cluster centres are chosen arbitrarily in the space; (2) data points are then associated with their nearest (by Euclidean distance) cluster centre, partitioning the set of data points into k clusters; (3) the positions of the cluster centres are updated so that they lie at the centroid of their associated data points. Steps (2) and (3) are repeated until the clusters stabilise. k-means can be sensitive to the initial positions chosen for the cluster centres. Typically, the algorithm is run multiple times from different starting points, and the partition that minimises the sum of the distances from the data points to their cluster centre is chosen as the best result. As k-means implicitly requires data points to lie in a Euclidean space, it is not a good fit for tree-distances, which are not Euclidean (Billera, Holmes, and

Vogtmann 2001). A similar procedure, *k*-medoids, can be used for arbitrary distances.

2.3.4 *k*-MEDOIDS

The *k*-medoids clustering procedure (Kaufman and Rousseeuw 1987) is related to *k*-means, however it differs in that the cluster centres are drawn from the observed data points: it uses k points from the dataset as exemplars to represent each of k clusters. Non-exemplar points are assigned to the group represented by their nearest exemplar. The roles of exemplar and non-exemplar are exchanged to minimise a cost function, which is the sum of the distances from each point to its representative exemplar. Assuming that each selection of exemplars uniquely defines a partition (i.e. that no point is exactly the same distance from more than one exemplar, which would make its group assignment ambiguous) there are $n!/(n-k)!k!$ possible partitions of n objects around k exemplars. Several algorithms exist to efficiently search among these for high scoring partitions (Park and Jun 2009).

2.4 A SIMULATION SURVEY OF METHODS

As previously mentioned, a straightforward way to cluster loci by their phylogenetic congruence is to (after tree inference) gauge the dissimilarities among the trees, and then proceed as with any other dissimilarity data. But there are numerous possible combinations of tree distance metric and clustering method, and little is known about which combination is likely to be the most effective at finding congruent groups in a phylogenetic inference context. To investigate this question I carried out a survey of combinations of a selection of metrics and methods. I chose three distance metrics and seven clustering methods to look at in detail.

Concerning distance metrics, I wanted to compare the results obtained using purely topological distances with those obtained from branch length sensitive metrics. The majority of the distance metrics available are topological, and to represent these I chose Robinson-Foulds as it is the most

well-established method. Of the metrics that incorporate branch length information I chose to use both the Euclidean distance and the geodesic distance. The Euclidean distance depends mainly on branch lengths, but it is unclear how much of an influence the topology has: the topology must have an influence on the Euclidean distance between trees due to its formulation from the splits present in each tree. Contrastingly, the geodesic distance explicitly incorporates both topology and branch lengths. Therefore I chose these two branch length sensitive metrics so as to see if the way topological information was incorporated made a difference to the clustering results. In summary, the tree distance metrics I included in the survey were the Robinson-Foulds, the Euclidean and the geodesic distances on trees.

For clustering I chose among methods that fit two requirements: the methods should be applicable to distance data and they should return a specific number of clusters. The first requirement is placed because the tree comparison produces a distance matrix. Many clustering methods are posed so that they work on inputs that are represented by real valued vectors, but there is no clear, unambiguous way to represent phylogenetic trees in this way. The second requirement is placed so that the results can be fed into the statistical stopping criteria described in section 2.5, which compare results obtained for specific numbers of clusters. This resulted in a selection of seven methods: four varieties of hierarchical clustering (single-linkage, complete-linkage, average-linkage and Ward's method; see section 2.3.1), plus k-medoids, spectral clustering with k-means, and classical multidimensional scaling with k-means.

I chose to test the combinations using simulated data, as it could be done using the common pattern of simulation studies: one simulates data for which the 'true' outcome is known, and then the performance of each tested method can be assessed by measuring how close it comes to reproducing the true answer. Simulation studies are useful as the investigator has complete control of the simulation parameters, and can investigate changes in them to an arbitrary degree of granularity. However, they are also limited as it is only practical to cover a subset of the combinations and ranges of

the parameters, which means that there will be certain types of data that will not be generated for testing. Also, simulation generates data that have perfect adherence to the known model, and so results are only derived for this idealised situation. Due to these limitations I chose to do a simulation study first, in order to test whether the method can work in the best case scenario, and later contrast the simulation results with studies on real data Chapter 3.

The datasets used in the survey were sequence alignments simulated according to incongruent evolutionary scenarios. The degree of incongruence was controlled to test how well the combinations performed compared to the difficulty of the problem case. I investigated the performance of combinations of distance metrics and clustering methods for a fixed and known number of clusters.

Variation of Information

To assess the accuracy of each resulting partition, I computed the difference between the true partition (known from simulation) and the inferred partition, using the variation of information (VI) measure, which is an information-theoretic measure of the difference between two partitions of the same set (Meilă 2007). In this measure a value of zero is obtained when the two partitions are the same, and increasing positive values are obtained for partitions that are increasingly different.

The variation of information between two alternative partitions of the same data, C and C', is computed as:

$$VI(C, C') = H(C) + H(C') - I(C, C') \tag{2.2}$$

where $H(C)$ is the entropy of C, and $I(C, C')$ is the mutual information of C and C':

$$H(C) = -\sum_{k=1}^{K} P(k)\, logP(k) \tag{2.3}$$

$$I(C,\ C') = \sum_{k=1}^{K}\sum_{k'=1}^{K'} P(k,k')\ log\frac{P(k,k')}{P(k)P(k')} \tag{2.4}$$

The probabilities required for the entropy and mutual information are calculated simply as:

$$P(k) = \frac{n_k}{n} \tag{2.5}$$

$$P(k,k') = \frac{|C_k \cap C'_{k'}|}{n} \tag{2.6}$$

An upper bound for VI on a dataset with n points is $VI(C,C') \leqslant \log n$.

2.4.1 SIMULATION

The purpose of the simulation procedure was to produce datasets that I could use to test the performance of various combinations of distance metrics and clustering methods. Important considerations were to be able to generate sets of data for which certain parameters could be controlled: things like how many clusters there should be underlying the data; how different from each other the loci in separate clusters should be (datasets consisting of similar loci are harder to separate than datasets in which they are very different); the number of loci belonging to each cluster; the number of taxa to simulate, and what proportion of missing data these taxa should have (as in real data it is common to have taxa for which sequence data for certain loci do not exist, or do exist but are missing from the sample).

The process for simulating datasets is given here in outline, with more detail in the sections that follow.

- First, generate a tree to describe the evolution of a set of taxa, common to the entire dataset—call this the 'common tree'.

- Then perturb the common tree to produce a 'cluster tree', that is, a tree that underlies loci that belong to a particular cluster.

- Repeat for as many clusters as are required for the dataset: each cluster is derived from a cluster tree that is produced by perturbing the common tree.
- Once the cluster trees are generated, then simulate locus alignments from the trees using Markovian models of sequence evolution. Simulation was done using my own software, the python package phylo_utils (`https://pypi.python.org/pypi/phylo_utils`).

Procedure for generating trees

The starting point for generating cluster trees is to simulate an ultrametric[†] common tree using a Yule process, a model of population growth in which each member of the population independently gives birth to offspring at a constant rate (Yule 1925). A prespecified number of cluster trees can then be derived from this. To produce one cluster tree, the common tree is perturbed by applying a series of either NNI or SPR tree rearrangements (see figures 1.3 and 1.4). The position at which a rearrangement is applied is selected randomly and uniformly by generating a random draw from a uniform distribution, $X \sim U(0, L)$, where L is the total tree length (sum of all branch lengths). This draw selects a position on the real line between 0 and L. A correspondence is made between this line and the tree's edges, allowing the selected position to be mapped back to a position on the tree. When applying an SPR rearrangement, this selected point acts as the 'donor' of an HGT (figure 2.4). The 'recipient' point is selected uniformly at random from among the other edges that exist in the tree at the same time depth (i.e. same distance from the root) as the donor. When applying an NNI the chosen position selects the edge around which the NNI is performed; the choice between the two possible NNI rearrangements is made uniformly at random. A sequence of one or more rearrangements is applied: the number of rearrangements applied controls the extent to which the cluster trees differ from each other. This whole process is repeated, beginning from the same underlying common tree, for each cluster tree.

[†]the root-to-tip distance is equal for all tips—see section 1.1.4

While the clustering method I have proposed is intended to be process-agnostic, the simulations were designed with plausible evolutionary scenarios in mind. I chose NNI or SPR rearrangements to mimic evolutionary processes, while enabling control over the degree of difference between cluster trees through the number of applications made to the common tree. Specifically, I used NNI and SPR as proxies for ILS and HGT, respectively (see sections 1.3.1 and 1.3.2). The NNI move represents a restricted version of ILS, in which the deep coalescence occurs in the immediate ancestor of the selected NNI branch (see figure 2.3). An alternative approach to simulate ILS more completely would be to use the multispecies coalescent to sample a gene tree from the common tree (Rannala and Yang 2003). However, this lacks the ability to control the degree of difference between clusters, except indirectly through control of population size parameters, which I felt would be too much of a blunt tool to be useful for these simulations.

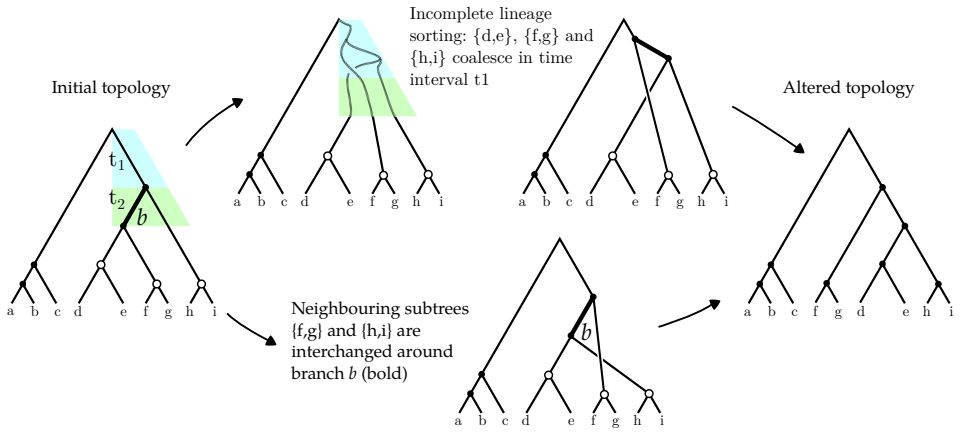


Figure 2.3. *Nearest neighbour interchange used as a proxy for incomplete lineage sorting (ILS). The figure shows two ways of arriving at the altered topology from the initial topology: the top route shows the topological change as deep coalescence in the ancestor of branch* b*, a case of ILS; the bottom route shows the same topological change as a nearest neighbour interchange operation among the three immediate descendants of the branch* b *(indicated by white circles).*

SPR moves were used to model HGT as summarised in figure 2.4. Here the tree represents the evolution of lineages through time, which proceeds from the root at the top of the figure, towards the leaves at the bottom. In

an HGT event genetic material is transferred from a donor lineage to a recipient lineage elsewhere in the tree. The HGT exchange causes a change in the shape of the tree, as the leaves that descend from the donor and recipient lineages now share a new most recent common ancestor at the point of the exchange. This change in tree shape is equivalent to an SPR rearrangement in which the recipient subtree is pruned from the tree and regrafted at the donor point (figure 2.4). As a side note, if the exchange takes place between neighbouring lineages in the tree then it may not lead to a change in tree topology, only in branch lengths. However, in my simulations only exchanges leading to a topological change were allowed.

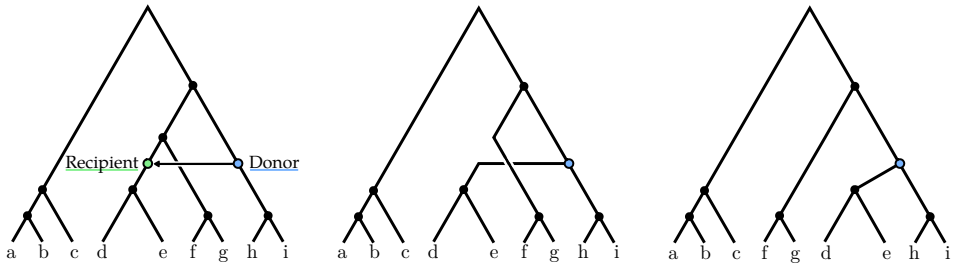


Figure 2.4. *Subtree prune and regraft used as a proxy for horizontal gene transfer (HGT). The figure shows the transfer of genetic material from a donor lineage to a contemporary recipient. The donation point becomes the most recent common ancestor of the descendants of both the donor and recipient lineages. This is equivalent to an SPR move that prunes the subtree at the receiving point and regrafts it at the donation point (equivalent to the move portrayed in figure 1.3).*

Branch lengths were then adjusted on the cluster trees by drawing values from separate Gamma distributions for internal ($\sim \Gamma(\kappa = 0.725, \theta = 0.12)$) and leaf edges ($\sim \Gamma(\kappa = 0.66, \theta = 0.265)$). These Gamma distribution parameters were estimated from the yeast trees inferred in Chapter 3.

Sequences were evolved along the cluster trees using Markovian models of sequence evolution, for example the WAG (Whelan and Goldman 2001) or LG (Le and Gascuel 2008) empirical protein substitution matrices. The simulation code was initially implemented using the Bio++ libraries (Guéguen et al. 2013), and later I implemented sequence simulation in my treeCl code. For simulations that were designed to have missing data, a proportion of the sequences in the sequence alignments were removed. For example, when

Name	Taxa	Clusters	Loci	Occupancy	Distribution of loci into clusters
Small uniform	20	4	60	100%	15, 15, 15, 15
Small skewed	20	4	60	100%	5, 10, 15, 30
Large uniform	40	6	90	100%	15, 15, 15, 15, 15, 15
Large skewed	40	6	90	100%	5, 5, 10, 10, 20, 40
Incomplete occupancy	50	4	60	40%–100%	15, 15, 15, 15

Table 2.1. *Attributes of the four simulated dataset scenarios with incongruence used to test combinations of distance metric and clustering method, and the scenario used to test the effect of incomplete occupancy.*

simulating an alignment belonging to a dataset with an occupancy of 60%, each taxon's sequence is retained in the alignment with probability 0.6.

I produced datasets according to scenarios with varying numbers of taxa, loci, clusters, occupancy and distribution of loci among these clusters. Simulation parameter settings for the scenarios investigated are summarised in table 2.1.

I will end this section with the acknowledgement that there are a huge number of choices that could have been made differently regarding these simulation parameters. In terms of the models used, for instance, the data could equally well have been simulated under nucleotide models as amino acid models. I chose to only use amino acid models, mainly for the reason that, because empirical amino acid models (such as WAG and LG) have been estimated from vast amounts of data, one might reasonably expect them to generate realistic sequences when used in simulation. Using nucleotide models would require alternative approaches: one could be to assemble a large corpus from which to estimate the model parameters, a large undertaking in itself; another could be to use arbitrary parameters (probably not too extreme) and assume the simulated sequences are reasonable. Other simulation parameters, such as dataset sizes, cluster distribution, number of taxa, occupancy levels and so on, could have been

explored with more granularity. However, the parameter space of this simulation study is already very large, and the amount of simulation (and computation) needed to explore the space in more detail would be prohibitive.

2.4.2 RESULTS

The results are summarised in figures 2.5 to 2.8. These figures each consist of six panels, in three columns and two rows. The three columns correspond to performance using the three distance metrics, Robinson-Foulds, Euclidean and geodesic. The two rows correspond to two simulation scenarios: 'uniform', in which all simulated clusters are the same size, and 'skewed', in which the clusters differ in size (for the cluster sizes, see table 2.1). Finally, each panel displays seven sets of points, one for each of the seven clustering methods investigated — these are coloured to distinguish the seven methods, and joined by lines of the same colour to display the trend.

Within each plot, each point conveys the corresponding method's ability to recover the true partition on data simulated with clusters separated by a particular number of tree rearrangements — each panel continues over a sequence of these numbers, indicated on the x-axis — by describing the recovered partition's distance from the true partition by means of the VI metric, indicated on the y-axis. The points indicate the mean VI value obtained from 1000 repetitions of the analysis, with error bars showing the standard error of the mean (this is often very small, so in many cases the error bars are hard to see).

On the y-axis, large VI values would indicate a partition that is very different from the true partition, i.e. a poorly performing clustering method. Conversely, small VI values — potentially reaching zero when the true partition is recovered exactly — would be obtained by clustering methods that perform well. The x-axis indicates the number of rearrangements that were applied to the common tree in generating the cluster trees (see section 2.4.1. The number of rearrangements correlates with how distant the clusters are

from each other. Small numbers of rearrangements produce clusters that are overall quite similar to each other, and are hence difficult to discriminate between. Conversely, large numbers of rearrangements can be expected to produce underlying trees that are quite different from each other, leading to more distinct clusters. Thus, one might expect the performance of clustering methods to improve from left to right on these plots. As can be seen in the four figures, the lines generally have a curved reverse 'J'-shape that is consistent with this expectation. The interpretation of the x- and y-axes according to the trends described in this paragraph are summarised by arrows in panel A of figure 2.5.

Panel A of figure 2.5, which shows the performance of the clustering methods applied to Robinson-Foulds distances calculated from datasets generated under the small, uniform partition scenario, with NNI rearrangements separating the clusters, illustrates how all seven methods perform better as the separation between clusters is increased. When cluster trees differ from the common tree by a single NNI (left-hand side of panel), all seven methods produce partitions that are comparably far away from truth, with a VI of approximately 2. As the number of rearrangements separating clusters (x-axis) is increased the methods produce partitions closer to the true partition. The different methods are spread apart as the x-value increases, and it is apparent that spectral clustering, in light blue, has the best performance, with lower VI than other methods. Contrastingly, single-linkage hierarchical clustering, in orange, has the worst performance of all the methods.

Panels B and C of figure 2.5 also show results from datasets generated under the small, uniform partition scenario, with NNI rearrangements, using Euclidean and geodesic distances, respectively. These show similar results: all methods tend to recover partitions closer to truth as cluster separation increases, single-linkage hierarchical clustering performs comparatively poorly, and spectral clustering performs well. Also, k-medoids and CMDS have performance comparable to spectral clustering — outperforming it slightly when clusters are close, separated by one or two NNI. The other simpler hierarchical clustering methods, complete-linkage and aver-

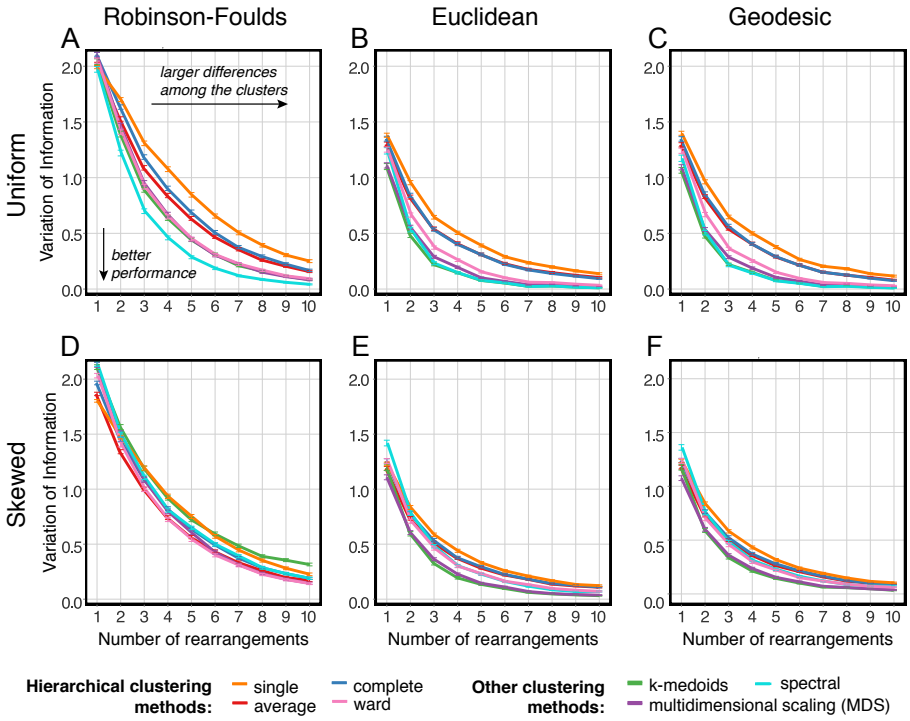


Figure 2.5. *Small dataset, NNI rearrangements. The relative performances of combinations of distance metric (varying over columns of panels) and clustering methods (shown by the colours of the lines), as measured by the variation of information metric (*y*-axes), which is a measure obtained when comparing the inferred solution with the true solution (higher values show a larger departure from the correct solution). Lines show the mean value obtained from 1000 replicates, and the error bars show the standard error of the mean (often this is very small, so the error bars are hard to see). Rows correspond to the experiments with a partition of uniformly-sized clusters (A–C) and those with a partition of clusters of skewed sizes (D–F). In each individual panel, the* x*-axis represents the number of NNI rearrangements separating the underlying clusters, so that increasing values along this axis correlate with the clustering problem becoming easier.*

age-linkage, perform only slightly better than single-linkage. The VI values are in general lower in panels B and C, suggesting that better partitions are obtained using Euclidean and geodesic distances than Robinson-Foulds, for this type of dataset. There is little difference in VI between the Euclidean and geodesic results.

Panels D–F show the results for datasets generated under the small, skewed partition scenario, again with NNI rearrangements. As with the panels A–C, lower values of VI are obtained with Euclidean and geodesic distances than with Robinson-Foulds, strongly suggesting that using these branch

length sensitive distances allows clustering methods to find better partitions on skewed cluster distributions also. However, the relative performances of clustering methods are different: under the uniform cluster distributions there was a separation between the poor performance of simpler hierarchical methods and the better performance of the other methods, but this is less pronounced under the skewed cluster distribution. Spectral clustering is no longer the best clustering method, falling somewhere in the middle for all distance metrics. The best performing clustering methods are Ward's method (pink) when using Robinson-Foulds distances (panel D), and *k*-medoids (green) or CMDS (purple) using Euclidean or geodesic (panels E and F).

Panels A–C of figure 2.6 shows the performance of the clustering methods applied to Robinson-Foulds, Euclidean and geodesic distances calculated from datasets generated under the small, uniform partition scenario, this time with SPR rearrangements separating the clusters. These show similar trends to their counterpart panels in figure 2.5: single-linkage hierarchical clustering shows the worst performance, and spectral, CMDS and *k*-medoids show the best. Once again, increasing cluster separation allows the clustering methods to produce partitions closer to the truth. This occurs more rapidly with SPR rearrangements: with five or more rearrangements all methods produce partitions that are on average very close to zero VI units away from the truth, which means they are producing the correct answer in almost all of the simulation replicates. This is likely due to SPR being a larger move in tree-space: the set of all SPR rearrangements also includes all NNI rearrangements.

The results in panels D–F of figure 2.6 are similar to those in panels D–F of figure 2.5 with the striking difference that *k*-medoids produces partitions that are all 0.4–0.5 VI units away from truth regardless of cluster separation, as shown by the almost horizontal green line. The erratic performance of *k*-medoids can also be seen in panels E and F, to a lesser degree. Problematic behaviour can also be seen in the CMDS results for 8–10 rearrangements in panel D.

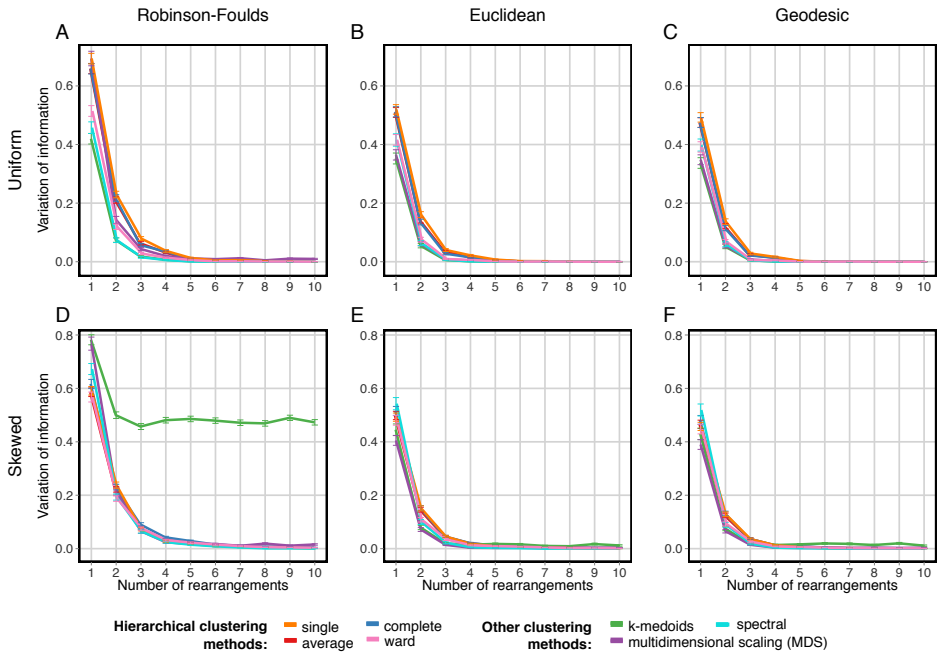


Figure 2.6. *Small dataset, SPR rearrangements. Panels show the relative performances of combinations of distance metric (varying over columns of panels) and clustering methods (shown by the colours of the lines), as measured by the variation of information metric (y-axes), which is a measure obtained when comparing the inferred solution with the true solution (higher values show a larger departure from the correct solution). Lines show the mean value obtained from 1000 replicates, and the error bars show the standard error of the mean. Rows correspond to the experiments with a partition of uniformly-sized clusters (A–C) and those with a partition of clusters of skewed sizes (D–F). In each individual panel, the x-axis represents the number of SPR rearrangements separating the underlying clusters, so that increasing values along this axis correlate with the clustering problem becoming easier.*

Figures 2.7 and 2.8 are analogous to figures 2.5 and 2.6, respectively, but for the large uniform and large skewed partition scenarios of table 2.1. Figure 2.7 shows the results for the large uniform and large skewed partition scenarios with clusters separated by NNI rearrangements. The results for the uniform cluster distributions, in panels A–C, show similar patterns to the results from the small uniform scenario, in that the cluster performance improves as clusters become more separated, partitions derived from Euclidean and geodesic distances are closer to truth than those derived from Robinson-Foulds distances, and the best performing clustering method in the majority of cases is spectral clustering. However, when clusters are very similar (separated by 1–2 NNI rearrangements) single-linkage hierarchical

clustering provides the best results. At greater separation hierarchical clustering methods give worse performance than other methods, as was the case for the small dataset scenario. For the skewed cluster distributions (panels D–F), once again better results are obtained when using Euclidean and geodesic distances, CMDS is generally the best performing method, and single-linkage hierarchical clustering performs well when clusters are similar. In panels E and F, *k*-medoids shows a similar, though less pronounced, erratic tendency to that shown in figure 2.6, panel D.

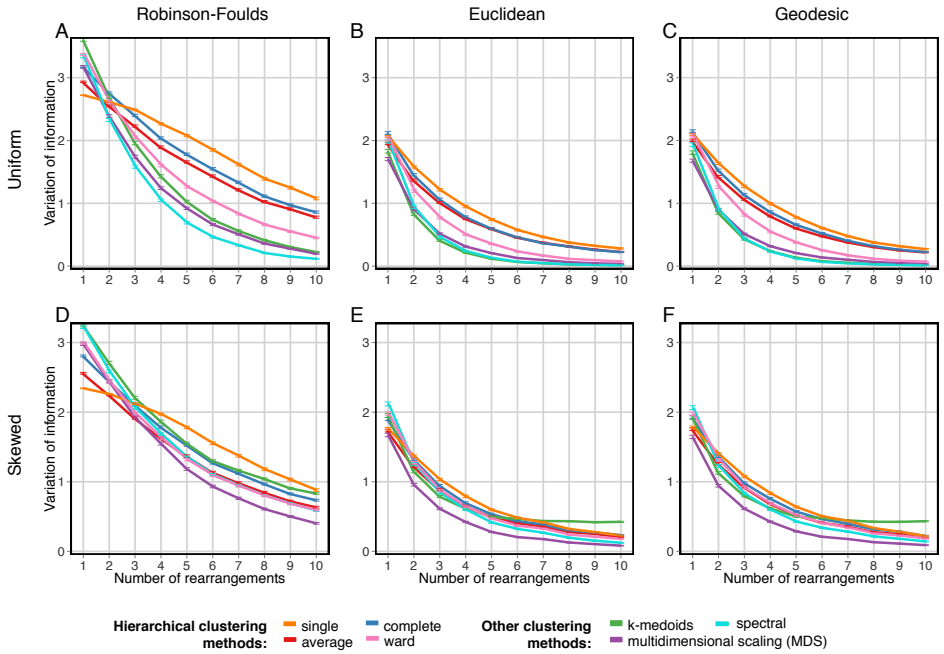


Figure 2.7. *Large dataset, NNI rearrangements. Panels show the relative performances of combinations of distance metric (varying over columns of panels) and clustering methods (shown by the colours of the lines), as measured by the variation of information metric (*y*-axes), which is a measure obtained when comparing the inferred solution with the true solution (higher values show a larger departure from the correct solution). In each individual panel, the* x*-axis represents the number of NNI rearrangements between underlying clusters.*

Figure 2.8 shows the results for the large uniform and large skewed simulation scenarios, now with clusters separated by SPR rearrangements. The difference in clustering performance using different distance metrics is less pronounced than in the three earlier figures, but shows the same trend: partitions derived from branch length-sensitive distances (Euclidean and geodesic) are in general closer to truth than those inferred from Robinson-

Foulds distances. The best performing clustering methods are spectral clustering and *k*-medoids for the uniform cluster sizes (panels A–C), and CMDS followed by spectral clustering for the skewed cluster sizes (panels D–F). In panels D–F, *k*-medoids again shows erratic behaviour.

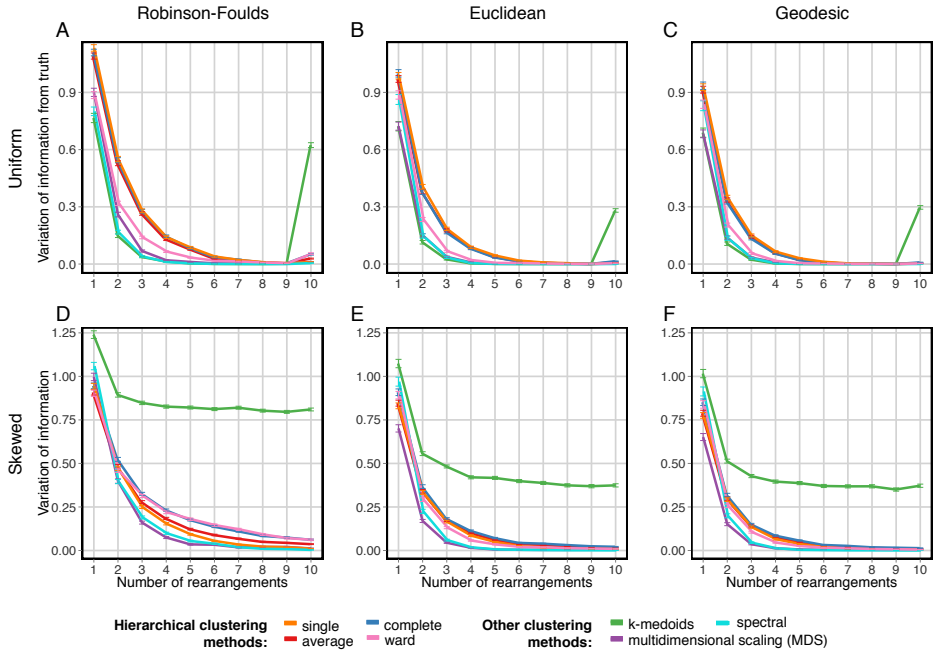


Figure 2.8. *Large dataset, SPR rearrangements. Panels show the relative performances of combinations of distance metric (varying over columns of panels) and clustering methods (shown by the colours of the lines), as measured by the variation of information metric (*y*-axes), which is a measure obtained when comparing the inferred solution with the true solution (higher values show a larger departure from the correct solution). In each individual panel, the* x*-axis represents the number of SPR rearrangements separating the underlying clusters.*

2.4.3 CONCLUSIONS

In terms of distance metrics, it is clear that including branch length information has a large influence of the clustering performance. Considerably better performance is obtained using the Euclidean and geodesic distances than using Robinson-Foulds. This is a strong argument for using branch length sensitive measures of phylogenetic tree distance when clustering based on phylogenetic congruence. Of these two branch length sensitive metrics, marginally better results are obtained from geodesic distances than Euclidean. This observation holds across all scenarios tested: for both uni-

form and skewed cluster distributions (compare panels A–C to D–F in each figure), for the small (figures 2.5 and 2.6) and large datasets (figures 2.7 and 2.8), and both for scenarios simulating ILS and horizontal gene transfer (compare figures 2.5 and 2.7 and figures 2.6 and 2.8).

In terms of clustering methods, it is less clear that a single best method exists. In many cases the performance is worse when using the hierarchical methods—single-linkage, complete-linkage, average-linkage and Ward's method, but this is not always the case. When the cluster size distributions are skewed the hierarchical methods do only slightly worse than other methods. The results from the large datasets show that when clusters are very similar to each other single-linkage hierarchical clustering can be the best performing method, even though in the majority of other scenarios it is the worst. Hierarchical clustering using Ward's criterion is generally more successful than other hierarchical methods. The other methods — spectral clustering, CMDS, *k*-medoids — generally outperform hierarchical methods, but of these there is no clear winner; each method is the best performing method in some of the panels. However, *k*-medoids shows erratic behaviour in some of the scenarios tested (figures 2.6 and 2.8), as does CMDS, to a lesser degree. As *k*-medoids and CMDS show occasional unreliability, these were not used in further analyses.

Summarising these observations, the combination of Euclidean or geodesic distances with spectral or Ward's clustering seem to provide consistently the best overall performance across various conditions tested here. These combinations were used in my further analyses.

2.5 INFERRING THE NUMBER OF CLUSTERS

The simulation survey described in section 2.4 was intended to find distance metrics and clustering methods that work well in combination to partition data for which the number of clusters is known in advance. However, it is rare to have this knowledge. For any clustering problem, the question of how to best choose the number of clusters is difficult, and finding pro-

cedures to tackle this question remains an open topic of research (Sugar and James 2011; Fujita, Takahashi, and Patriota 2014). Many of the existing approaches for estimating the number of clusters work along the following lines: they derive a statistic that varies with the clustering of the data, and then choose the number of clusters for which the statistic is maximised or minimised (Akaike 1973; Schwarz 1978; Rousseeuw 1987; Kelley, Gardner, and Sutcliffe 1996), or where a large gap occurs between adjacent numbers of clusters (Tibshirani, Walther, and Hastie 2001; Fujita, Takahashi, and Patriota 2014). In this section I introduce two special purpose methods I have developed for choosing the number of clusters in phylogenetic datasets that are based on likelihood ratio testing. The likelihoods used in the likelihood ratio test are quantities I call *partition likelihoods*. These are described in the next section.

2.5.1 PARTITION LIKELIHOOD

In order to assess partitions, which may be obtained from different clustering approaches, and compare them, I need to devise some statistic that I can use to gauge their 'goodness of fit'. As the tree inferences are done under maximum likelihood, a natural measure to use is the likelihood of the data over the whole partition. I used this as a quality score to assess the improvement to the fit of the model as the number of clusters was increased.

The three-step pipeline (section 2.1) ends with the data having been divided into a partition, whereby each cluster comprises a subset of the loci, and is a collection of loci that putatively share a common evolutionary history. Now I introduce a fourth step, in which, hoping to benefit from a more robust evolutionary inference by combining the data from homogeneous sources, I concatenate the alignments of the member loci and infer the maximum likelihood (ML) tree—denoted the (inferred) cluster tree. As the loci putatively share an underlying phylogeny, this could lead to more robust estimates due to using more data, as in supermatrix approaches (Queiroz and Gatesy 2007). Ambiguous or unknown characters introduced by concatenation, which will occur if any sequences are present in only a subset of the

alignments, are encoded as missing data (represented by an 'X', 'N' or '?'). I use a partitioned model for the analysis of concatenated sequences. This means that each locus within the concatenated alignment shares the same tree topology and set of branch lengths, but the substitution model parameters are estimated on a per-locus basis. It should be noted that this use of 'partition' refers to assigning model parameters *a priori* to groups of sites within an alignment, and is not the same as the use of 'partition' to refer to a clustering result, although the ideas are related. For the simulation studies presented here I have used the WAG model, in which all parameters are fixed to predetermined values, so using a partitioned model has no effect: each locus receives the same set of parameters. However, when using other models for which parameters are estimated from the data, the use of partitioning is likely to increase the accuracy of the inferred tree (Brown and Lemmon 2007; Darriba and Posada 2015). The maximum likelihood tree is inferred for each concatenated cluster tree. The partition log-likelihood, ℓ_k^P, is the sum of the k optimal cluster log-likelihoods for the entire partition. This is in effect the maximum log-likelihood under a model where the genes within each cluster share a common evolutionary history and evolutionary dynamics, but there are no constraints that different clusters share any evolutionary parameters. Partitions obtained for different numbers of clusters can be compared using the partition likelihood, and statistical significance gauged through the use of likelihood ratio tests. As partition likelihoods as defined here are expressed as log-likelihoods, the ratio is actually calculated as the difference between partition (log-)likelihoods.

2.5.2 A STOPPING CRITERION BASED ON LIKELIHOOD RATIO TESTS

The process I have devised for choosing the number of clusters is to partition the data into sequentially increasing numbers of clusters, and then for each number do a likelihood ratio test to see if adding another cluster is statistically justified. If the likelihood ratio test fails, adding another cluster is not justified, so the procedure is stopped. The number of clusters reached

before the test fails is the number decided upon, which will be somewhere in the interval $[1, m]$, where m is the number of loci in the dataset.

To fully explain the process, here are some further details about each likelihood ratio test. Let us consider the case of choosing between k and $k+1$ clusters. This is equivalent to choosing between the hypotheses that either the loci are sampled from k evolutionary trees, or from $k+1$ evolutionary trees, and these form our null and alternative hypotheses, respectively. We can thus cluster our data into partitions of k and $k+1$ clusters, calculate the partition log-likelihood of each result, ℓ_k^P and ℓ_{k+1}^P, and the increase in log-likelihood, $\Delta_k = \ell_{k+1}^P - \ell_k^P$ (i.e. the log of the ratio of the partition likelihoods), and use this result as a test statistic.

Being able to compute Δ_k provides a natural basis for a statistical test for deciding the number of clusters, similar to the tests introduced to phylogenetics in Goldman (1993) and Yang, Goldman, and Friday (1994, 1995). Because specifying a greater number of clusters provides more freedom for the model to fit the data, the likelihood is expected to increase (i.e. Δ_k will be positive). However, as in all likelihood ratio tests, the key consideration is by how much the likelihood must increase to warrant using the more complex model. In order to distinguish whether or not the increase is 'enough' to justify the complex model, we must know something about the expected distribution of Δ_k under the null hypothesis.

2.5.3 DISTRIBUTION OF THE TEST STATISTIC

In classical statistics, for nested hypotheses, $2\Delta_k$ is asymptotically chi-squared-distributed, with the number of degrees of freedom corresponding to the difference in the number of parameters between the null and alternative hypotheses (Wilks 1938). In this phylogenetic clustering context the alternative hypothesis has an extra cluster compared to the null hypothesis. The parameters associated with the extra cluster are a set of phylogenetic model parameters, a tree topology, and branch lengths on the tree. However, counting parameters proves difficult in this case: the extra parameters in the alternative hypothesis include an inferred tree topology; tree topo-

logies are discrete entities and thus are not the same as parameters in a typical sense (Yang, Goldman, and Friday 1994). As a consequence of not being able to quantify the number of excess parameters in the alternative hypothesis I cannot specify which chi-squared distribution I should use for the test. The inability to count parameters also precludes the use of information criteria — such as the AIC or BIC (Akaike 1973; Schwarz 1978) — as alternative methods of estimating the number of clusters.

As an alternative to the classical statistical route, I can sample from the distribution to obtain it empirically, through the use of pseudoreplicated data. I can estimate the distribution of Δ_k by repeatedly calculating Δ_k values from new datasets generated under the null hypothesis. This is the essence of the parametric bootstrap introduced to phylogenetics by Goldman (1993). Such a procedure, unlike classical statistics, does not require that the difference in degrees of freedom be known or that the hypotheses be nested (Cox 1961, 1962; Goldman 1993). I devised two procedures to resample the data: a permutation test, in which the new datasets are produced by randomising the original data, and a parametric bootstrap test, in which new datasets are generated via simulation. These permit me to compare whether $k + 1$ clusters are statistically supported over k clusters, and I apply such tests successively for $k = 1, 2, 3, \ldots$ and use the stopping criterion that k^* clusters are taken to be optimal where k^* is the smallest value of k for which $k + 1$ clusters are not statistically supported over k clusters.

2.5.4 PSEUDOREPLICATE RESAMPLING METHODS

To test the stopping criterion I developed two special-purpose procedures that generate resampled pseudoreplicate datasets, allowing empirical distributions of the likelihood ratio test statistic to be obtained. One procedure involves permutation of the original data to generate new pseudoreplicate data, which is a non-parametric technique (Efron 1985); the other simulates new data using parameters inferred when analysing the original data, in the manner of a parametric bootstrap (Goldman 1993). These two procedures are described in sections 2.5.5 and 2.5.6, respectively. Generating and ana-

lysing multiple replicates creates a computational burden, but the nature of the problem necessitates such an approach, as certain requirements of classical likelihood ratio testing are not met (see section 2.5.3).

I implemented the stopping criterion for both these methods of generating empirical distributions of Δ_k. I compared their ability to select the correct number of clusters to a well-known heuristic method, the silhouette (Rousseeuw 1987, see also section 2.5.7). It should be noted that, whatever stopping criterion is applied, the cluster memberships are determined by the original data, and are the same.

2.5.5 PARAMETRIC BOOTSTRAP

In the parametric bootstrap I use simulation to generate new datasets using parameters estimated during the analysis of the original data. After performing the analysis on the original data, each locus has been assigned to one of k clusters, and is therefore associated with one of k cluster trees. In the simulated pseudoreplicate dataset, each locus is simulated along its associated cluster tree using the evolutionary model and its estimated parameters from the analysis, with the alignment length and any gap positions duplicated from the initial data (Goldman, Thorne, and Jones 1998). Consequently, the data are simulated under the null hypothesis that loci evolved along k underlying trees. The simulated pseudoreplicate datasets are independently analysed in the same way the original data were, and are clustered into k and $k+1$ clusters to calculate the consequent increase in the partition log-likelihood, Δ_k. This "parametric bootstrap" procedure is repeated for 100 datasets to estimate the null hypothesis distribution of Δ_k. The value of Δ_k obtained in the analysis of the original data is compared to this distribution: 'small' values indicate no significant improvement, suggesting that k clusters provide the better model; significantly large values suggest that $k+1$ clusters is better.

2.5.6 NON-PARAMETRIC PERMUTATION

As a non-parametric alternative to the parametric bootstrap, the permutation test generates a new dataset from the input dataset by permuting the columns of all the multiple sequence alignments in the data set. The alignment columns that comprise the data set are shuffled, and redistributed over the individual alignments (see figure 2.9). The effect of this is to uniformly distribute the columns from each alignment over the dataset, removing any between-locus incongruence that might form the basis for clustering. Each alignment contains a mixture of histories. These resampled data are analysed twice: trees are inferred and the data are partitioned into k and $k+1$ clusters, from which Δ_k is calculated. The whole permutation procedure is repeated some large number of times (in my experiments this was commonly 100, more unusually 1000) to estimate the distribution of Δ_k.

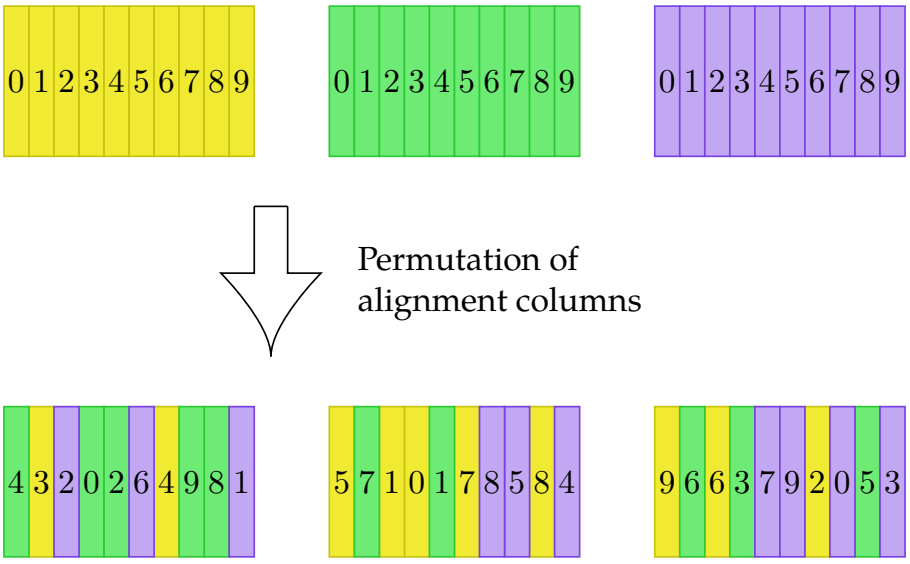


Figure 2.9. *Generating a permuted data set. A data set of three alignments (upper row) is permuted to generate a pseudoreplicate data set (lower row). Alignment columns are shuffled into a random order, and redistributed over the three alignments. The alignment columns are represented by numbered, coloured rectangles. Numbers are included to illustrate where each column ends up.*

Note that this procedure essentially assumes that all loci are sampled from the same, mixed, evolutionary history, and, in expectation, there is no 'clusterable' variation among the loci. Δ_k is a comparison of the null hy-

pothesis of identifying k clusters *in the absence of any real clusters* versus the alternative of identifying $k+1$ clusters *in the absence of any real clusters*. This differes from the parametric bootstrap, in which Δ_k is a comparison of finding k clusters when there are k clusters, versus finding $k+1$ clusters when there are k clusters.

Computational burden

While this method of resampling the data is incorrect, it has the benefit that the resampled datasets do not depend on k, and can be reused for various values of k. This is unlike the parametric bootstrap, which requires that pseudoreplicate datasets be simulated independently for each value of k. Because the pseudoreplicates are simulated according to parameters estimated from the original data, they must be generated independently for each value of k being investigated, as the parameters inferred for each value of k are independent and almost certainly will differ. This means simulated datasets cannot be reused for different values of k, so parametric bootstrapping is somewhat more computationally costly than the permutation test. The cost is incurred because in addition to generating more pseudoreplicate datasets (low cost), there is also a greater amount of analysis (inferring trees, calculating distances, etc.—high cost) required to implement the stopping criterion using parametric bootstrapping. It is these analyses that impose a high computational burden.

For the non-parametric test it would be conceptually preferable to permute the columns such that a distribution of loci among exactly k underlying trees is preserved, so that Δ_k is founded on comparing indentifying k and $k+1$ clusters when there actually are k clusters. However, this would reintroduce a dependency on k and undo the computational saving permutation offers over the parametric bootstrap. Therefore, for the pragmatic reason of wanting to reduce computational burden, I used the improper resampling procedure detailed above. My simulation results suggest that the simpler permutation procedure described above works well in practice over the range of scenarios tested (see section 2.5.8 for further discussion).

2.5.7 SILHOUETTE

I also use the silhouette criterion (Rousseeuw 1987) for comparison with the novel likelihood-based criteria I have introduced, as it is a general purpose, widely used heuristic for estimating the number of clusters in a dataset. A brief description of the silhouette is as follows. For a single point, i, one calculates the mean of its distances to all the other points in its own cluster (c_0), and also to each of the other clusters ($c_{K,i\notin K}$). The cluster with the minimum c_K is designated the "neighbouring cluster" ($c_{K_{min}}$) for point i. The silhouette value for point i is the ratio $S_i = \frac{c_{K_{min}} - c_0}{\max(c_0, c_{K_{min}})}$. The silhouette value of the whole dataset is the mean of the pointwise silhouettes. As higher values of silhouette imply that between-cluster distances exceed within-cluster distances, the clustering that maximises the silhouette is preferred.

2.5.8 SIMULATION DESIGN AND RESULTS

I simulated datasets using the procedure corresponding to the "small uniform" setup (see table 2.1). I wanted to investigate the performance of the stopping criterion on well separated clusters as well as close clusters, so I generated data with two levels of separation: 100 datasets from trees separated by 1 SPR rearrangement, and 100 separated by 5 SPR rearrangements. As was illustrated in figure 2.6, when clustering methods are given the correct number of clusters to find, with a separation of 1 SPR the clustering methods all fail to find the true partition, whereas with a separation of 5 SPR rearrangements all clustering methods typically do find the correct partition. I term the simulations made with 5 SPR rearrangements separating clusters as the "moderate" cases, as, while for these cases the methods will likely find very good partitions of the data, the main challenge is to find the correct number of clusters. The simulations made with 1 SPR rearrangement separating clusters I term the "difficult" cases, as these the methods are less likely to find very good partitions of the data, and still must estimate the number of clusters, which is likely to be a more challenging problem.

Each dataset was analysed using the four combinations of Euclidean and geodesic distances with spectral and Ward's method clustering to generate

partitions of the data into 1–10 clusters (a range that includes the true value of 4). For each dataset, 100 sets of pseudoreplicate data were generated using the parametric bootstrap procedure and 100 sets by permutation, which were used to implement my stopping criterion. Additionally, the silhouette value was calculated using code I implemented in treeCl.

For clarity, I first describe the results for a single simulated dataset, before describing the aggregate results over many simulations. I will use the term "problem instance" to describe a single dataset analysed using one of the four combinations of distance metric and clustering method. For a single problem instance, I calculate inter-tree distances using the chosen metric, use the chosen clustering method to cluster the loci into 1–10 clusters, and compute the partition likelihood for each. I use the permutation and parametric bootstrap procedures (sections 2.5.5 and 2.5.6) to generate empirical distributions of the likelihood increase, Δ_k, expected for each number of clusters. The Δ_k from the original data is compared to the empirical distribution to determine whether the greater number of clusters is statistically supported, which is the case if Δ_k for the original data is greater than the 95th percentile of the distribution of the pseudoreplicate Δ_k. Additionally, the silhouette score is calculated for each of the 1–10 clusters. The silhouette criterion chooses the number of clusters for which the silhouette score is maximised. Figure 2.10 illustrates this procedure for one problem instance in which all three methods correctly infer that there should be four clusters.

Aggregated results over many problem instances are presented as histograms in four two-panel figures (figures 2.11 to 2.14). In each figure the results using geodesic distances are shown in the panel on the left, and those using Euclidean distances are shown on the right. Figures 2.11 and 2.12 show the results for the moderate (5 SPR) cases, and figures 2.13 and 2.14 the results for the difficult (1 SPR) cases. Each panel is a summary of the results from 100 problem instances.

Clearly, the methods all perform more reliably in the moderate cases than the difficult cases, as was expected. The left panel of figure 2.11 shows that

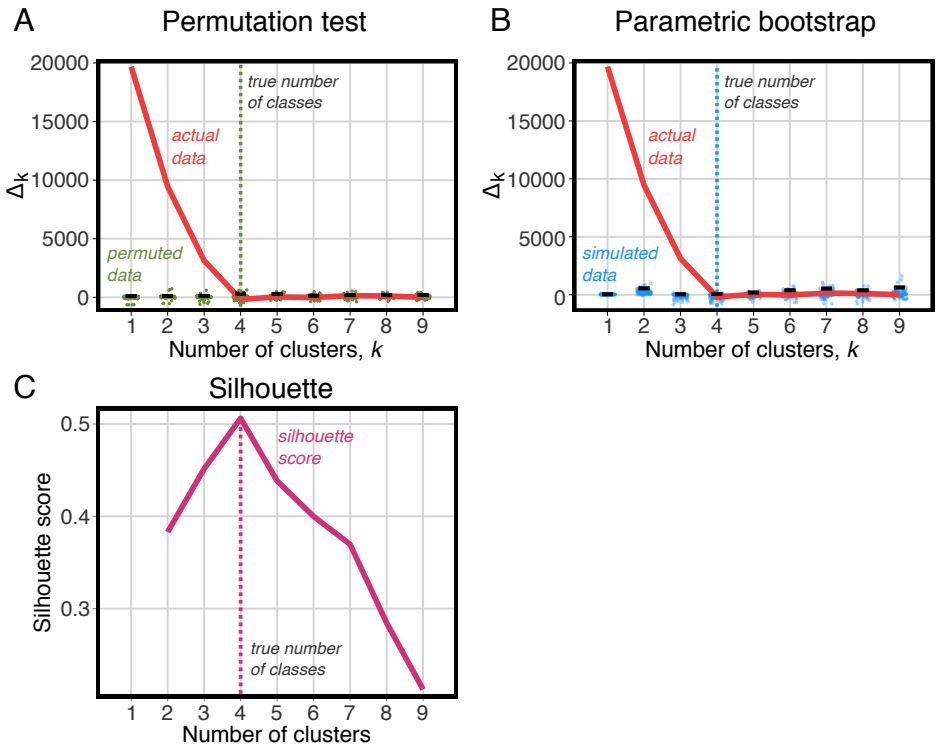


Figure 2.10. *Comparison of the criteria used to determine the number of clusters on a single problem instance — in this example, data simulated for 60 loci belonging to 4 clusters, each of size 15, with the clusters' trees separated by 1 SPR rearrangement. As the proposed number of clusters increases, the likelihood increases, which is expected because of the greater number of free parameters in the model. (A) Permutation test: the improvement in likelihood for each additional cluster (red curve) is significantly greater than that observed for permuted datasets (green dots show the distribution of values over 100 permutations) until the comparison between 4 and 5 clusters is reached, correctly implying that the use of 4 clusters is optimal. (B) Parametric bootstrap test: again, the improvement for each additional cluster (red curve) is significantly greater than that for datasets simulated for one fewer cluster (blue dots) until the true number of clusters (4) has been reached. (C) Silhouette score: the general-purpose silhouette stopping criterion has its maximum at the true value of 4. Note that in this instance, comprising a single dataset from one simulation design, the three methods agree on the true answer.*

the permutation and parametric bootstrap criteria both obtain the correct answer of 4 clusters 100% of the time, when spectral clustering and geodesic distances are used. For both the "difficult" and "moderate" cases the distribution of the number of clusters chosen is centred on the true value, 4, for all three criteria (figures 2.11 to 2.14). In the "moderate" cases all the criteria perform well (figures 2.11 and 2.12): they make the correct call a clear majority of the time, although it is apparent that the silhouette cri-

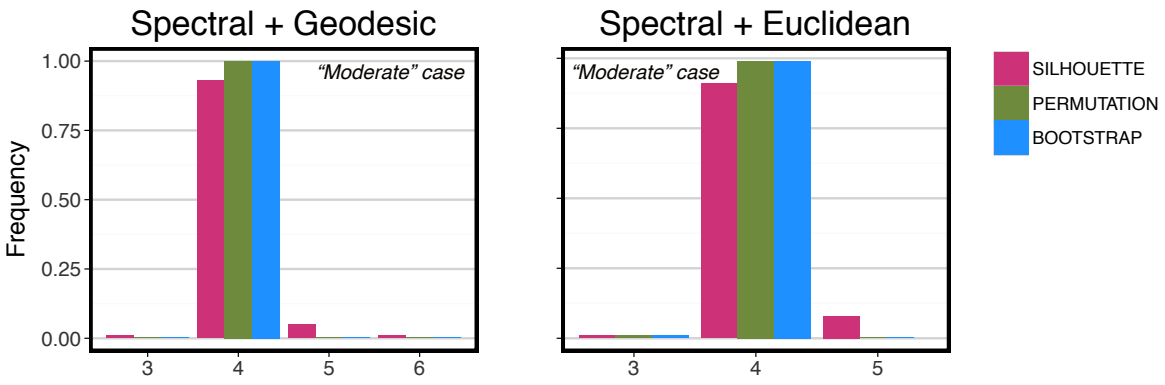


Figure 2.11. *Distributions of the number of clusters found for 100 "moderate" problem instances, using spectral clustering. In every instance the true number of clusters is 4. The height of the bars corresponds to the frequency that the number of clusters indicated by the bar's position on the* x*-axis was chosen by each criterion—silhouette (pink), permutation-based stopping criterion (green), parametric bootstrap-based stopping criterion (blue).*

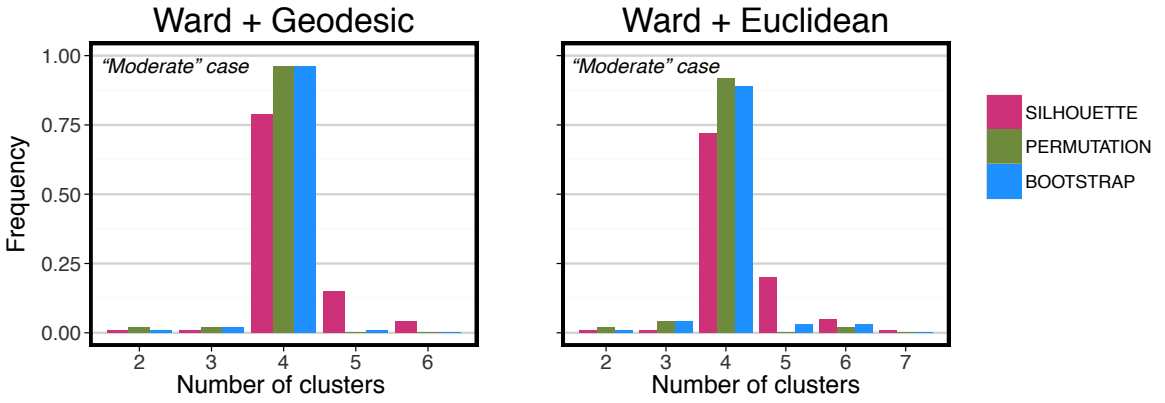


Figure 2.12. *Distributions of the number of clusters found for 100 "moderate" problem instances, using Ward's method clustering. In every instance the true number of clusters is 4. The height of the bars corresponds to the frequency that the number of clusters indicated by the bar's position on the* x*-axis was chosen by each criterion—silhouette (pink), permutation-based stopping criterion (green), parametric bootstrap-based stopping criterion (blue).*

terion performs less well than the permutation and parametric bootstrap variants. In the "difficult" cases (figures 2.13 and 2.14), the distributions of results of the permutation and bootstrap variants of the stopping criterion are much tighter than the results of the silhouette criterion, indicating that these two special purpose methods make correct calls more often. When the criteria identify an incorrect number of clusters, they tend towards un-

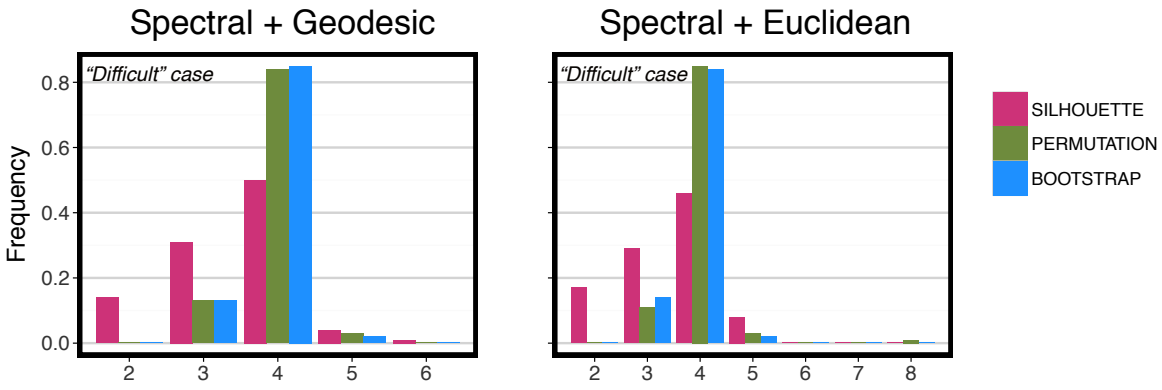


Figure 2.13. *Distributions of the number of clusters found for 100 "difficult" problem instances, using spectral clustering. In every instance the true number of clusters is 4. The height of the bars corresponds to the frequency that the number of clusters indicated by the bar's position on the x-axis was chosen by each criterion—silhouette (pink), permutation-based stopping criterion (green), parametric bootstrap-based stopping criterion (blue).*

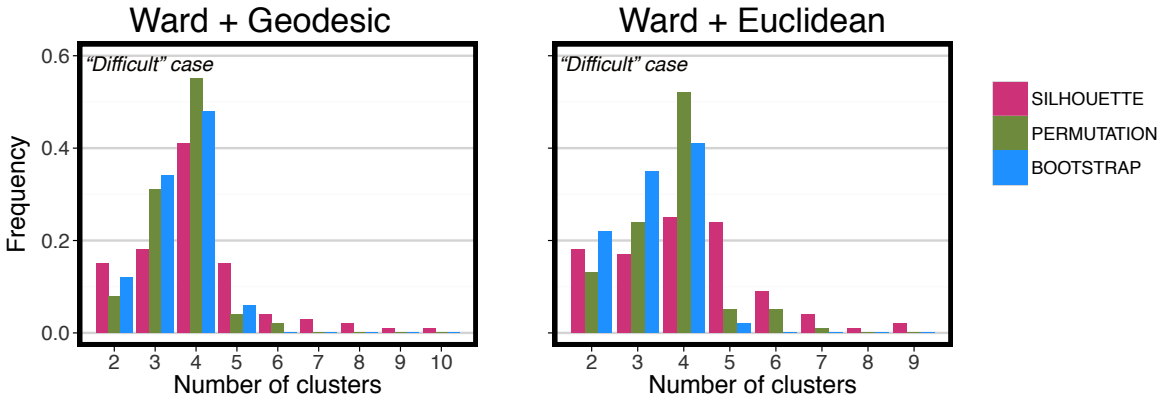


Figure 2.14. *Distributions of the number of clusters found for 100 "difficult" problem instances, using Ward's method clustering. In every instance the true number of clusters is 4. The height of the bars corresponds to the frequency that the number of clusters indicated by the bar's position on the x-axis was chosen by each criterion—silhouette (pink), permutation-based stopping criterion (green), parametric bootstrap-based stopping criterion (blue).*

derestimating the number more frequently, as can be seen by the somewhat right-skewed shape of the histograms in figure 2.14. This suggests that all the methods are slightly conservative.

In terms of distance metrics, the results using geodesic and Euclidean distances are broadly similar, but using Euclidean distances causes more incorrect calls of the number of clusters. This can be seen most clearly in

the results using Ward's method of clustering figure 2.14: comparing the left (geodesic) and right (Euclidean) panels shows that the bars at the position of 4 clusters (which is the correct value) are shorter for inferences made with Euclidean distances, which means fewer correct calls of 4 clusters were made. This affects the silhouette criterion more than my special purpose criterion in either of its variants.

Comparing clustering methods, one can observe that all three stopping criteria perform noticeably better in combination with spectral clustering than with Ward's method. Comparison of the results for the moderate cases (figures 2.11 and 2.12) shows this, as spectral clustering obtains the correct answer in all or very nearly all cases for the permutation and parametric bootstrap criteria, while Ward's method, while successful, does not achieve the same level of performance. A larger drop-off in performance can be seen comparing the performance of the silhouette criterion between these two figures: for spectral clustering its success rate is approximately 90%; for Ward's method it is around 75%. However, the difference between spectral clustering and Ward's method can be seen most clearly by comparing the difficult cases (figures 2.13 and 2.14): using spectral clustering allows the correct number of clusters to be found more than 80% of the time, whereas with Ward's method it is less than 60%, whether geodesic or Euclidean distances are used. Furthermore, the permutation and bootstrap variants of the new stopping criterion outperform the silhouette criterion by a greater margin on the spectral clustering runs.

In summary, the special purpose stopping criterion I devised for determining the number of clusters works very successfully on these simulated data, outperforming the general-purpose silhouette criterion. The results also provide strong evidence for preferring spectral clustering over Ward's method, and also for preferring geodesic distances over Euclidean distances, although this distinction is less strong.

2.6 EFFECT OF INCOMPLETE LOCUS OCCUPANCY

When analysing real data, one cannot guarantee that all loci will be available for all taxa. Any given locus may genuinely not exist in the genomes of some of the taxa, or it does exist but it has not yet been sequenced. Using datasets with missing sequences means that locus trees will be inferred that have different sets of leaves. This creates the requirement that distances be calculated for trees with different leaf sets, a circumstance for which distance metrics have not been defined. A simple measure to work around the lack of metrics for trees with different leaf sets is to prune trees to the intersection of their leaf sets, and then use existing metrics to measure the distance between these reduced trees. This what I have done in treeCl. My justification for doing this is that I want the clustering approach to separate loci that have conflicting evolutionary histories, only assigning loci to different groups if conflict exists in the observed portions of the tree. However, I acknowledge that this approach is not ideal, as pruning trees may violate the metric nature of the distances. On a pragmatic note, it is also unsatisfying to prune trees as it slows down the distance calculation substantially, although this effect could be lessened by improving the implementation of the pruning algorithm I use in treeCl.

To assess the impact of incomplete occupancy on my approach's ability to infer the correct clusters, I generated additional simulated datasets containing a varying proportion of randomly selected missing loci (see section 2.4.1 for details of how missing data was produced). I simulated data with larger trees (50 taxa) than in the previous studies, to reduce the frequency of the occurrence of having to compare trees with very small leaf set overlaps. All analysis on datasets with missing data was done using geodesic distances and spectral clustering.

To assess the impact of missing data on clustering accuracy I repeated the approach of section 2.4. Here, the number of clusters is known in advance, and the goal is to try to recover the true partition of the data. I used datasets with varying proportions of locus occupancy, from 40–100%. I chose to ex-

amine locus occupancy as low as 40% as it is a realistic level of sparseness in real data generated by rapid protocols such as RAD sequencing (Chattopadhyay, Garg, and Ramakrishnan 2014)—see also the *Chiastocheta* fly dataset examined in Chapter 3. These results are summarised in figure 2.15A. This figure can be interpreted the same way as figure 2.6, with cases getting easier along the x-axis, and success being conveyed by how low the value is on the y-axis. It shows that the true partition is recovered with high accuracy (as measured by variation of information) as long as the clusters are separated by a few topological rearrangements—even when data is sparse. When clusters are not well separated—differing by just one or two SPR rearrangements—sparseness has a detrimental effect on accuracy, as can be seen by the high VI values obtained for 40% (red) and 60% (blue) occupancy at 1 and 2 SPR rearrangements.

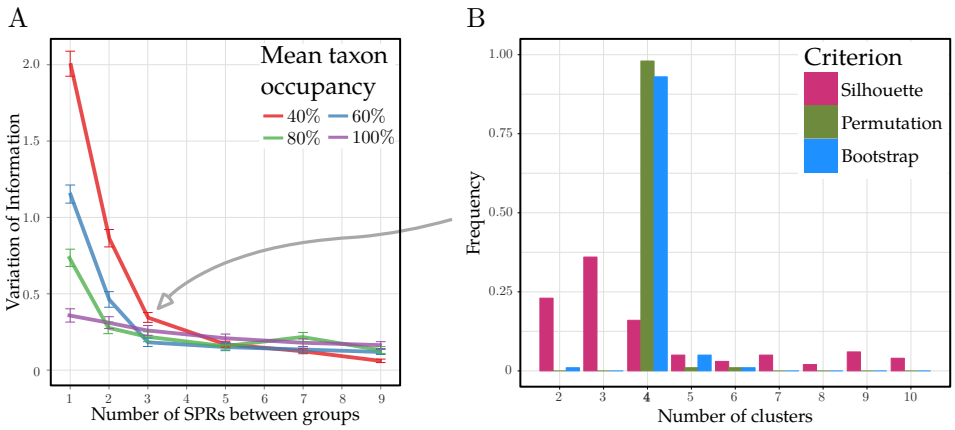


Figure 2.15. *(A) Distance of the spectral clustering of geodesic distances from the "true" clustering for varying levels of locus occupancy. Just as with complete groups, partial groups converge to the correct assignment as the distance between clusters increases. When clusters differ from the underlying common tree by 3 SPRs or more, the effect of incomplete occupancy on performance is very slight. (B) Effect of incomplete locus occupancy on cluster number selection criteria. Non-parametric permutation and parametric bootstrap recover the true number of clusters (4) in more than 90% of cases. The clusters were separated by 3 SPRs, and each locus had 40% mean locus occupancy, which corresponds to the point on panel (A) indicated by the grey arrow.*

I also repeated the approach of section 2.5 on datasets with incomplete locus occupancy, to test whether I could automatically determine the number of clusters using the cluster selection criteria. Figure 2.15B shows an example of the performance of the cluster selection criteria for data with

40% locus occupancy, separated by 3 SPR rearrangements. Here both of the permutation and bootstrap variants of the stopping criterion show high accuracy when inferring the number of clusters, strongly outperforming the silhouette. More complete results are presented in the four panels of figure 2.16. These show the effects that locus occupancy (rows) and cluster separation (columns) have on the cluster selection criteria's ability to select the correct number of clusters. In all but the top-left panel, which corresponds to the case with the least well-separated clusters and the sparsest data, both variants of the special purpose stopping criterion make very accurate calls on the correct number of clusters (which again is 4). Even in the top-left panel the special purpose stopping criterion strongly outperforms the silhouette criterion.

In summary, these results show that the treeCl clustering approach works well even when there is up to 60% missing data, provided the clusters are reasonably well separated, and therefore is suitable for use on real data for which incomplete locus occupancy is likely to be found.

2.7 DISCUSSION

The clustering method described in this chapter is intended to partition incongruent datasets in a process agnostic way. Through simulation it has been shown that clustering based on inter-tree distances can effectively identify clusters of loci that share a common evolutionary history. This is more effective when branch length information is included in the tree comparisons. Spectral clustering and Ward's method are shown to be the best performing clustering methods across a wide range of simulation scenarios. The number of clusters can be estimated by a stopping criterion that involves resampling and reanalysing data, using likelihood to evaluate the improvement in goodness of fit when increasing the number of clusters. Two variants of the stopping criterion were tested; both proved to be better at estimating the number of clusters than the general purpose silhouette measure. Both variants are computationally expensive, but of the two variants the one based on permutation is the least onerous. The method

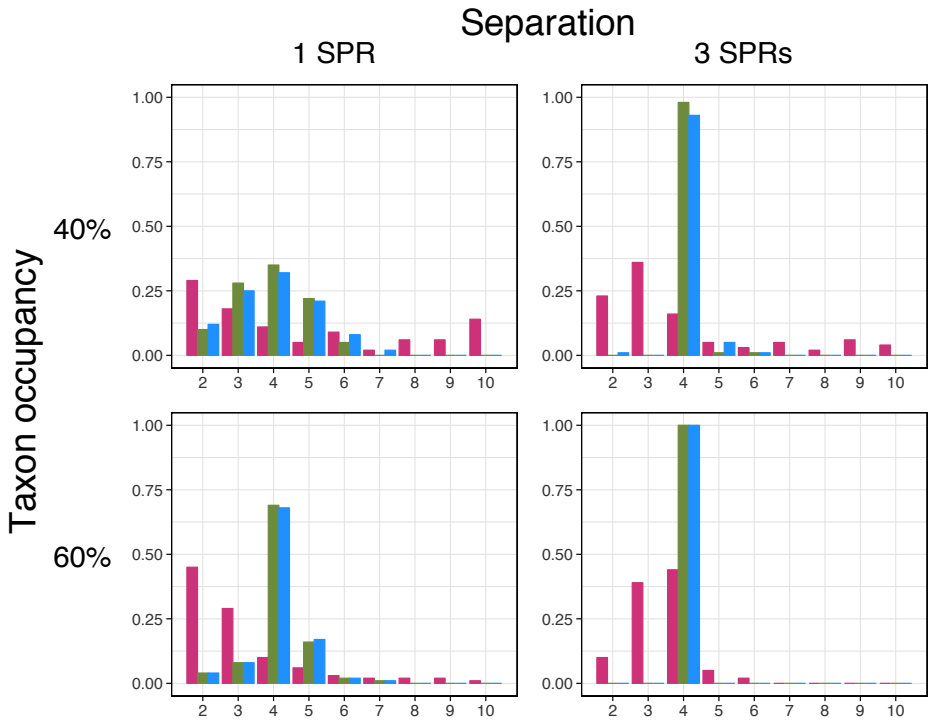


Figure 2.16. *The performance of criteria for determining the number of clusters, with sparse data. The correct number of clusters is 4. Four examples are shown, for two levels of occupancy (40 and 60%; rows), and two levels of cluster separation (1 and 3 SPRs; columns). Occupancy is expressed as a percentage; 60%, for example, means that a sequence was included for each taxon with probability 0.6.*

is robust to missing data. As a result of performing these simulation experiments I favour the use of spectral clustering and geodesic distances, as these produce accurate partitions in a range of scenarios, and the non-parametric permutation based variant of the stopping criterion, as it is as accurate as the parametric bootstrap variant, and less costly to calculate. In the next chapter I apply these versions of the clustering method to real datasets, including cases with missing data.

2.7.1 FUTURE DIRECTIONS

One possible follow-up that could build on this work would be to measure the performance on data that has varying degrees of rate heterogeneity, with loci that share the same tree topology but evolve at different speeds. This scenario was not covered by the simulation design presented above,

but is likely to be common in real data. It may be the case that including branch length information could be misleading when applied to this kind of data, in which case topology-only metrics may show better performance. This could be further assessed by testing other distance metrics, such as the quartet distance or matching.

Another possibility that could have been interesting is to investigate whether selection might have an effect on clustering. This could be done using simulation, like the example shown with incomplete locus occupancy section 2.6, but varying the amount of positive and purifying selection. Simulating sequences would require use of a codon model like GY (Goldman and Yang 1994) that incorporates a selection parameter. To produce realistic sequences this would, like a nucleotide model, benefit from the selection of reasonably realistic parameters, either estimated from a corpus of data or otherwise.

Currently the methods use a homogeneous set of either nucleotide or amino acid alignments as input. It would also be possible to use a mixture of datatypes—also including data made up of other types, such as morphological characters or copy number variation—to build the trees from which clusters are estimated. Further investigation would be required to determine whether the clustering is made more accurate by the inclusion of more data (and presumably more information), or if evolutionary information is obscured, as potentially clustering could split the data by type, rather than by evolutionary history.

As was noted earlier (section 2.5.6), the non-parametric bootstrap procedure assumes that all loci share a single, mixed history. Future work should investigate whether using a null hypothesis with k mixed histories provides better estimates of the correct number of clusters, sufficient to outweigh the increased computational overhead.

A limitation of clustering methods such as the ones presented in section 2.3 is that they are not informed by the evolutionary model used to analyse the clusters. The partition likelihood (see section 2.5.1) would provide an

objective function which could be optimised by altering a partition, in order to find the best evolutionary clustering. Furthermore, no account is made of the uncertainty in the tree estimates. These ideas are expanded in Chapter 4.

3

APPLICATION TO BIOLOGICAL DATA

In the previous chapter I presented the treeCl clustering method, and showed that it is effective at identifying clusters of congruent loci and estimating the number of clusters present when applied to simulated data. While testing on simulated data is useful, it suffers from a number of drawbacks. To give two examples: (i) the models from which we simulate are inadequate at reproducing the nuances of real data, which may lead to methods performing better on simulated data than they would otherwise; and (ii), in order to be able to produce results in a reasonable length of time, it is necessary to restrict investigation to limited ranges of parameter values (and perhaps not test the effect of changing certain parameters at all), reducing the number of situations for which the simulations are informative.

This chapter is concerned, therefore, with testing the method on real data, and with interpreting the results from these studies. As the combination of geodesic distance and spectral clustering was shown to be highly effective across a range of simulation scenarios, it was this combination that I applied to two real datasets: protein sequences from 344 loci derived from 18 ascomycetous yeast species, for which data occupancy was complete—each locus's alignment contained sequence data for each of the 18 species—and nucleotide sequences obtained from RAD sequencing for 176 loci from 306 *Chiastocheta* genus globeflower flies sampled from seven species, with each locus's alignment having on average ~56% of the 306 taxa missing.

3.1 YEAST DATASET

The first application of the treeCl method to empirical data was on a set of 344 curated orthologous sets of protein sequences derived from 18 ascomycetous yeast species.

I chose to look at this dataset because it had previously been analysed by Jacky Hess, a PhD student in my group. She had found that there was widespread incongruence among these loci, and had focussed in her work to find a tree that fitted well to the entire dataset using complex phylogenetic models with locus-specific parameters. My approach instead looks to explain the data in terms of its incongruence. I thought it would be interesting to see if subdividing the loci according to their incongruent nature might provide further evolutionary insights.

These 18 species belong to the subphylum *Saccharomycotina*, and include the very well-characterised laboratory model organism *Saccharomyces cerevisiae*, as well as *Candida albicans*, an important human pathogen. The full list of organisms is given in table 3.1. In this thesis when referring to these species I use the nomenclature of Hess and Goldman (2011); many of these species have been given several different names during the history of their classification, and there is not always a universally accepted name. Table 3.1 lists the NCBI taxonomy database IDs for each species (NCBI Resource Coordinators 2015), to aid disambiguation.

ORGANISM NAME	FAMILY	TAXONOMY ID
Yarrowia lipolytica	Dipodascaceae	4952
Debaryomyces hansenii	Debaryomycetaceae	4959
Pichia guilliermondii	Debaryomycetaceae	4929
Pichia stipitis	Debaryomycetaceae	4924
Lodderomyces elongisporus	Debaryomycetaceae	36914
Candida tropicalis	Debaryomycetaceae	5482
Candida albicans	Debaryomycetaceae	5476
Kluyveromyces waltii	Saccharomycetaceae	4914

Saccharomyces kluyveri	Saccharomycetaceae	4934
Ashbya gossypii	Saccharomycetaceae	33169
Kluyveromyces lactis	Saccharomycetaceae	28985
Candida glabrata	Saccharomycetaceae	5478
Saccharomyces castellii	Saccharomycetaceae	27288
Saccharomyces bayanus	Saccharomycetaceae	4931
Saccharomyces kudriavzevii	Saccharomycetaceae	114524
Saccharomyces mikatae	Saccharomycetaceae	114525
Saccharomyces cerevisiae	Saccharomycetaceae	4932
Saccharomyces paradoxus	Saccharomycetaceae	27291

Table 3.1. *This table lists the 18 yeast species that comprise the yeast dataset. Taxonomic families and identifiers are according to the NCBI taxonomy database.*

The species *Candida glabrata*, *Saccharomyces castellii*, *Saccharomyces bayanus*, *Saccharomyces kudriavzevii*, *Saccharomyces mikatae*, *Saccharomyces cerevisiae* and *Saccharomyces paradoxus* comprise the *Saccharomyces sensu stricto* clade, in which there is inferred to have been a whole genome duplication (WGD) (Wolfe and Shields 1997). Recent investigation has suggested that the WGD is a result of a hybridisation between two closely related ancestral species of yeast, rather than a duplication of the genomic content of a single ancestral species (Marcet-Houben and Gabaldón 2015; Wolfe 2015). If this scenario is true, it is likely that there will be distinct sets of loci in the extant species that find their common ancestor in one or the other of these hybridising lineages, which may manifest as incongruence in the locus trees. Additionally, there may be some incongruence due to differential gene duplication and loss, as a consequence of the widespread loss of genes that followed WGD (only 10–30% of paralogs persist in the *Saccharomyces* genomes; Byrne and Wolfe 2005).

3.1.1 DATA COLLECTION

The data were collected and curated by Jackie Hess, a former PhD student of the Goldman group at the EMBL-European Bioinformatics Institute. These

data were based on the 8 species investigated by Rokas et al. (2003), with the addition of 10 more species for which genome sequences had become available in the intervening time (Hess 2011). Orthology relationships were derived from the Fungal Orthogroups Repository, (FOR; Wapinski et al. 2007) for 14 of the 18 species. FOR assigns orthology based on a procedure that incorporates the SYNERGY algorithm of Wapinski et al. (2007) and the information curated at the Yeast Gene Order Browser database (Byrne and Wolfe 2005). This takes into account both sequence similarity, and synteny information — the position that genes occupy on chromosomes, relative to each other — when assigning orthology relationships. The orthology relationships for the remaining 4 species were inferred and filtered using BLAST (Altschul et al. 1990), using a process described in Hess (2011). The data were previously used by Hess and Goldman (2011) to infer a species phylogeny robust to inter-gene heterogeneities, and it was observed that the data showed a high degree of incongruence: from the 344 loci, 336 different individual locus tree topologies were inferred (Hess 2011).

To investigate the orthology relationships among sequences within loci, I collected additional data to perform reciprocal best hit analysis using BLAST. I downloaded proteomes for each species and used them to build BLAST databases (see section 3.1.2). These were obtained from Ensembl (Flicek et al. 2013), except for the proteome of *S. kluyveri*, not available from Ensembl, which I obtained from FOR.

3.1.2 METHODS

I applied treeCl, using spectral clustering and geodesic distances, to this dataset to partition it into a range of numbers of clusters. Both variants of the stopping criterion were used to choose a statistically supported number of clusters from this range. Phylogenetic inference for both the individual locus trees, and the inference of concatenated cluster trees, was performed with PhyML (Guindon et al. 2010), using the WAG model of sequence evolution (Whelan and Goldman 2001). The WAG model was also used when simulating alignments for the parametric bootstrap stopping criterion.

I also applied a phylogenetic outlier detection package, kdetrees (Weyenberg et al. 2014) to the dataset, using geodesic distances, and otherwise using the default parameters.

I used BLAST (Altschul et al. 1990) to perform reciprocal best hit analysis on the sequences in each of the 344 loci, using *S. cerevisiae* as the reference proteome. These involved searching for each non-*S. cerevisiae* sequence for the single best hit in the *S. cerevisiae* proteome, and also searching for the *S. cerevisiae* sequence from each locus in the other proteomes. I built BLAST databases from the proteomes obtained from the sources given in section 3.1.1, and performed searches locally to obtain the single best hit in each case.

3.1.3 RESULTS AND DISCUSSION

I applied treeCl to partition the dataset into 2–5 clusters. The results of the tests to assess the significance of each number of clusters are shown in figure 3.1.

The upper panel in figure 3.1 shows the results of applying the non-parametric permutation variant of the stopping criterion (the resampled distribution is indicated by green boxplots), and the lower panel shows the parametric bootstrap variant (blue boxplots). In both variants the resampled distributions show a narrow range of Δ_k. Of the two variants, the parametric bootstrap shows perhaps the largest variability of Δ_k among the resampled distributions, as a small number of outlier points are visible. As in the other plots of this type, the 95th percentile of the resampled distribution is shown as a horizontal black bar. The values of Δ_k observed on the original data are shown as red points, connected by lines, in both panels. Both variants of the stopping criterion strongly suggest that the data should be partitioned into three clusters (figure 3.1), as this is the value of k for which the Δ_k from the original data first falls within the distribution of Δ_k observed for the resampled data. The silhouette criterion favours four clusters over three (figure 3.2).

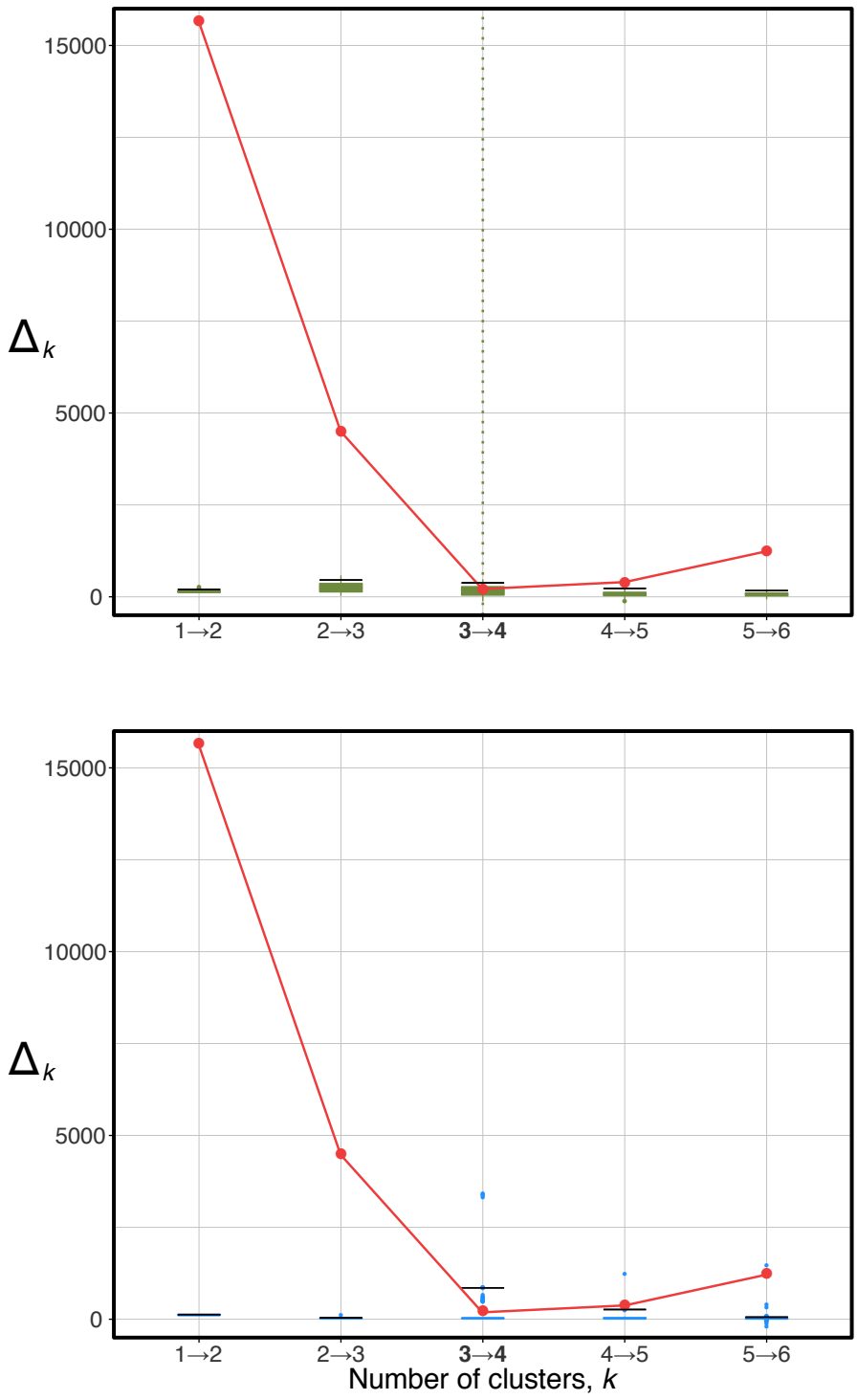


Figure 3.1. *Stopping criteria applied to yeast dataset. The upper panel shows the result of applying the permutation-based (non-parametric) variant of the stopping criterion to the yeast dataset. The lower panel shows the result of applying the parametric bootstrap variant of the stopping criterion. Resampled distributions are based on 100 replicates. Both variants suggest that the data should be partitioned into three clusters.*

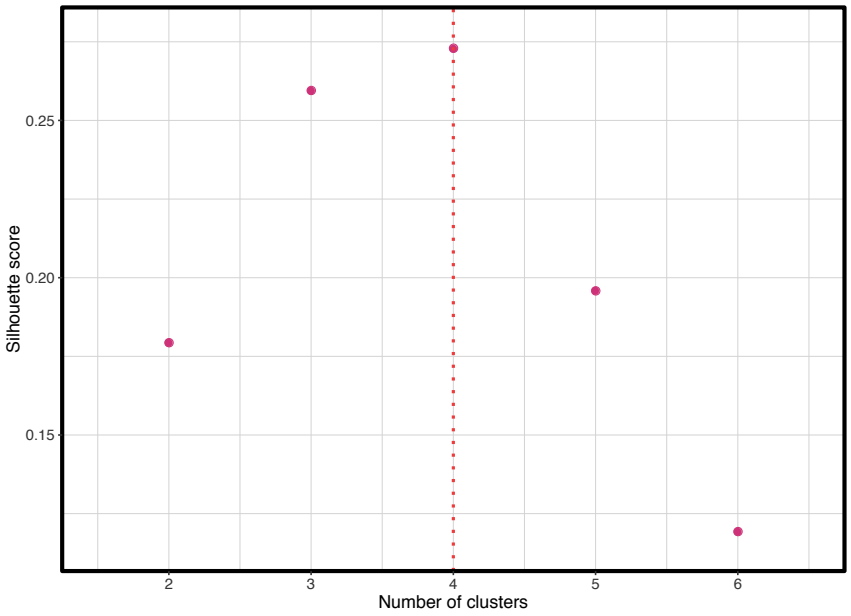


Figure 3.2. *Silhouette score of yeast clusters. This plot shows the silhouette score obtained when clustering the yeast data set into 2–6 clusters. The number of clusters that yields the highest silhouette score should be chosen using the silhouette criterion. In this case the chosen number is four.*

Partitioning the yeast loci into three clusters results in clusters of unequal sizes: a large cluster, consisting of 307 loci, and two small clusters containing 26 loci and 11 loci. Although the numbering of clusters produced by treeCl has no special meaning, I will consistently use 'cluster 1' to refer to the cluster of 307 loci, and 'cluster 2' and 'cluster 3' to refer to the clusters of 26 and 11 loci, respectively. The assignment of loci to clusters is given in table 3.2.

CLUSTER	LOCI ASSIGNED TO CLUSTER				
1	YAL009W	YAL010C	YAL032C	YAL044C	YAR003W
	YAR007C	YBL015W	YBR061C	YBR095C	YBR101C
	YBR111C	YBR135W	YBR170C	YBR193C	YBR217W

YBR243C	YBR248C	YBR251W	YBR254C	YBR260C
YBR271W	YBR282W	YCL016C	YCL031C	YCL034W
YCL052C	YCL054W	YCL059C	YCR047C	YDL033C
YDL045C	YDL051W	YDL087C	YDL098C	YDL100C
YDL116W	YDL165W	YDL198C	YDL201W	YDL205C
YDL207W	YDL212W	YDL215C	YDR036C	YDR041W
YDR049W	YDR101C	YDR121W	YDR140W	YDR148C
YDR152W	YDR175C	YDR204W	YDR265W	YDR292C
YDR298C	YDR301W	YDR306C	YDR315C	YDR339C
YDR361C	YDR362C	YDR364C	YDR373W	YDR405W
YDR449C	YDR454C	YDR460W	YDR465C	YDR468C
YDR470C	YDR496C	YDR529C	YEL024W	YEL037C
YEL038W	YEL050C	YEL062W	YER009W	YER012W
YER016W	YER023W	YER043C	YER049W	YER078C
YER086W	YER107C	YER136W	YER156C	YER183C
YFL017C	YFL046W	YFR044C	YFR050C	YFR052W
YGL017W	YGL018C	YGL040C	YGL043W	YGL085W
YGL110C	YGL122C	YGL129C	YGL137W	YGL192W
YGL243W	YGR005C	YGR057C	YGR078C	YGR080W
YGR083C	YGR095C	YGR149W	YGR150C	YGR171C
YGR172C	YGR178C	YGR193C	YGR208W	YGR215W
YGR255C	YGR262C	YGR285C	YHL004W	YHL013C
YHR007C	YHR011W	YHR025W	YHR058C	YHR059W
YHR062C	YHR088W	YHR114W	YHR132C	YHR144C
YHR187W	YHR191C	YHR193C	YJL006C	YJL011C
YJL030W	YJL033W	YJL046W	YJL072C	YJL085W
YJL104W	YJL115W	YJL121C	YJL166W	YJL180C
YJL203W	YJL208C	YJR006W	YJR010W	YJR014W
YJR050W	YJR052W	YJR062C	YJR063W	YJR073C
YJR102C	YJR113C	YJR119C	YJR121W	YJR122W
YKL003C	YKL016C	YKL028W	YKL041W	YKL045W
YKL047W	YKL058W	YKL080W	YKL119C	YKL138C
YKL149C	YKL175W	YKL179C	YKL193C	YKL205W
YLR015W	YLR017W	YLR023C	YLR026C	YLR051C
YLR059C	YLR078C	YLR084C	YLR195C	YLR239C
YLR244C	YLR253W	YLR288C	YLR292C	YLR330W
YLR370C	YLR409C	YLR418C	YML004C	YML021C
YML030W	YML036W	YML080W	YML096W	YML110C
YML121W	YML127W	YMR002W	YMR009W	YMR013C
YMR015C	YMR026C	YMR038C	YMR055C	YMR061W

	YMR131C	YMR150C	YMR188C	YMR193W	YMR197C
	YMR201C	YMR208W	YMR211W	YMR218C	YMR236W
	YMR241W	YMR260C	YMR276W	YMR281W	YMR290C
	YMR314W	YNL005C	YNL010W	YNL025C	YNL071W
	YNL072W	YNL177C	YNL220W	YNL252C	YNL287W
	YNL291C	YNL292W	YNL306W	YNL308C	YNL310C
	YNL315C	YNR007C	YNR017W	YNR039C	YNR052C
	YNR054C	YOL010W	YOL022C	YOL041C	YOL093W
	YOL124C	YOL135C	YOL142W	YOL145C	YOR065W
	YOR067C	YOR070C	YOR077W	YOR095C	YOR111W
	YOR128C	YOR142W	YOR150W	YOR160W	YOR164C
	YOR166C	YOR168W	YOR211C	YOR236W	YOR243C
	YOR249C	YOR250C	YOR251C	YOR262W	YOR271C
	YOR289W	YOR361C	YOR370C	YPL001W	YPL002C
	YPL030W	YPL065W	YPL094C	YPL104W	YPL107W
	YPL109C	YPL126W	YPL149W	YPL157W	YPL172C
	YPL183C	YPL210C	YPL239W	YPR031W	YPR056W
	YPR073C	YPR103W	YPR110C	YPR133C	YPR139C
	YPR143W	YPR166C			
2	YBL080C	YBR094W	YBR290W	YCR068W	YDL104C
	YEL053C	YFR051C	YGL236C	YHR019C	YHR020W
	YHR024C	YHR075C	YJL054W	YJL071W	YKL060C
	YMR224C	YNL219C	YNL232W	YNL256W	YNL325C
	YNR029C	YOL097C	YOR125C	YPL188W	YPL244C
	YPR118W				
3	YDL043C	YDR023W	YDR448W	YHR201C	YJL025W
	YJR141W	YKR038C	YLR209C	YOL005C	YOR201C
	YPR025C				

Table 3.2. *This table lists all 344 loci in the yeast dataset. The data at each locus are putatively orthologous sequences taken from the 18 ascemycetous yeast species described in the text. The loci are listed according to the name of the locus in* Saccharomyces cerevisiae. *The loci are listed according to their cluster membership when data are partitioned into 3 clusters using treeCl with spectral clustering and geodesic distances, which was the number of clusters supported by the both variants of the stopping criterion.*

Despite the high degree of incongruence among trees estimated from individual loci, the overall species tree relating these yeasts has been well-studied, and has been established with little controversy (see Dujon 2010 for a review). This species tree can be seen as the tree on the left in fig-

ure 3.3, which is also the cluster tree derived from cluster 1. The trees on the right of figure 3.3 are the cluster trees inferred for clusters 2 and 3.

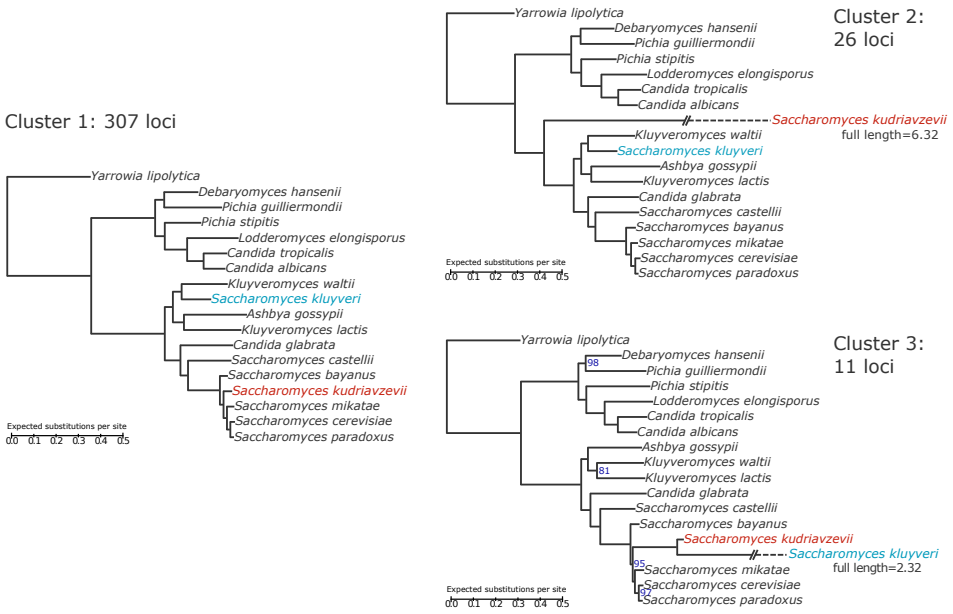


Figure 3.3. *Phylogenetic trees inferred from the three clusters found in the yeast analysis with treeCl. The tree on the left is that inferred from the largest cluster of 307 loci. This matches the established species tree for these 18 species of yeast. The taxa highlighted in red (*S. kudriavzevii*), and blue (*S. kluyveri*), are those that are found on long branches in the trees inferred from clusters 2 and 3, shown on the right (respectively right, upper, and right, lower). In these trees the branches leading to* S. kudriavzevii *(in cluster 2) and* S. kluyveri *(in cluster 3) have been truncated to so as to fit reasonably on the plot. Their full lengths are indicated. Otherwise, branch lengths can be determined by the scale bars shown (all equal scales). Where aBayes branch supports are less than the maximum possible value of 100% their values are indicated by a number to the right of the branch.*

The tree for cluster 2 yields nearly the same topology as that for cluster 1, with the modification that *S. kudriavzevii* appears basal to, rather than within, the *Saccharomyces sensu stricto* clade. Branch lengths are also modified in the cluster 2 tree: the branch leading to *S. kudriavzevii* is very much longer than in the cluster 1 tree—so much longer that it has been truncated to fit on the figure.

A similar observation can be made of the inferred cluster tree from cluster 3. In this case, it is *S. kluyveri* that is placed on a very long branch. Topologically, cluster 3 also differs from cluster 1, specifically in the arrangement of the clade consisting of the species *K. waltii*, *A. gossypii* and *K. lactis*, which

is the clade to which *S. kluyveri* belongs in the other two trees. Otherwise, the cluster 3 tree topology shows the same relationships among the Debaryomycetaceae family, among the *Saccharomyces sensu stricto*, and in the position of the outgroup. Cluster 3's tree is also the only one for which the branch support values, as measured using approximate Bayes (Anisimova et al. 2011), take values below 100%. The lowest branch support, 81%, is found within the rearranged *K. waltii*, *A. gossypii*, *K. lactis* clade. With this exception, the remaining branches all show greater than 95% approximate Bayes branch support, even though there is incongruence among the loci underlying these trees. However, this may not necessarily be a strong case for these topologies being correct, as it has been suggested that concatenation tends to inflate branch support values (Larget et al. 2010; Weisrock et al. 2012).

As an attempt to visualise the distribution of the individual locus trees I computed their coordinates using multidimensional scaling (see section 2.3.2, and also Chapter 4 for further details). The arrangement of points in two dimensions is shown in figure 3.4. In this figure the loci belonging to cluster 1 are shown as red circles, loci belonging to cluster 2 as blue triangles, and those belonging to cluster 3 as green triangles. Cluster 1 appears as a very tight cluster of points in the centre of the figure. Clusters 2 and 3 are more diffuse, with cluster 2 tending towards the lower-left quadrant, and cluster 3 to the upper-right. Clusters 1 and 3 appear to overlap, though one should keep in mind that, while it may seem to be difficult to assign these clusters on the basis of this figure, the actual clustering is done in a higher dimensional space, and using a different coordinate transform, than the one visualised here. What can be noted from this figure is that all members of clusters 2 and 3 are positioned relatively large distances away from cluster 1, which suggests that these clusters consist of loci for which the underlying tree distances are large, when measured from those loci from cluster 1. The clusters' genomic locations show no particular pattern figure 3.5.

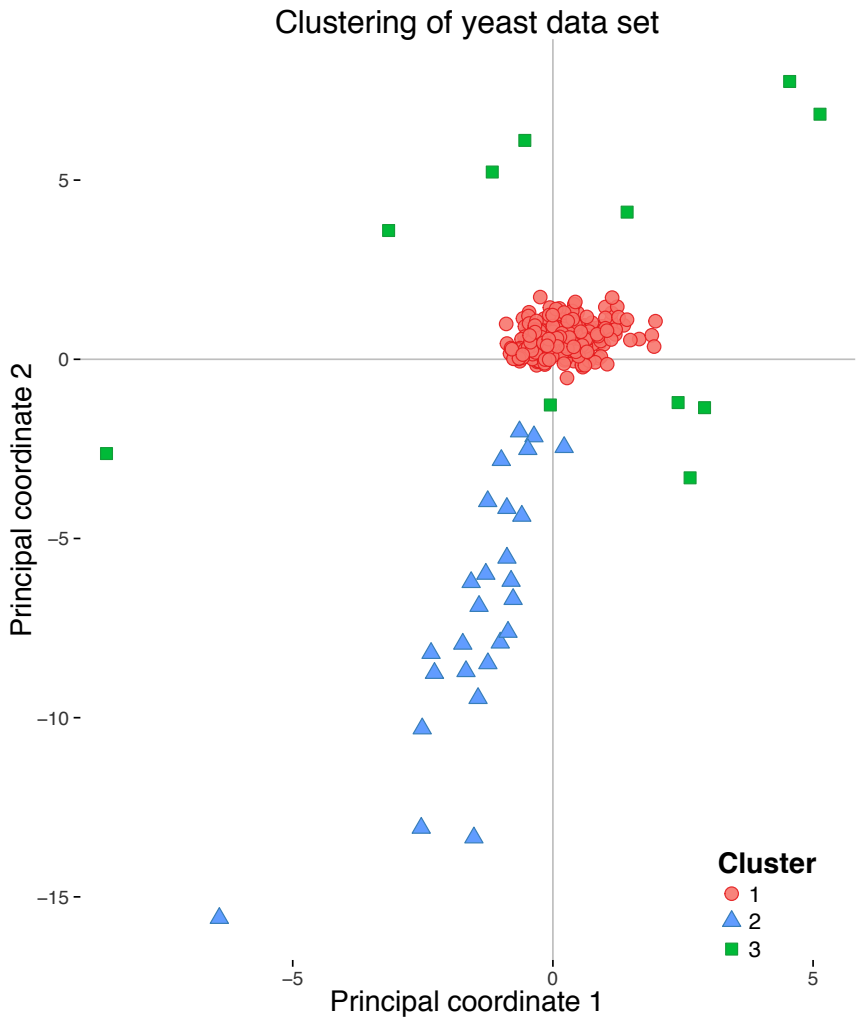


Figure 3.4. *Visualisation of application of treeCl to the yeast dataset. The scatterplot shows the embedding, by multidimensional scaling, of the geodesic distances between the 344 trees. Three clusters were found by spectral clustering: red circles indicate the largest cluster, with 307 members; the 37 remaining loci are indicated by blue triangles (cluster 2) and green squares (cluster 3). Loci belonging to the first, largest cluster are tightly grouped and when analysed together yield the established species phylogeny, whereas trees belonging to the second and third clusters are disparate and all have odd and inconsistent phylogenies as result of incorrectly called orthology (see text for full details).*

To try to understand the source of incongruence in the smaller clusters, I returned to the individual locus trees to see if they offered any insight. The 37 trees of clusters 2 and 3 are reproduced in table 3.2. Out of the loci for clusters 2 and 3, none of the single-locus topologies matched the one inferred for its cluster as a whole. Among these 37 loci, each tree has usually one or at most two taxa associated with a long branch: long compared to

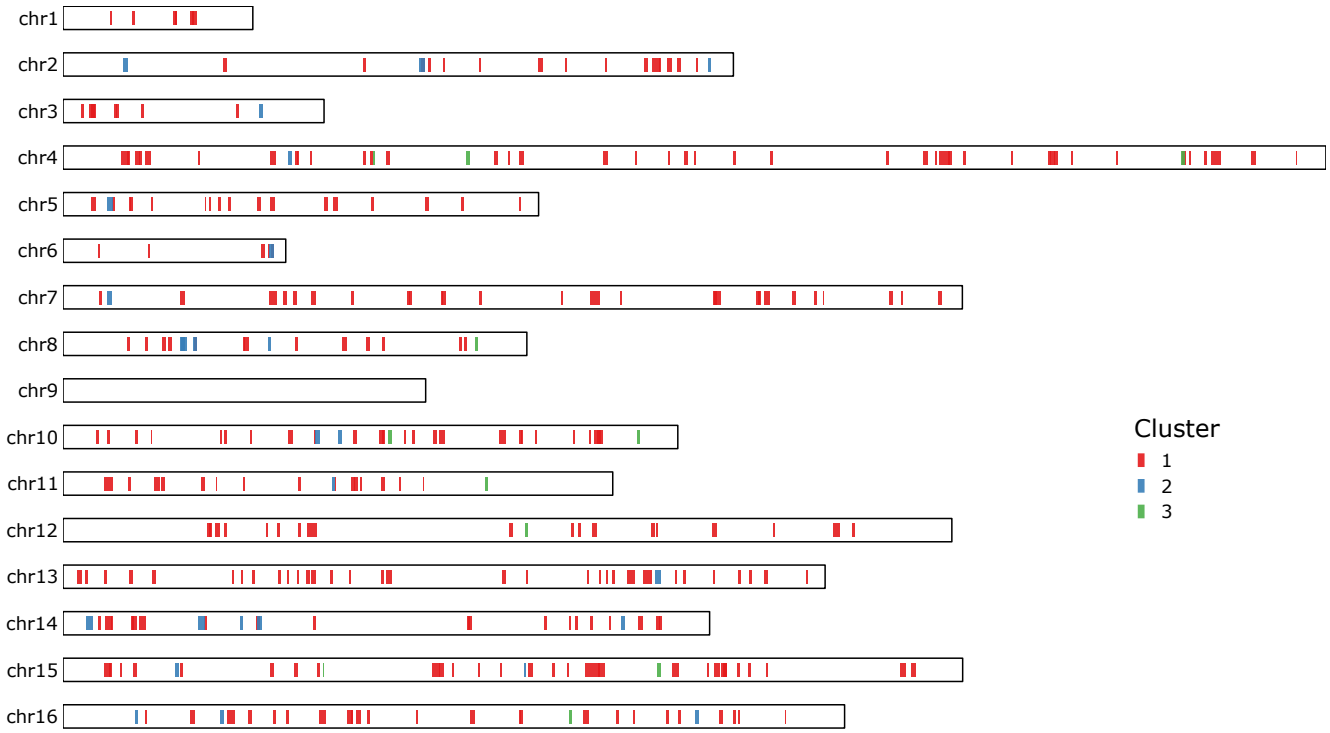
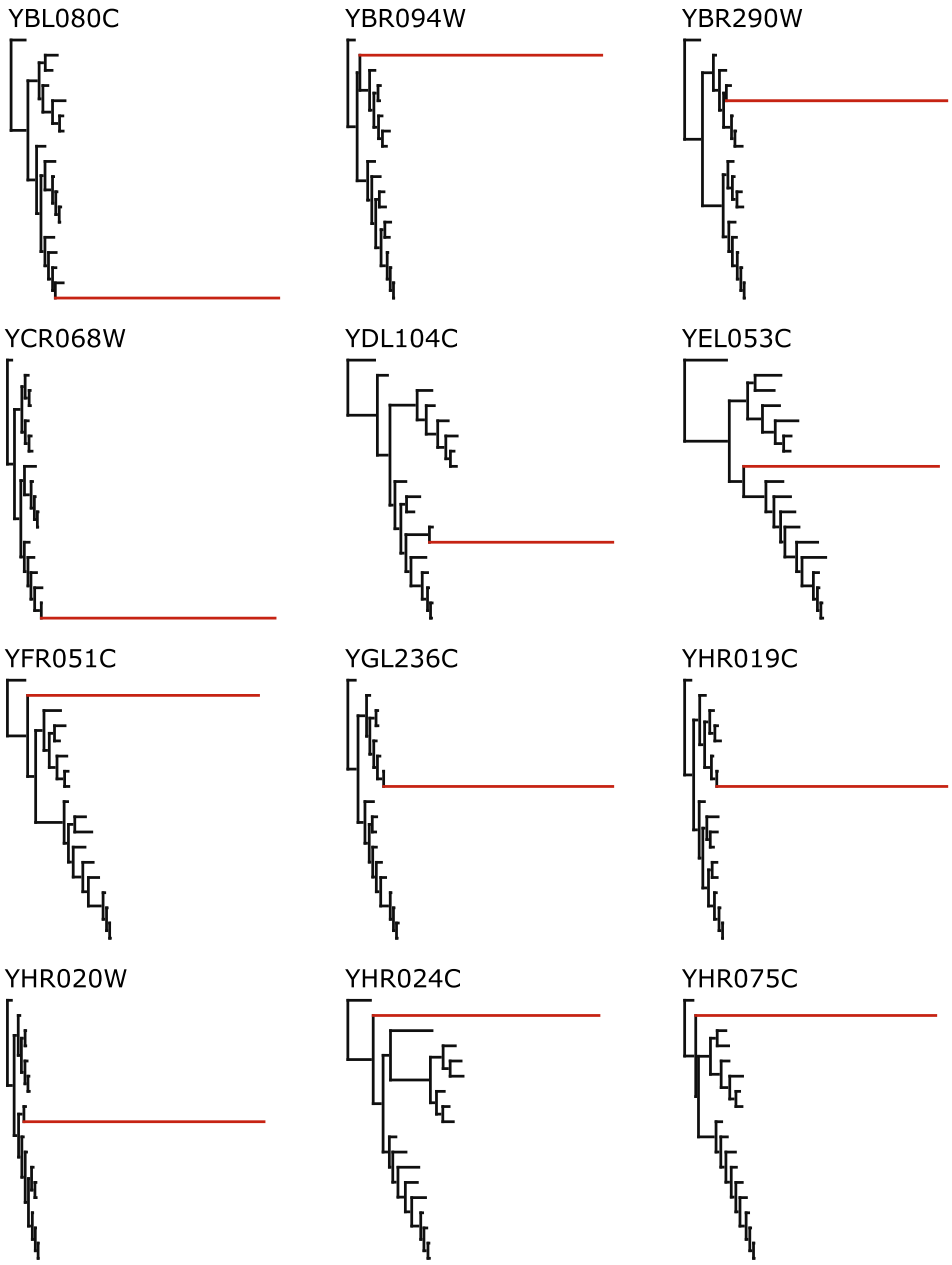


Figure 3.5. *Genomic location of clustered loci. The diagram shows the arrangement along the sixteen yeast chromosomes of the 344 loci, coloured by their cluster memberships. The colour scheme is the same as in figure 3.4*

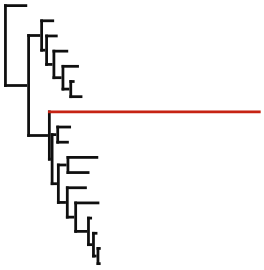
the average length of branches in the tree, and much longer than one would expect compared to the tree inferred from cluster 1 (left-most tree in figure 3.3). These branches are highlighted in red. This indicates that within each tree, the sequences for these taxa are evolutionarily very distant from the sequences belonging to the other taxa, so much so that it suggests that the distant sequences may not be orthologs of the remainder.

What table 3.2 makes clear is that among the 37 loci only a subset of the taxa are associated with long branches — most frequently these are *S. kudriavzevii* and *S. kluyveri*. Also, the distribution of affected taxa between the clusters is not random. In all 26 loci in cluster 2 the long branches lead exclusively to *S. kudriavzevii*. Contrast this with cluster 3, within which — out of the 11 loci — in five loci the long branch is associated with *S. kluyveri*, in three it leads to *S. kluyveri* plus one other, in two to a pair of taxa neither of which is *S. kluyveri* or *S. kudriavzevii*, and in one to the outgroup, *Y. lipolytica*.

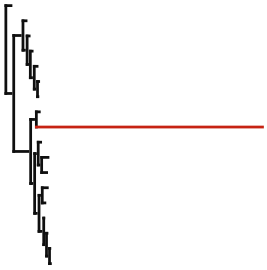
The distribution of affected taxa between cluster 2 and cluster 3 suggests the following interpretation of the three clusters: cluster 1 contains the loci with correctly called orthology, cluster 2 contains the loci for which the sequence from *S. kudriavzevii* is potentially not an ortholog, and cluster 3 is a catch all of those loci for which there is some mistakenly inferred orthology, but not restricted to *S. kudriavzevii*.



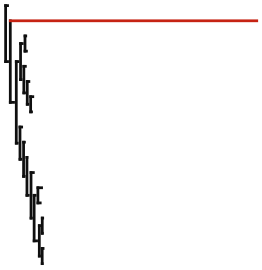
YJL054W



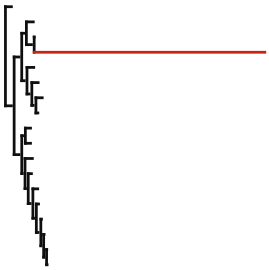
YJL071W



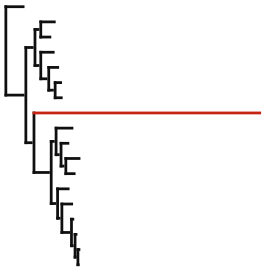
YKL060C



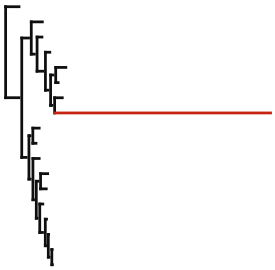
YMR224C



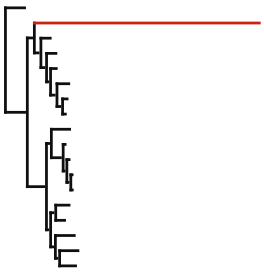
YNL219C



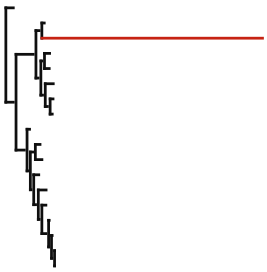
YNL232W



YNL256W

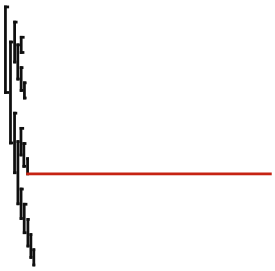


YNL325C

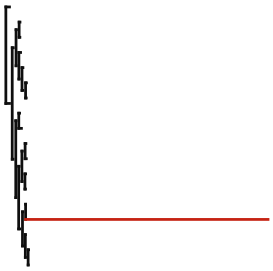


YNR029C

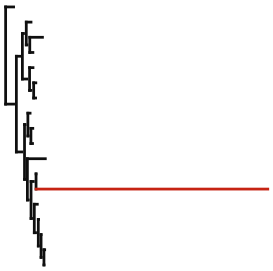
YOL097C



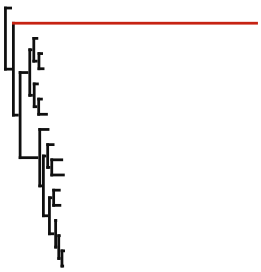
YOR125C



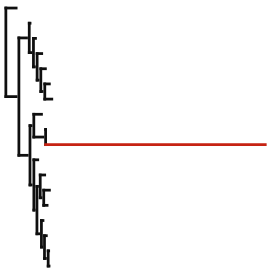
YPL188W



YPL244C



YPR118W



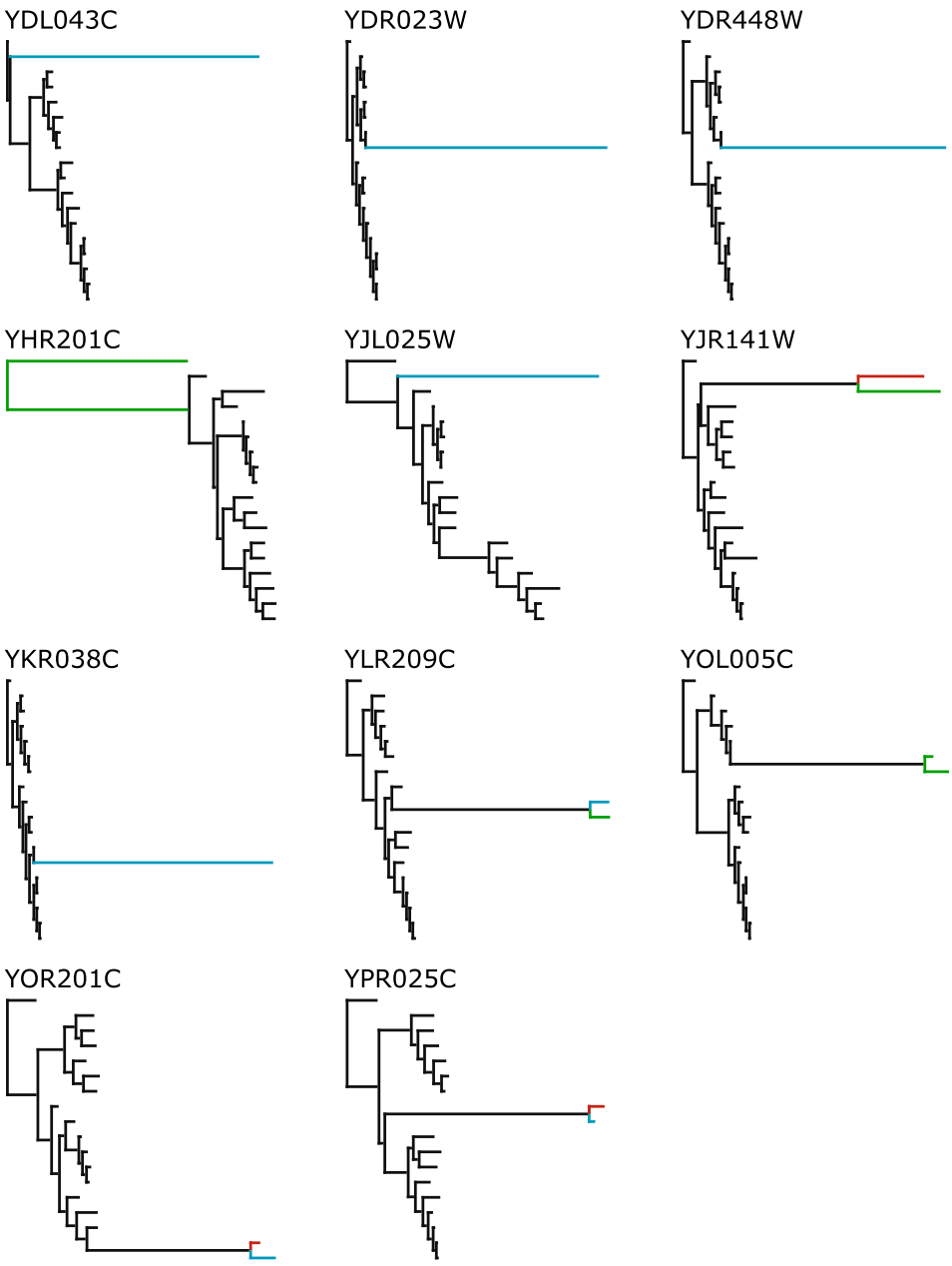


Figure 3.6. *Phylogenetic trees for the 37 yeast loci discovered to have erroneous orthology, with the non-orthologous sequences highlighted as dashed lines. The first 26 trees (comprising the first nine rows) are the loci from cluster 2; the remaining 11 trees are those from cluster 3. Within clusters the locus trees are arranged alphabetically by gene name. Certain species are coloured: red,* S. kudriavzevii*; blue,* S. kluyveri*; green, other species affected by incorrectly called orthology (see text for details).*

To investigate this possible failure of orthology inference, I used BLAST (Altschul et al. 1990) to perform reciprocal best hit analyses of the se-

quences associated with the long branch in each tree with the sequence from *S. cerevisiae*. To illustrate the approach I took, consider the locus YBL080C (that is, the orthologous group containing the *S. cerevisiae* gene YBL080C) as an example. The tree inferred for this orthologous group has a noticeably long branch that leads to *S. kudriavzevii* (table 3.2, top-left tree). BLASTing the *S. kudriavzevii* sequence from the YBL080C orthologous group against an *S. cerevisiae* database results in a close match for gene YMR219W, rather than YBL080C. YBL080C was not found among the close matches to this sequence. Doing the complementary search, querying an *S. kudriavzevii* database with the *S. cerevisiae* YBL080C sequence, results in a match to a sequence other than the one that was included in the YBL080C orthologous group for *S. kudriavzevii*. These results suggest that the *S. kudriavzevii* sequence that was included as an ortholog of *S. cerevisiae* for the locus YBL080C was not in fact an ortholog. Similar analyses were performed for all orthogroups for the 344 loci, for all species. Only for the 37 loci belonging to clusters 2 and 3 were any discrepancies found.

The results of similar analyses for each of the other loci in clusters 2 and 3, shown in table 3.3, are similar to the case just presented and indicate that they were all erroneously classified as orthologs. The branches leading to the misannotated orthologs correspond to those highlighted in red in table 3.2. I thus conclude that the major source of incongruence in this 344 gene yeast dataset is derived from errors in designating orthology relationships among the sequences assigned to each of the 37 loci discovered in clusters 2 and 3. These errors particularly involve the *S. kudriavzevii* genome, which is implicated in all 26 loci in cluster 2, as well as in three loci in cluster 3.

To summarise the clusters, cluster 1 consists of all loci for which orthology relationships seem to be correctly inferred, while the common feature of clusters 2 and 3 is that they are affected by erroneous orthology relationships: in cluster 2 these exclusively involve *S. kudriavzevii*; in cluster 3 these involve sequences from another species, not *S. kudriavzevii*, or from two species (one of which may be *S. kudriavzevii*).

ORTHOLOGOUS GROUP	MISANNOTATED SPECIES	BEST HIT IN *S. CEREVISIAE*	CLUSTER
YBL080C	*Saccharomyces kudriavzevii*	YMR219W	2
YBR094W	*Saccharomyces kudriavzevii*	YLR357W	2
YBR290W	*Saccharomyces kudriavzevii*	YLR114C	2
YCR068W	*Saccharomyces kudriavzevii*	YJR107W	2
YDL104C	*Saccharomyces kudriavzevii*	YKR038C	2
YEL053C	*Saccharomyces kudriavzevii*	YOL080C	2
YFR051C	*Saccharomyces kudriavzevii*	YPL259C	2
YGL236C	*Saccharomyces kudriavzevii*	YBL098W	2
YHR019C	*Saccharomyces kudriavzevii*	YCR024C	2
YHR020W	*Saccharomyces kudriavzevii*	YER087W	2
YHR024C	*Saccharomyces kudriavzevii*	YLR163C	2
YHR075C	*Saccharomyces kudriavzevii*	YLR133W	2
YJL054W	*Saccharomyces kudriavzevii*	YBL052C	2
YJL071W	*Saccharomyces kudriavzevii*	YPR185W	2
YKL060C	*Saccharomyces kudriavzevii*	YER043C	2
YMR224C	*Saccharomyces kudriavzevii*	YAL035W	2
YNL219C	*Saccharomyces kudriavzevii*	YGL142C	2
YNL232W	*Saccharomyces kudriavzevii*	YBL052C	2
YNL256W	*Saccharomyces kudriavzevii*	YPL070W	2
YNL325C	*Saccharomyces kudriavzevii*	YNL106C	2
YNR029C	*Saccharomyces kudriavzevii*	YPL009C	2
YOL097C	*Saccharomyces kudriavzevii*	YGR185C	2
YOR125C	*Saccharomyces kudriavzevii*	YER086W	2
YPL188W	*Saccharomyces kudriavzevii*	YKR056W	2
YPL244C	*Saccharomyces kudriavzevii*	YEL004W	2
YPR118W	*Saccharomyces kudriavzevii*	YKR026C	2
YDL043C	*Saccharomyces kluyveri*	YDL051W	3
YDR023W	*Saccharomyces kluyveri*	YHR011W	3
YDR448W	*Saccharomyces kluyveri*	YFR037C	3
YHR201C	*Yarrowia lipolytica*	YMR052C-A	3
YJL025W	*Saccharomyces kluyveri*	YDR285W	3
YJR141W	*Saccharomyces kudriavzevii*	YLR019W	3
	Pichia stipitis	YNL144C-like	

YKR038C	*Saccharomyces kluyveri*	YDL104C	3
YLR209C	*Saccharomyces kluyveri*	YLR017W	3
	Pichia stipitis	YLR017W	
YOL005C	*Candida tropicalis*	YNL113W	3
	Pichia guilliermondii	YNL113W	
YOR201C	*Saccharomyces kudriavzevii*	YLR051C	3
	Saccharomyces kluyveri	YLR051C	
YPR025C	*Saccharomyces kudriavzevii*	YNL025C	3
	Saccharomyces kluyveri	YNL025C	

Table 3.3. *Table of the misannotated yeast orthologs found in clusters 2 and 3. Each row gives the name of the locus in* S. cerevisiae*, the species in which it is misannotated, the name of the locus that is wrongly included, and the cluster to which the locus was assigned.*

Given the "outlier" nature of the loci identified in the small clusters, I also applied a specialised outlier detection package entitled kdetrees (Weyenberg et al. 2014). This fits a kernel density estimate to the inter-tree distances, and estimates the density at points corresponding to each of the sampled trees. From these densities kdetrees is able to discern 'outlier' trees from trees that are drawn from the 'true' tree distribution. Remarkably, with geodesic distances, it identified as outliers precisely those 37 loci that were assigned to clusters 2 and 3 when analysed with treeCl (figure 3.7). This provides additional evidence that these 37 loci should indeed be excluded in the inference of the species tree. The advantage that treeCl offers is that these 37 trees were subdivided further into two clusters: one in which the incongruence is due solely to the misannotated orthology of sequences derived from *S. kudriavzevii*, and one in which the mistaken orthology assignments are from more complex combinations of species.

3.1.4 SUMMARY

Application of treeCl to this dataset of yeast loci has demonstrated that treeCl can be applied to realistic, biological datasets of significant size, and perform well. In this case I was able to identify cases of mis-ascribed orthology. Aside from showing the value of treeCl in a practical application, this highlights the importance, and difficulty, of identifying ortholog-

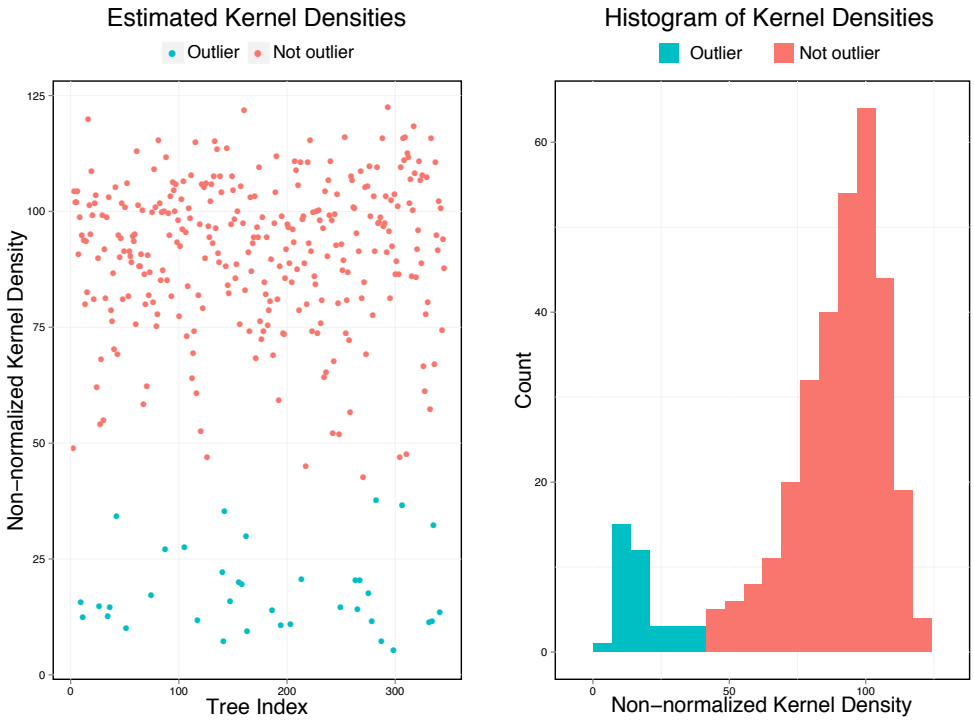


Figure 3.7. *Application of kdetrees to yeast dataset. The scatterplot in the left panel shows the kernel density estimate for each tree. The order of the trees along the* x*-axis is arbitrary with respect to cluster membership, rather being derived from the alphabetical ordering of the names of the loci. The right panel shows a histogram of the kernel density scores. In both panels, the outliers are coloured blue.*

ous sequences when assembling a dataset for phylogenetic use. Orthology assignments were based upon those in the Fungal Orthogroups Repository (FOR), which are based on a mixture of a purpose-built orthology detection algorithm, SYNERGY, syntenic information, and manual curation. It is telling that the species with the most errors in orthology-assignment, *S. kudriavzevii*, did not have its orthology called using multiple sources of evidence, as in FOR's approach, but rather by an *ad hoc* process based on BLAST. This emphasises the importance of using principled techniques when detecting orthology between species.

While this result is perhaps less spectacular than discovering an important biological result, it does showcase the process-agnostic nature of treeCl: it less likely that a mechanistic model of phylogenetic incongruence would have been able to produce these results. This assertion is supported by the

fact that kdetrees, which may also be considered 'process-agnostic', also detected the loci affected by miscalled orthology as outliers.
The severe incongruence owing to misannotated orthology may be masking lesser incongruence — perhaps of genuine biological interest — within cluster 1. It seems not, however, as if the entire treeCl analysis is rerun using only the loci belonging to cluster 1, then treeCl finds a single cluster. This suggests that if there is incongruence due to differential gene duplication and loss or the whole genome duplication, then it is not reflected in these loci.

3.2 CHIASTOCHETA

Here I apply the treeCl clustering method to nucleotide sequence data obtained from globeflower flies. The globeflower flies, genus *Chiastocheta*, are pollinators and seed parasites of the plant species from the *Trollius* genus of globeflowers, family Ranunculaceae (Pellmyr 1992; Suchan et al. 2015). *Chiastocheta* have a recent origin, with most diversification events occurring less than ca. 1.6 million years ago, and their phylogenetic relationships are uncertain (Després et al. 2002), even to the level of species delimitation: of the seven putatively separate recognised species of globeflower flies, only two species were found to be phylogenetically supported using mitochondrial markers (Espíndola, Buerki, and Alvarez 2012). The aim of my work with treeCl on this example to see if meaningful phylogenetic relationships can be discovered on a new, large dataset obtained from these organisms.

3.2.1 RAD SEQUENCING

The data were obtained using restriction site-associated DNA sequencing (RAD sequencing; Baird et al. 2008). Samples were taken from numerous individuals covering the seven putatively separate species of *Chiastocheta* (see section 3.2.2 for details). RAD sequencing isolates DNA fragments by digesting genomic DNA with a restriction enzyme, ligating these fragments to an adapter, and shearing the overhangs to random lengths. Early RAD sequencing methods would identify single nucleotide polymorphisms (SNPs) from these fragments by hybridising them with microarrays

(Baird et al. 2008). In more modern methods SNPs are identified through high-throughput sequencing (Peterson et al. 2012).

RAD sequencing is simultaneously a boon and a problem for phylogeneticists. Among its positive aspects are that it provides a comparatively cheap and abundant source of sequence data, and that it can be applied directly to non-model organisms for which there is no established reference genome (Davey et al. 2011); a negative aspect is that, due to it being targeted towards restriction digest sites that must already exist within an organism's genome (or, indeed, multiple organisms' genomes in a comparative study), there is little opportunity to predict ahead of time which loci will be sequenced — loci are sampled when they share restriction sites that have been maintained through evolution. As many of these restriction sites are only maintained contingently, and not through the maintenance of a genetic function, they are removed from strains at a rate proportional to the background mutagenic rate, and are not preserved through evolutionary selective means. Many orthologous positions in the genomes of related species are lost to RAD sequencing analysis because a flanking restriction site has been lost (Arnold et al. 2013; Gautier et al. 2013; Chattopadhyay, Garg, and Ramakrishnan 2014).

3.2.2 *CHIASTOCHETA* DATA COLLECTION

To investigate the phylogenetic relationships of *Chiastocheta*, I obtained a dataset of *Chiastocheta* globeflower flies (*Diptera*: Anthomyiidae) from collaborators Nadir Alvarez, Nils Arrigo and Tomasz Suchan at the University of Lausanne. Data collection was performed by the Alvarez group, who sampled from various locations reflecting the whole European range of *Chiastocheta*. RAD sequencing — also performed by the Alvarez group — of 306 samples from 7 European *Chiastocheta* species (25 *C. dentifera* individuals, 48 *C. inermella*, 52 *C. lophota*, 34 *C. macropyga*, 70 *C. rotundiventris*, 36 *C. setifera*, 41 *C. trollii*) yielded a data matrix of 5574 orthologous sets of sequences (loci), containing in total 253 866 variable and 81 379 parsimony informative sites. Because of inherent technical limitations of RAD sequencing (Chattopadhyay, Garg, and Ramakrishnan 2014), the ma-

jority of these loci had sparse coverage over the individuals. To focus on the phylogenetically most informative loci, I disregarded loci present in fewer than 100 individuals. This resulted in a matrix of 176 loci, retaining 31 648 of the parsimony informative sites. These had an average length of 228 sites, in the range 192–306. Each locus contained, on average, 44.2% of the taxon set.

3.2.3 METHODS

Data collection was made according to the following protocols: samples were genotyped using a modified ddRAD protocol (Peterson et al. 2012; Mastretta-Yanes et al. 2015). *De novo* locus assembly was performed using the pyRAD 2.0 package (Eaton 2014), for which the read clustering similarity threshold was set to 75%, on both the within- and among-sample levels. Other parameter values were as follows: all nucleotides with Phred quality lower than 20 were treated as unknown bases, and reads with more than 4 unknown bases were removed from the dataset. Possible paralogs were removed by filtering out loci that had more than five heterozygous positions per locus within individuals, more than 10 heterozygotes per nucleotide position among samples, and loci for which more than two alleles were present per individual. In total 273 individuals were sequenced, with 33 technical replicates.

For the purpose of this study, only high coverage loci (i.e. present in at least 100 individuals) were retained. Low coverage loci were excluded as their inclusion would only add to the computational complexity of the problem, while being likely to add little extra discriminatory information. This resulted in a matrix of 176 loci covering the 306 samples. Phylogenetic analysis was performed using the GTR model plus four categories of gamma-distributed rates across sites, using RAxML (Stamatakis 2014). The clustering mechanism used was geodesic distances combined with spectral clustering, and the number of clusters was estimated using both the non-parametric permutation, and the parametric bootstrap variants of the stopping criterion.

3.2.4 RESULTS AND DISCUSSION

After applying treeCl using geodesic distances and spectral clustering, I initially used the non-parametric permutation variant (100 replicates) of the stopping criterion to estimate the statistically supported number of clusters in the *Chiastocheta* dataset. The permutation stopping criterion (100 replicates) identified eight clusters. However, the plot of the likelihood improvement against the number of clusters (figure 3.8) is not smooth: by far the largest likelihood improvements are obtained by increasing the number of clusters up to four and by increasing it from five to six; in contrast, adding a fifth or seventh cluster only moderately improves the fit. Thus, a cautious interpretation of the results of the non-parametric permutation stopping criterion analysis is that there are at least four, and possibly up to eight distinct clusters of loci.

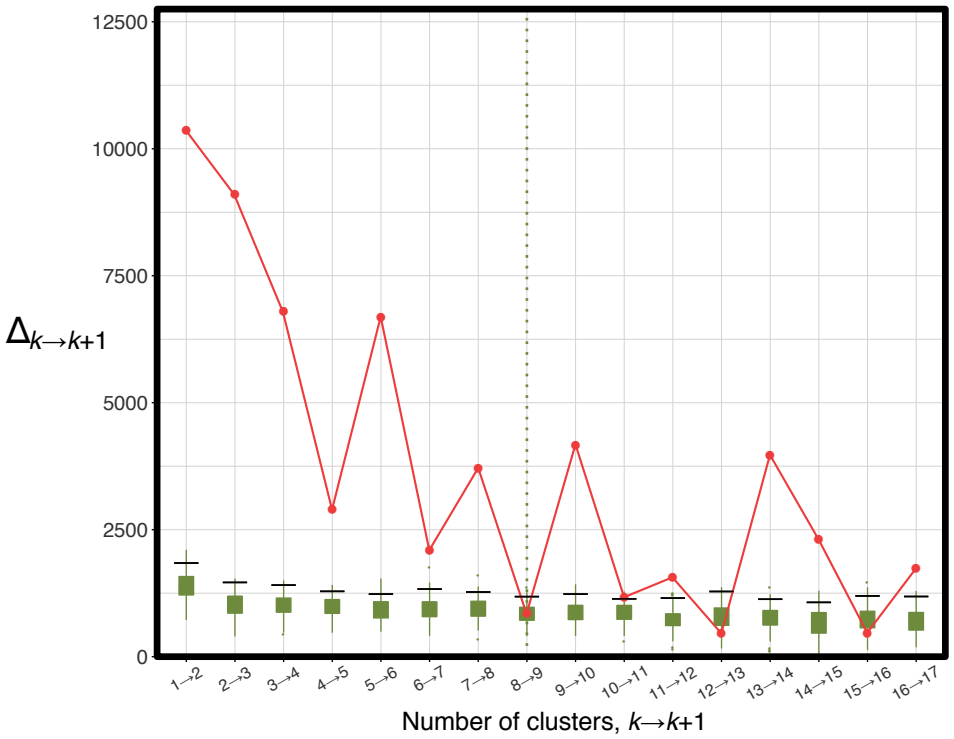


Figure 3.8. *Likelihood improvement gained when partitioning the* Chiastocheta *data into increasing numbers of clusters (red points). Resampled distributions (green boxplots) were generated using the non-parametric permutation procedure with 100 replicates. The number of clusters selected by the stopping criterion, 8, is indicated by the vertical dotted line. For 2–8 clusters the improvement is statistically significant; increasing to 9 clusters is not.*

Using the parametric bootstrap variant of the stopping criterion figure 3.9, I obtain an estimate of 6 for the number of clusters. However, the results show an unexpectedly large amount of variance in the distribution of Δ_k among the simulated pseudoreplicates, far in excess of that observed for the non-parametric permutation variant. This high degree of variance seems to be a feature of large cluster numbers, as the distributions for k=1–4 are narrow. What seems to be behind this is that in general the locus alignments in the original data are phylogenetically quite uninformative, individually (they have on average 12 informative sites per alignment, which is usually insufficient to resolve >100 taxa). When partitioning into large numbers of clusters, the locus trees are built from small subsets of the original data. This makes the cluster trees difficult to estimate correctly. Parametric bootstrapping uses these trees when simulating resampled datasets, and so errors are propagated. When running treeCl on the resampled data it is difficult to infer back the partition that was used to generate the data. Consequently, the cluster trees for the resampled datasets are poorly estimated. If the optimisation algorithms used to infer cluster trees get trapped in local maxima and terminate prematurely — before the maximum likelihood estimate is obtained — then this renders the partition likelihood suboptimal. As the stopping criterion depends entirely on the partition likelihood, these knock-on effects result in the problematic case shown here. However, the performance of the stopping criterion is not ideal for either of the variants, as the values of Δ_k obtained from the analysis of the original data are not smooth. Therefore, for this dataset, I have used the stopping criterion as a guide to choosing the number of clusters, rather than an absolute estimate. This leads me to be conservative about claiming clusters which may or may not be real, so I proceed by choosing what seem to be the best-justified ones, which is four.

As the yeast example showed, even when the species tree is uncontroversial, the individual gene trees can show a large degree of incongruence. The 176 *Chiastocheta* loci show 176 distinct topologies. That we can reduce these to four groups simplifies our interpretation considerably.

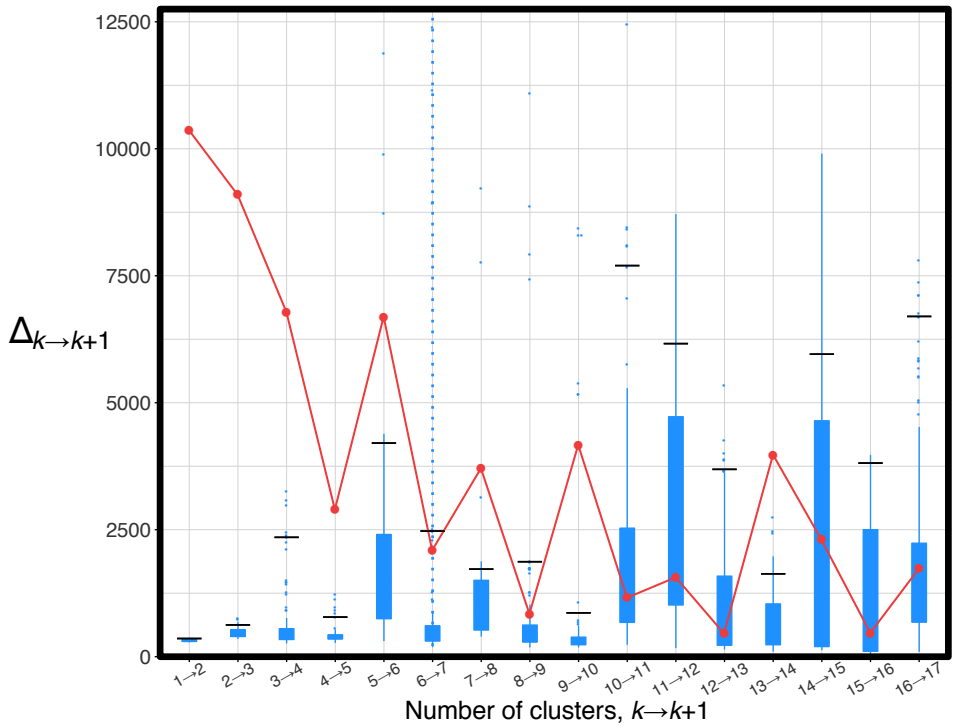


Figure 3.9. *Likelihood improvement gained when partitioning the* Chiastocheta *data into increasing numbers of clusters (red points). Resampled distributions (blue boxplots) were generated using the parametric bootstrap procedure with 100 replicates. The number of clusters selected by the stopping criterion, 6, is indicated by the vertical dotted line. For 2–6 clusters the improvement is statistically significant; increasing to 7 clusters is not.*

The trees inferred for the four clusters differ substantially, both in topology and branch lengths (figure 3.10). In particular, many of the deep relationships are well-resolved but different across clusters, suggesting genuine differences in the history of the loci. However, with few exceptions, samples from each species form distinct monophyletic groups. This is consistent with well-documented differences in genital morphology across most of these species (Després et al. 2002). With more data available, phylogeny and morphology now agree. Furthermore, the even cluster size distribution (cluster sizes of 29, 58, 42 and 47 loci) suggests that the method is not simply finding clusters that consist of one or two outliers.

The cluster trees show variability in tree length. Overall, the tree lengths are quite short, suggesting that there is little evolutionary distance between the species. However, as can be seen in figure 3.10, the tree inferred for

cluster 2 is noticeably shorter than the three other trees. What this may suggest is that clustering using geodesic distances can split loci according to the underlying rate of evolution, as well as according to the topology of the locus trees. Further work is required to see whether this is beneficial, or is something that obscures evolutionary interpretation.

The greatest departure from monophyly is shown in the cluster consisting of 29 loci (figure 3.10, first cluster). In this cluster the majority of the representatives of species *C. lophota* are found at the base of a clade that also contains *C. macropyga*, *C. trollii*, *C. setifera* and *C. inermella*. For partitions into greater numbers of clusters than four, we observe at least one tree in which species monophyly is largely absent (for an example obtained from a partition of five clusters, see figure 3.11). There is no plausible biological explanation for these trees, which may indicate that likelihood improvements gained when partitioning into more than four clusters are due to fitting to the noise in the data, extracting loci with weak or conflicting signal.

Overall, the picture that emerges from the analysis confirms the existence of seven distinct species in the *Chiastocheta* genus, but implies that the branching order among them varies substantially across loci. Such variation is suggestive of incomplete lineage sorting, particularly as six of the seven species (all except *C. rotundiventris*) are thought to have radiated more or less synchronously (Espíndola, Buerki, and Alvarez 2012). To test this hypothesis, future work could assess the fit of this data to a mechanistic model of ILS.

3.2.5 SUMMARY

The results above demonstrate that the treeCl clustering method is applicable to large scale empirical datasets. They also confirm that treeCl is applicable to datasets with incomplete "occupancy" among species, as was the case in the sparse *Chiastocheta* dataset. This dataset was produced by RAD sequencing, a technique that is prone to having a large proportion of missing data (Chattopadhyay, Garg, and Ramakrishnan 2014). As a result, the dataset analysed here is sampled haphazardly. More targeted sequencing of the *Chiastocheta* flies could yield greater understanding of the evolution of this widespread group.

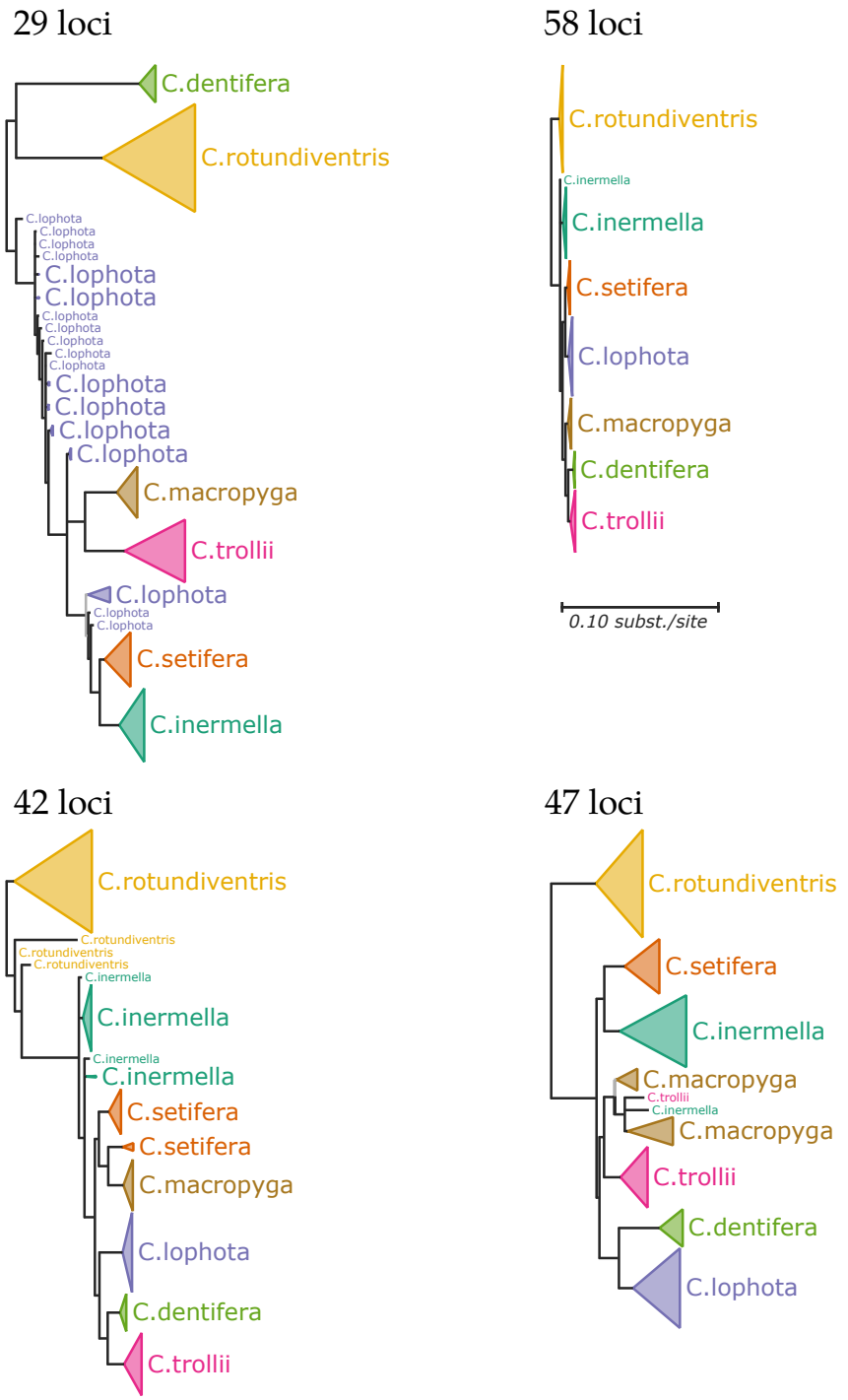


Figure 3.10. *Trees obtained when partitioning RAD sequencing data from globeflower flies of the genus* Chiastocheta *into 4 clusters. The trees are drawn to scale, and are rooted at their midpoint, as the outgroup is unknown. Leaves are coloured according to species membership. Branch support is indicated as follows: branches with support values below 0.9 are collapsed into multifurcations; those with support in the range 0.9–0.95 are coloured grey; those with support >0.95 are coloured black. Support values are calculated using approximate Bayes (Anisimova et al. 2011).*

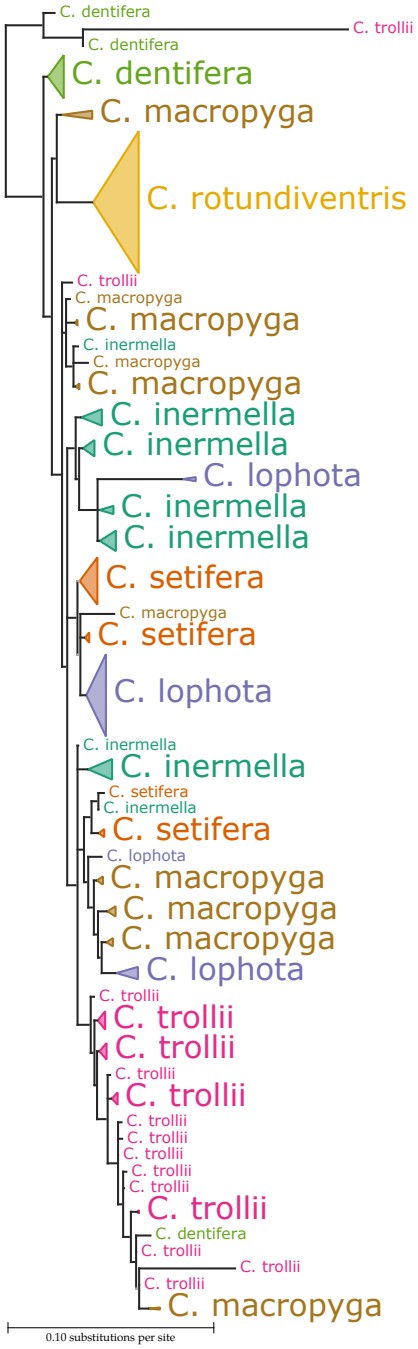


Figure 3.11. *A cluster tree obtained from partitioning the* Chiastocheta *dataset into five clusters. Here, with the exception of the outgroup species,* C. rotundiventris*, the individual species are not grouped into monophyletic clades. This contrasts with the remaining four clusters (not shown, as the trees were largely the same as those obtained when partitioning into four clusters; figure 3.10), which resulted in four trees in which the species did form monophyletic groups.*

The simulations presented (section 2.4) suggested that as long as the clusters are separated by a few topological moves, occupancy as low as 40% incurs negligible performance degradation (figure 2.15). In the *Chiastocheta* study, although low occupancy may be responsible for the high variance in Δ_k test statistic distribution generated by parametric bootstrapping, there is no other sign that low occupancy has caused other problems. However, while the new stopping criterion introduced in this study is able to cope with sparse data matrices, it seems that there is room for further improvement to the parametric bootstrap procedure to increase its robustness to estimation errors caused by weak phylogenetic signal and missing data.

3.3 DISCUSSION

I applied the treeCl method to two empirical datasets, one from yeast and one from *Chiastocheta* flies. The examinations of large datasets presented in this chapter demonstrate that the treeCl method is applicable to real world datasets, as well those derived from simulations, such as were considered in Chapter 2. Both the yeast and *Chiastocheta* datasets contain sequence alignments numbered in the hundreds, placing them on a similar scale to data that is routinely produced by high-throughput sequencing approaches such as RAD sequencing.

Both datasets show a high degree of phylogenetic incongruence, although this is likely to be for different reasons: for the yeast dataset, the principle discovery was that misannotated orthology caused the greatest amount of incongruence, with little incongruence being witnessed for those loci for which the orthology was accurately established. For the *Chiastocheta*, ILS is likely to be the leading cause of incongruence. Due to the process-agnostic nature of treeCl, I was able to apply the method in the same way to both datasets, and learn something about the incongruent signals in the data. This allows me to speculate as to the likely processes at play, and prioritise different types of follow-up analysis—stringent orthology identification in the first case, and analysis under a mechanistic ILS model in the second. In this way a process-agnostic approach is complementary to, rather than in opposition to, the use of mechanistic models of incongruence.

The unexpected behaviour of the parametric bootstrap variant of the stopping criterion suggests that care needs to be taken when applying treeCl to datasets with weak phylogenetic signal, as early errors in parameter and tree estimation seem to be propagated into the resampling procedure. This causes the resampled distributions of Δ_k to have a much larger variance than those obtained through the non-parametric permutation variant of the stopping criterion, which does not depend on the inferred parameters from the analysis of the original data, and seems to be more robust to errors made in the initial analysis (figure 3.8).

Although the results of these empirical studies are positive, and show treeCl to be effective when applied to real data, they also suggest that there may be some weaknesses of the stopping criterion that would benefit from further testing. For example, it would be beneficial to be able to assess how well estimated the partition likelihood is. In the simulations (see figure 2.10), and in the yeast analysis (see figure 3.1) the values of Δ_k estimated from the original data (red lines) are smoothly decreasing, whereas in the *Chiastocheta* analysis the values are jagged, with no smoothly decreasing trend. One would expect partition likelihoods calculated from more complex models with greater numbers of clusters to obtain higher likelihoods than simpler models, and consequently that Δ_k — the difference between these log-likelihoods — to be positive. In fact that is what we observe in both these empirical studies. However, one would also expect that there would be diminishing returns: these positive values would tend to get smaller as the compared models become more complex, so that adding a single extra cluster exerts less influence. This is not observed in the *Chiastocheta* study, as Δ_k shows oscillating behaviour up to large values of k. This may indicate that the partition likelihood is not being fully optimised in some cases. Optimisation algorithms can become trapped in local maxima, and fail to find the global maximum, as is well known (Hordijk and Gascuel 2005; Guindon et al. 2010). This may be happening in the maximum-likelihood inference of cluster trees that forms the basis of the partition likelihood. One common way to mitigate against this in many optimisation

contexts, including phylogenetic tree estimation, is to run optimisations several times from different initial parameter values (Chor et al. 2000). In this clustering context, both the phylogenetic tree inference and the cluster assignments could be run multiple times to increase the chances of finding a maximised value of the partition likelihood. It would be interesting to see if this could be used to diagnose whether the partition likelihood is well estimated.

Related to the partition likelihood, the results of applying the parametric bootstrap resampling procedure (figure 3.9) show that difficulties in optimising cluster trees may be propagated into simulated datasets derived from the inferred cluster trees. This could perhaps be explored by running targeted simulation experiments to better understand the factors that influence the resampled distribution of Δ_k. This would aid in determining whether there is a fundamental problem with the stopping criterion—which seems unlikely, as it has worked well in both simulation and in the analysis of the yeast dataset—or if the difficulties are only observed when analysing datasets with a particular set of features, such as sparse datasets with little phylogenetic variation between taxa, such as was observed for the *Chiastocheta* dataset.

The clustering approach I have described is a "greedy" approach, in the context of finding an optimal maximum likelihood clustering, in that it does not optimise according to the maximum likelihood criterion. An alternative approach to tackling the effect of cluster assignment on optimising the partition likelihood is to include cluster assignment as an optimisable parameter within the cluster tree inference step. This approach will be discussed in Chapter 4.

4

FURTHER DEVELOPMENTS: VISUALISATION AND OPTIMISATION

This chapter addresses the topic of uncertainty: the uncertainty in tree inference, and the uncertainty in cluster assignment. In the first section I discuss how uncertainty in the initial tree inference can be incorporated into the treeCl procedure by including bootstrap replicate trees, and how to do so without significantly increasing the overall runtime. I will discuss how to include bootstrap trees in visualisation of tree-space, and how this can be used to improve cluster assignment. The second section focuses on the uncertainty in the inferred partition, and describes a likelihood-based approach to optimising the assignment of loci into clusters.

4.1 UNCERTAINTY IN TREE ESTIMATION

It may be helpful to recap the treeCl process for clustering loci as it was introduced in section 2.1. In the first step, trees are inferred from the sequence alignments of each locus, and in step two the inter-tree distances are used as the basis for clustering. What was omitted concerning the trees estimated in the initial step was any consideration of the uncertainty in the estimate, particularly the uncertainty concerning the branching pattern of the tree. This can be difficult to resolve, particularly if the data are sparse, uninformative about a region of the tree, or contains sites that are in conflict over the resolution of a branch. The statistical support of each branch of the tree can be estimated: many measures have been proposed to statistically estimate the degree of topological uncertainty in phylogenetic tree estimation (Felsenstein 1985; Anisimova and Gascuel 2006; Stamatakis, Hoover, and Rougemont 2008; Anisimova et al. 2011; Minh, Nguyen, and Haeseler 2013). Failure to account for the uncertainty in tree estimation potentially leads

to an elevated level of confidence when making use of the trees in downstream analysis. In the context of my phylogenetic clustering approach, this may lead to making over-eager assignments to clusters that the data do not support, so a means of assessing tree distances while respecting uncertainty could be desirable.

One of the most widely used methods is the bootstrap (Efron 1981). In the phylogenetic bootstrap, several replicate alignments — with the same number of columns as the original alignment — are generated by randomly sampling, with replacement, the original alignment's columns (Felsenstein 1985, 1988). After inferring trees from the replicate alignments, the frequency with which each split is observed among the set of bootstrap trees is used as a support measure for each split present in the tree inferred from the original alignment. This is a measure of how strongly the data support an inference under a given estimation method, but is not a measure of the probability of 'correctness' because it involves no model of how data are created (it is non-parametric).

A straightforward incorporation of uncertainty into treeCl is to include bootstrap replicate trees when calculating inter-tree distances, resulting in a much larger distance matrix. This can then be embedded in a space (e.g. by classical multidimensional scaling (CMDS)), as before, and clusters can be assigned from this embedding by application of a technique such as k-means to the embedded points (note that MDS is not a clustering technique in itself). The clustering step could then incorporate the extra information provided by the bootstrap distances, perhaps through use of constraint-based clustering, in which trees estimated from the original data are constrained to belong to the same cluster as their bootstrap replicates' trees.

However, as the number of comparisons required to generate the distance matrix grows proportionally to the square of the number of trees, and the inclusion of bootstrap replicates swells the dataset size by a factor that is typically in the region of 100 (see Pattengale et al. 2010 for a discussion of

how many bootstraps replicates are appropriate), this is liable to become too computationally burdensome to be practical.

In order to circumvent these extra computations, I have devised approximate methods for generating multidimensional scaling embeddings of bootstrap replicate trees into a space already defined by the relationships between the trees defined by the data. Under these approximations, each bootstrap tree can be embedded after calculating its distance from each of the n locus trees, avoiding the need to calculate any distances among bootstrap trees. I show the application of these methods to a small sample of simulated data, to demonstrate that the methods can work.

A side effect of preparing tree distance data for clustering by MDS and k-means is that the trees are embedded in a Cartesian space. By placing the embedding in two or three dimensions, these embeddings can be easily visualised, which can be an aid to the interpretation of the phylogenetic similarities between loci. Having a scatterplot of the positions of trees relative to each other in a space that makes intuitive sense allows us to get a feeling for whether the trees form distinct clusters, or if there are outliers. Incorporating bootstrap trees into the embedding also allows the uncertainty in the estimates to be visualised.

4.1.1 THE CLASSICAL MULTIDIMENSIONAL SCALING ALGORITHM

I will first summarise the classical multidimensional scaling algorithm, and then move on to describe my proposed approximations.

Classical multidimensional scaling (CMDS) provides an analytical solution to finding the embedding in reduced dimensions of a set of points while best preserving the distances between them (Torgerson 1952). Provided the observed distances are Euclidean then this is the optimal solution within the given number of dimensions (call this d), with distortion being introduced otherwise. CMDS provides a transformation that maps an $n \times n$ distance matrix, M, of the pairwise distances among n data points, to a $n \times d$ matrix, $\boldsymbol{X}$, of the coordinates of n points in d dimensions.

The procedure happens in four steps:

1. Square the input distances ($\boldsymbol{M}$) to give the matrix $\boldsymbol{B}$:

$$b_{ij} = m_{ij}^2 \tag{4.1}$$

2. Translate this matrix to put its centroid at the origin, yielding the double-centred matrix $\boldsymbol{B}^*$, by subtracting the column and row means of $\boldsymbol{B}$, adding the overall "grand" mean, and dividing by negative two:

$$b_{ij}^* = -\frac{1}{2}\left(b_{ij} - \frac{1}{n}\sum_{k=1}^{n} b_{kj} - \frac{1}{n}\sum_{l=1}^{n} b_{il} + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} b_{kl}\right) \tag{4.2}$$

3. Decompose the matrix $\boldsymbol{B}^*$ into its eigenvectors and eigenvalues. Eigenvalues are placed on the main diagonal of $\boldsymbol{\Lambda}$, which is zero elsewhere, and the eigenvectors are arranged as the columns of matrix $\boldsymbol{U}$. Eigenvalues and eigenvector columns are arranged in descending order of eigenvalue size:

$$\boldsymbol{B}^* = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1} \tag{4.3}$$

4. $\boldsymbol{X}$, the embedded coordinates of n points in n dimensions, are calculated as the product of $\boldsymbol{U}$ and $\boldsymbol{\Lambda}^{\frac{1}{2}}$. The Euclidean distances among the n points of $\boldsymbol{X}$ are the closest Euclidean approximation to the input distances, $\boldsymbol{M}$ (if the distances in $\boldsymbol{M}$ are Euclidean, then the distances of $\boldsymbol{X}$ exactly reproduce $\boldsymbol{M}$):

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}} \tag{4.4}$$

5. Finally, dimension reduction into d dimensions is achieved by truncating the matrices $\boldsymbol{\Lambda}^{\frac{1}{2}}$ and $\boldsymbol{U}$ to the d largest eigenvalues and eigenvectors. Truncated matrices are notated as $\boldsymbol{X}_{(d)}$, $\boldsymbol{\Lambda}^{\frac{1}{2}}_{(d)}$ and $\boldsymbol{U}_{(d)}$. The subscripts signify the dimension of each matrix:

$$\underset{n\times d}{\boldsymbol{X}_{(d)}} = \underset{n\times d}{\boldsymbol{U}_{(d)}}\ \underset{d\times d}{\boldsymbol{\Lambda}^{\frac{1}{2}}_{(d)}} \tag{4.5}$$

What this means, in brief, is that the input distance matrix is squared, its row, column and overall means are calculated in order to double-centre it,

and embedded coordinates are recovered from the resulting matrix's eigenvectors and the square roots of its eigenvalues. As $\boldsymbol{B}^*$ is symmetric and real, its eigenvalues will be real, although they are not guaranteed to be positive. Eigenvalues beyond the dimensionality of the data that generated the (Euclidean) distances will be zero. Non-Euclidean distances can lead to there being negative eigenvalues. When the number of dimensions in the embedding is much smaller than the dimensionality of the data, zero or negative eigenvalues are unlikely to be encountered.

An alternative expression for the coordinate matrix

An alternative expression for $\boldsymbol{X}$ is given in (4.6), which is derived easily from equations (4.3) and (4.4). In this formulation $\boldsymbol{X}$ is dependent on $\boldsymbol{B}^*$ as well as $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$, so looks more complicated than (4.4). However, inclusion of $\boldsymbol{B}^*$ into (4.6) will be useful in the approximations described in the next section. Dimension reduction into d dimensions is achieved again by truncating the eigenvalue and eigenvector matrices to the d largest eigenvalues (4.7).

$$\boldsymbol{X} = \boldsymbol{B}^* \boldsymbol{U} \boldsymbol{\Lambda}^{-\frac{1}{2}} \tag{4.6}$$

$$\underset{n\times d}{\boldsymbol{X}_{(d)}} = \underset{n\times n}{\boldsymbol{B}^*}\ \underset{n\times d}{\boldsymbol{U}_{(d)}}\ \underset{d\times d}{\boldsymbol{\Lambda}_{(d)}^{-\frac{1}{2}}} \tag{4.7}$$

The term $\boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}$ can be interpreted as a transformation (itself derived from $\boldsymbol{B}^*$) which is applied to $\boldsymbol{B}^*$ in order to get embedded coordinates $\boldsymbol{X}$.

4.1.2 APPROXIMATIONS TO CLASSICAL MULTIDIMENSIONAL SCALING, FROM A SUBSET OF THE DATA

As outlined above, one can perform some simple linear algebra on a complete set of distances to determine its embedding in coordinate space. However, obtaining the complete set of distances, including the distances between all the bootstrap replicate trees, requires a large number of inter-tree distance calculations. What I am looking for is a method to approximate the embedding of the locus trees and their bootstrap replicates, that avoids explicitly calculating all pairwise distances between locus and boot-

strap trees. My proposal is to instead calculate the distances for a subset of the trees, and embed these using CMDS as described above; then, by reference to this subset, insert the remainder of the trees into the same embedded space. Concretely, the subset of distances used to construct the initial embedding would be those between the locus trees, with the bootstrap trees being inserted into the space defined on the locus trees, instead of themselves contributing to the creation of the coordinate space.

To see how this can be done, consider the CMDS as a procedure that maps each row of $\boldsymbol{B}^*$ to a coordinate point. If a bootstrap tree can be converted to row information that takes the same form as the rows of $\boldsymbol{B}^*$, it should be possible to use (4.7) to map this row into the embedded space. This row information can be obtained from the bootstrap tree's distance from each of the embedded locus trees, by following a similar procedure to that outlined in section 4.1.1. As this bootstrap tree comes from outside of the sample of trees used to construct the embedding, its mapped coordinates are just an approximation to where it would be mapped to if it had been included (in fact, all the coordinates would have been shifted somewhat, if the CMDS embedding had been calculated from a distance matrix that included the bootstrap tree).

To insert a single bootstrap tree into the embedding, I calculate its distance from each of the locus trees. As the bootstrap tree was derived from one of the locus trees, I replace the row of $\boldsymbol{B}$ that corresponds to that locus tree with the squares of the distances obtained from the bootstrap tree, to produce matrix $\widehat{\boldsymbol{B}}$. Under the assumption that these distances are reasonably close to the distances they are replacing the approximation should be fairly good. As the bootstrap tree was derived from the alignment that gave the locus tree, this seems reasonable.

This new row of the matrix $\widehat{\boldsymbol{B}}$ is double-centred using either the means derived from $\boldsymbol{B}$ (method variant 1), or means that have been updated to reflect the new values inserted in $\widehat{\boldsymbol{B}}$ (method variant 2)—this results in the approximate double-centred matrix $\widehat{\boldsymbol{B}^*}$. The new coordinate point is obtained by calculating $\widehat{\boldsymbol{B}^*}\boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}$ (4.7). The eigenvectors $\boldsymbol{U}$ and eigenvalues $\boldsymbol{\Lambda}$ are those

derived from $\boldsymbol{B}^*$, and are not recalculated when embedding a bootstrap tree. This process is then repeated for every bootstrap tree. Ultimately, this gives a set of coordinates for all locus trees and bootstrap trees.

4.1.3 A WORKED EXAMPLE

Here is an example to clarify the process I have just described. Suppose there are four locus trees, and I wish to find their embedding in two dimensions. The distances between the locus trees can be calculated in tree-space using one of the metrics discussed in section 2.2. For this example, suppose the calculated distances are those given in the matrix $\boldsymbol{M}$, in which m_{ij} is the distance between tree i and tree j:

$$\boldsymbol{M} = \begin{bmatrix} 0 & 5 & 7 & 6 \\ 5 & 0 & 8 & 9 \\ 7 & 8 & 0 & 4 \\ 6 & 9 & 4 & 0 \end{bmatrix}$$

$\boldsymbol{B}$ is obtained by squaring $\boldsymbol{M}$ element-wise, i.e. $m_{ij} = b_{ij}^2$:

$$\boldsymbol{B} = \begin{bmatrix} 0 & 25 & 49 & 36 \\ 25 & 0 & 64 & 81 \\ 49 & 64 & 0 & 16 \\ 36 & 81 & 16 & 0 \end{bmatrix}$$

We calculate matrices of column and row means to calculate $\boldsymbol{B}^*$ (see equation 4.2). The column-wise means are 27.5, 42.5, 32.25 and 33.25 (the row-wise means are the same, by symmetry). The "grand" mean is 33.875. Arranging these as the matrices $\boldsymbol{\mu}_{col}$, $\boldsymbol{\mu}_{row}$ and $\boldsymbol{\mu}_{grand}$ allows us to calculate $\boldsymbol{B}^* = -\frac{1}{2}(\boldsymbol{B} - \boldsymbol{\mu}_{row} - \boldsymbol{\mu}_{col} + \boldsymbol{\mu}_{grand})$:

$$\boldsymbol{\mu}_{col} = \begin{bmatrix} 27.5 & 42.5 & 32.25 & 33.25 \\ 27.5 & 42.5 & 32.25 & 33.25 \\ 27.5 & 42.5 & 32.25 & 33.25 \\ 27.5 & 42.5 & 32.25 & 33.25 \end{bmatrix}$$

$$\boldsymbol{\mu}_{row} = \begin{bmatrix} 27.5 & 27.5 & 27.5 & 27.5 \\ 42.5 & 42.5 & 42.5 & 42.5 \\ 32.25 & 32.25 & 32.25 & 32.25 \\ 33.25 & 33.25 & 33.25 & 33.25 \end{bmatrix}$$

$$\boldsymbol{\mu}_{grand} = \begin{bmatrix} 33.875 & 33.875 & 33.875 & 33.875 \\ 33.875 & 33.875 & 33.875 & 33.875 \\ 33.875 & 33.875 & 33.875 & 33.875 \\ 33.875 & 33.875 & 33.875 & 33.875 \end{bmatrix}$$

$$\boldsymbol{B}^* = \begin{bmatrix} 10.563 & 5.563 & -11.563 & -4.563 \\ 5.563 & 25.563 & -11.563 & -19.563 \\ -11.563 & -11.563 & 15.313 & 7.813 \\ -4.563 & -19.563 & 7.813 & 16.313 \end{bmatrix}$$

The square roots of the two largest eigenvalues ($\boldsymbol{\Lambda}_{(2)}^{\frac{1}{2}}$) and corresponding eigenvectors ($\boldsymbol{U}_{(2)}$) of $\boldsymbol{B}^*$ are used to obtain the embedded coordinates in 2 dimensions, $\boldsymbol{X}_{(d)}$ (4.1.3):

$$\boldsymbol{\Lambda}_{(2)}^{\frac{1}{2}} = \begin{bmatrix} 7.10 & 0 \\ 0 & 3.97 \end{bmatrix}$$

$$\boldsymbol{U}_{(2)} = \begin{bmatrix} -0.280 & 0.626 \\ -0.677 & -0.373 \\ 0.432 & -0.594 \\ 0.525 & 0.342 \end{bmatrix}$$

$$\boldsymbol{X}_{(2)} = \begin{bmatrix} -1.99 & 2.48 \\ -4.81 & -1.5 \\ 3.07 & -2.36 \\ 3.73 & 1.36 \end{bmatrix}$$

As a final check, we can calculate the Euclidean distances between each pair of embedded coordinates and compare them to the tree-space distances we

started with. These 'induced' distances are given in $\boldsymbol{M}_{induced}$. The values in $\boldsymbol{M}_{induced}$ are close to $\boldsymbol{M}$.

$$\boldsymbol{M}_{induced} = \begin{bmatrix} 0.0 & 4.9 & 7.0 & 5.8 \\ 4.9 & 0.0 & 7.9 & 9.0 \\ 7.0 & 7.9 & 0.0 & 3.8 \\ 5.8 & 9.0 & 3.8 & 0.0 \end{bmatrix}$$

Adding a bootstrap tree

Now suppose we have a bootstrap tree, derived from the first locus tree, to add to this embedding. First we calculate its distance from the four locus trees. Let us suppose for this example that these distances are 1 unit away from the first locus tree, and 6 units from the other three locus trees. Then we square these distances. The squared distances can be mapped to the double-centred space using the previously established values of $\boldsymbol{\mu}_{row}$, $\boldsymbol{\mu}_{col}$ and $\boldsymbol{\mu}_{grand}$ (variant 1). Alternatively, these μs could be updated to reflect the new distances, by replacing the first row and column of $\boldsymbol{B}$ with the squared distances, and recalculating the means (variant 2). The first row is replaced because the bootstrap tree was derived from the first locus.

The new row of $\widehat{\boldsymbol{B}^*}$ is calculated as:

$$\begin{aligned} & -\tfrac{1}{2} \begin{bmatrix} 1^2 & - & 27.5 & - & 27.5 & + & 33.875 \\ 6^2 & - & 42.5 & - & 27.5 & + & 33.875 \\ 6^2 & - & 32.25 & - & 27.5 & + & 33.875 \\ 6^2 & - & 33.25 & - & 27.5 & + & 33.875 \end{bmatrix}^T \\ = & \begin{bmatrix} 10.063 & 0.063 & -5.063 & -4.563 \end{bmatrix} \end{aligned}$$

These values are inserted as a new row of $\boldsymbol{B}^*$, replacing the old first row:

$$\widehat{\boldsymbol{B}^*} = \begin{bmatrix} 10.063 & 0.063 & -5.063 & -4.563 \\ 5.563 & 25.563 & -11.563 & -19.563 \\ -11.563 & -11.563 & 15.313 & 7.813 \\ -4.563 & -19.563 & 7.813 & 16.313 \end{bmatrix}$$

The new set of coordinates including the bootstrap tree is obtained by calculating $\widehat{\boldsymbol{B}^*}\boldsymbol{U}_{(2)}\boldsymbol{\Lambda}_{(2)}^{-\frac{1}{2}}$. As we are only interested in the coordinates for the bootstrap tree, i.e. the first row of $\widehat{\boldsymbol{B}^*}\boldsymbol{U}_{(2)}\boldsymbol{\Lambda}_{(2)}^{-\frac{1}{2}}$, this can just be applied to the new row of $\widehat{\boldsymbol{B}^*}$:

$$\begin{aligned}\boldsymbol{X}_{(2)\ 1,*} &= \begin{bmatrix}10.063 & 0.063 & -5.063 & -4.563\end{bmatrix}\begin{bmatrix}-0.280 & 0.626\\ -0.677 & -0.373\\ 0.432 & -0.594\\ 0.525 & 0.342\end{bmatrix}\begin{bmatrix}1/7.10 & 0\\ 0 & 1/3.97\end{bmatrix}\\ &= \begin{bmatrix}-1.049 & 1.946\end{bmatrix}\end{aligned}$$

As a check, we can again inspect the induced distance of this newly embedded bootstrap tree from the four embedded locus trees by calculating Euclidean distances between them within the embedded space — the values are given in $\boldsymbol{M}_{induced}$. These are close to the values 1, 6, 6, 6 obtained from tree-space:

$$\boldsymbol{M}_{induced} = \begin{bmatrix}1.1 & 5.1 & 6.0 & 4.8\end{bmatrix}$$

4.1.4 AN ITERATIVE METHOD FOR FITTING BOOTSTRAP TREES

As an alternative to the approximation to CMDS described in the last section, we can look at adding bootstrap trees to best reflect their distances from locus trees as an optimisation problem. To reiterate, we begin with an embedding of a set of reference trees (the locus trees). It should be noted that now the problem has been recast as an optimisation, this embedding need not be made using CMDS, but could be generated using any multidimensional scaling approach, such as nonmetric multidimensional scaling (Kruskal 1964).

To fit a bootstrap tree into the embedding we need to select coordinate values within the embedded space such that the induced distances within the embedding match as closely as possible to the calculated distance in tree-space between the bootstrap tree and each of the reference locus trees. If $\boldsymbol{x}$

is the vector of coordinates of the bootstrap tree being fitted (one entry per dimension), $\boldsymbol{a}$ is the matrix of the coordinates of the n reference locus trees (one row in a per tree, one column per dimension), and $\boldsymbol{c}$ is the vector of n tree-space distances between the fitted point and each reference point, then (4.8) is a system of n equations that give the discrepancy between the within-embedding Euclidean distances between $\boldsymbol{x}$ and embedded loci $\boldsymbol{a}$, and the distance between the bootstrap tree and the loci in tree-space:

$$r_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{a}_i\| - \boldsymbol{c}_i \tag{4.8}$$

It is unlikely that there exists a value of x for which all discrepancies are zero. Instead, we search for the value of $\boldsymbol{x}$ that minimises the sum of squared discrepancies between the induced distances within the embedding, and the distances calculated in tree-space:

$$f(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}) = \sum_i r_i(\boldsymbol{x})^2 \tag{4.9}$$

4.1.5 SOLVING THE SYSTEM

For overdetermined systems of linear equations the minimising value can be found analytically, but when the equations are non-linear the minimising value generally has to be found iteratively: one selects a starting value, and then improves upon it, step by step, according to an objective function (4.9) that decreases with the discrepancy (Nocedal and Wright 2006).

As the equations in the non-linear system are easily differentiable one can use standard derivative-aware optimisation procedures such as Newton-Raphson optimisation, which uses the first derivative and the Hessian matrix of second partial derivatives of the residual equations to converge rapidly to a minimum. The matrix of partial derivatives of the residuals (4.8)—the Jacobian, $\boldsymbol{J}$ — is illustrated in (4.10), again for two dimensions. The gradient is calculated according to 4.11 (Nocedal and Wright 2006:252). In non-linear least squares optimisation it is common to use 4.12 as an ap-

proximation to the true Hessian (Dasgupta 2006:187; Nocedal and Wright 2006:259):

$$\boldsymbol{J}_{ij} = \frac{x_j - a_{ij}}{\|\boldsymbol{x} - \boldsymbol{a}_i\|} \tag{4.10}$$

$$\nabla f(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}) \tag{4.11}$$

$$\boldsymbol{H} \approx \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}) \tag{4.12}$$

The Newton-Raphson method works by proposing an improved minimising value of $x^{(i+1)}$, in terms of the objective function, from the candidate solution $x^{(i)}$. The new x value is produced according to the update procedure in 4.13, which continues over a range of iterations (indexed by i), until convergence is reached. Any reasonable starting point for the iterations can be chosen, e.g. initialise $x^{(0)}$ as the position of the locus tree from which the bootstrap tree was derived.

$$\begin{aligned} \boldsymbol{x}^{(i+1)} &= \boldsymbol{x}^{(i)} - \boldsymbol{H}^{-1}(\boldsymbol{x}^{(i)}) \nabla f(\boldsymbol{x}^{(i)}) \\ &\approx \boldsymbol{x}^{(i)} - \left(\boldsymbol{J}(\boldsymbol{x}^{(i)})^T \boldsymbol{J}(\boldsymbol{x}^{(i)}) \right)^{-1} \nabla f(\boldsymbol{x}^{(i)}) \end{aligned} \tag{4.13}$$

4.1.6 VALIDATION OF APPROXIMATIONS

I produced sets of simulated data to assess how well CMDS and the approximations described above capture the distances between locus and bootstrap trees in reduced dimensions. Using the simulations I tested how well the embeddings can reproduce the distances calculated in tree-space. The variants I am using are the standard CMDS algorithm, the two approximations to CMDS — with and without recalculating the means — and the non-linear least squares fitting procedure. The discrepancy between the induced distances within the embedding and the tree-space distances is quantified using the root-mean-square deviation (RMSD).

Simulation design and methods

I simulated data consisting of five locus alignments per cluster from four clusters, separated by one SPR rearrangement, according to the method in section 2.4.1. Each locus alignment contained 250 characters simulated using the WAG model (Whelan and Goldman 2001), and 20 taxa. I generated 100 bootstrap alignments from each locus alignment. These datasets are somewhat smaller than those in previous chapters, as they only contain 20 loci. I chose to limit the number of locus trees for this experiment because, with the addition of 100 bootstrap replicates per locus, the total number of alignments per dataset rises to 2020. Inferring phylogenetic trees for these alignments, and calculating inter-tree distances is a significant amount of computation.

Following simulation of the data, I estimated trees for all locus and bootstrap alignments using RAxML with the WAG model and four categories of gamma-distributed rates. I calculated the inter-tree distances for all trees, including the bootstrap replicates, using the geodesic distance metric.

Embeddings were made in three dimensions. I embedded the entire set of trees, locus and bootstrap using CMDS. For the approximate embeddings, I first embedded the locus trees using CMDS, and added the bootstrap trees using the approximate CMDS procedure described in section 4.1.2 with and without recalculation of means, and by non-linear least squares. For each of the five resulting embeddings, I calculated the RMSD discrepancy between the induced distances between embedded points and the distances calculated in tree-space using the geodesic metric. For each embedding I noted the number of distance calculations required, and measured the time taken to compute the distances and the embedding. All runs were done on the same Intel Xeon CPU X5650 2.67GHz processor, to make the run times comparable. This procedure was repeated five times, and the average results are reported.

Results and discussion

The results are given in table 4.1, which summarises two quantities for each method: (i) the length of time taken by each method; and (ii) how well the

METHOD	DISTANCES	TIME (S)	RMSD
CMDS	2 039 190	1194.8	0.0520
Approx. CMDS, no recalculation (ACMDS1)	40 190	52.7	0.0523
Approx. CMDS, recalculation (ACMDS2)	40 190	52.9	0.0496
Non-linear least squares (NLLS)	40 190	62.4	0.0431

Table 4.1. *Results of approximate embedding procedures. The methods are: CMDS – distances between all bootstrap and locus trees are fit using CMDS; Approx. CMDS, no recalculation – bootstrap trees are fit to the CMDS embedded locus trees without recalculation of means; Approx. CMDS, recalculation – bootstrap trees are fit to the CMDS embedded locus trees with recalculation of means; Non-linear least squares – bootstrap trees are fit to the CMDS embedded locus trees using linear least squares (refer to text for details of the procedures). RMSD values compare the resulting embedding with the distances measured in tree-space. Values given are the average of five independent runs.*

input distances are recapitulated by the embedding. This second quantity is measured as the RMSD between the Euclidean distances among all the embedded points and the distances computed in tree-space among all locus and bootstrap trees. The values given in the table are the average of quantities computed for five independently simulated datasets.

It is striking how much computation is saved when using approximate methods compared to full CMDS. For datasets of the size considered (20 locus trees each with 100 bootstrap replicates), more than two million inter-tree distances must be calculated to produce the full CMDS embedding of all locus and bootstrap trees. These trees are moderately small, with only 20 taxa, each distance can be computed in milliseconds, but the time taken to produce the CMDS embedding is on the order of 20 minutes. Contrast this with the approximate methods, which each require around forty thousand distance computations, and are all completed in approximately 50–60 seconds. There is little difference in the time taken for each of the approximations to CMDS; non-linear least squares optimisation takes $\sim$20% longer. Surprisingly, the RMSD values indicate that, with the exception of the approximate CMDS procedure that does not recalculate means, the approximate methods provide a better fit to the original tree-space distances than the full CMDS embedding, with non-linear least squares optimisation having the best fit of all.

The approximations all have in common that they fit bootstrap trees independently to an embedding of the locus trees. It can be argued that this would make the approximate embeddings *less* likely to fit well to the tree-space distances, as the approximations do not use information that is available about the between-bootstrap tree-distances. However, as table 4.1 shows, two of the three approximations produce a better fit, as assessed by RMSD. It may be that fitting each bootstrap tree independently allows for a lifting of constraints that are otherwise present when using CMDS on the entire set of trees that are acting detrimentally to the overall quality of the embedding. In addition, because CMDS is prone to distortion when the input distances are not Euclidean (Torgerson 1952), and tree-space is not a Euclidean space (Billera, Holmes, and Vogtmann 2001), it may be that fitting each bootstrap tree independently mitigates against potential distortions arising from the non-Euclidean nature of the input distances.

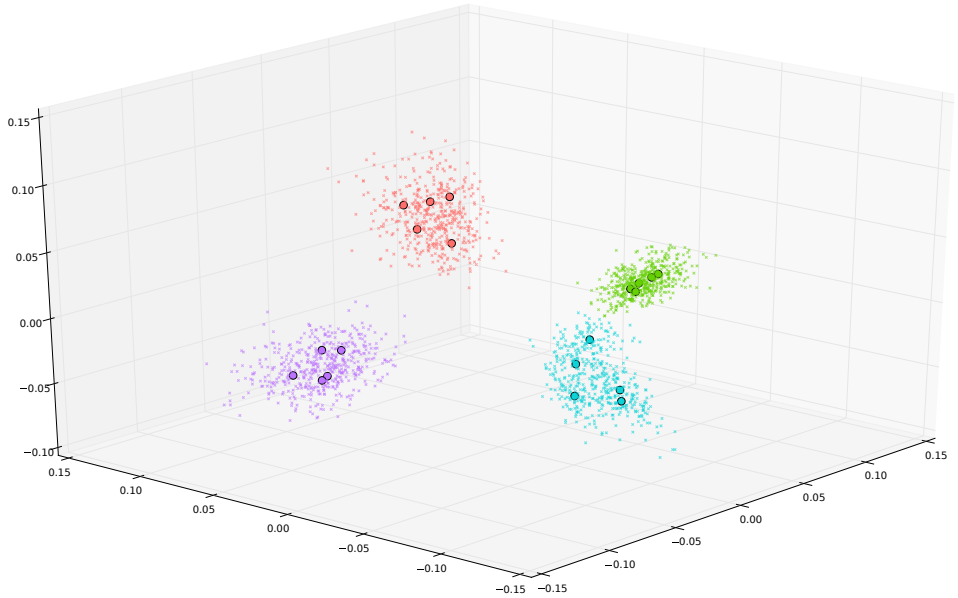


Figure 4.1. *Locus and bootstrap trees embedded in three dimensions using CMDS. The positions of locus trees are indicated by circles, bootstrap trees by small crosses. Trees were simulated from four clusters; cluster membership is colour coded. The space is defined by principal coordinate axes determined by the CMDS procedure. Distances within this space are intended to correspond to distances in tree-space. However, the numbers on the axes have no special meaning.*

The embeddings obtained for the first of the five repetitions are plotted in figures 4.1 to 4.4, with locus trees represented by large circles and bootstrap

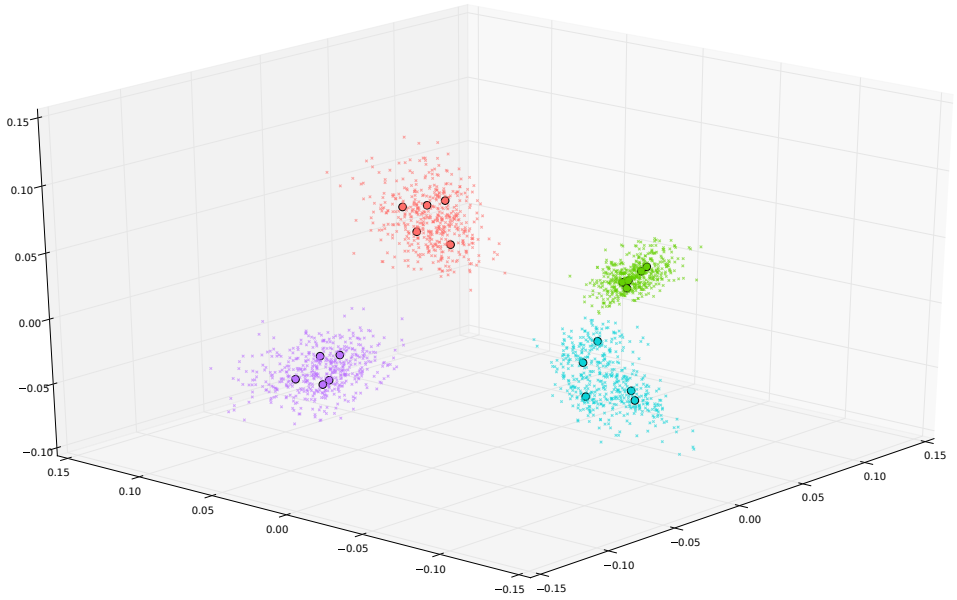


Figure 4.2. *As figure 4.1, except locus trees were fitted to the space using CMDS, and bootstraps added using an approximation based on CMDS, without recalculation of means.*

trees by small crosses. Cluster membership is colour coded as a visual convenience. In all the plots the clusters can lie in separate areas of the embedded space, and are neatly separated. The embedded bootstrap trees form clouds surrounding the embedded locus trees. These clouds occupy a greater volume of the space than the locus trees, providing a visual cue summarising uncertainty in the locus tree estimation. Figure 4.1 shows the result of embedding the entire dataset of locus and bootstrap trees together using CMDS, and figures 4.2 and 4.3 show the two approximate CMDS embeddings, without and with recalculation of means, respectively. It is hard to see any differences among these three embeddings, although the RMSD between the induced distances and the tree-space distances indicate that there some differences do exist. This suggests that the approximate methods are able to substitute for the full CMDS procedure, with little reduction in the quality of the embedding obtained, at least in this toy example.

The embedding produced using non-linear least squares to fit the bootstraps (figure 4.4) shows a clear difference from the other plots. In this case the bootstrap clouds do not align with the locus trees, and do not cover the space surrounding the locus points uniformly. This is likely an artifact of

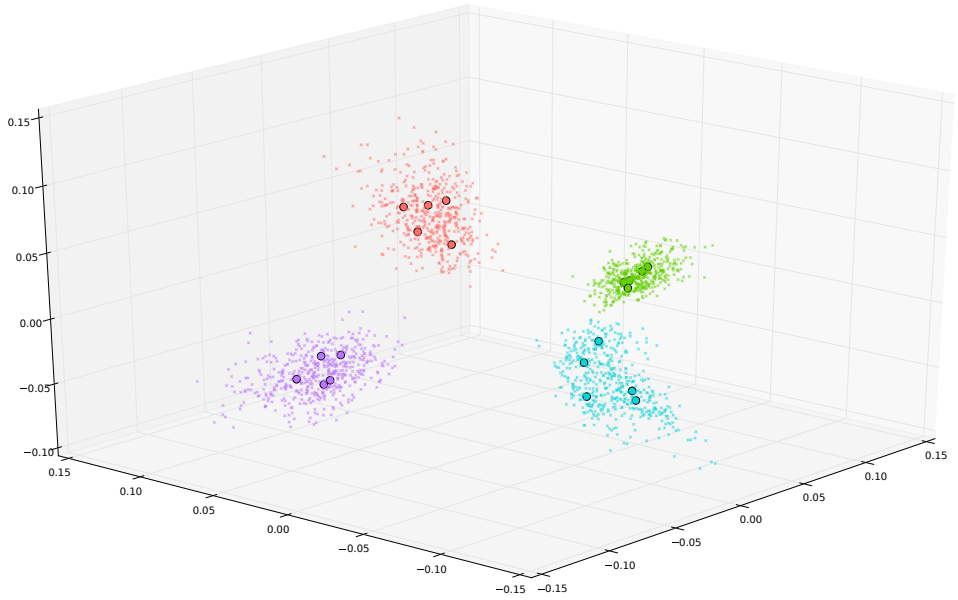


Figure 4.3. *As figure 4.1, except locus trees were fitted to the space using CMDS, and bootstraps added using an approximation based on CMDS, with recalculation of means.*

the method: as the bootstrap trees are introduced to the embedded space independently, the method cannot place the bootstrap trees in a way such that the distances between bootstrap trees are respected. However, this was the embedding that, by RMSD, provided the best fit to tree-space. A possible explanation for this is that distances between bootstrap trees from the same cluster are small, and errors in their placement have little influence on the RMSD.

4.1.7 SUMMARY

I have presented in this section a number of approaches for approximating embedding tree distances in reduced dimensions, based on CMDS. The motivation for developing these approximations is the consideration that the widely used bootstrap procedure increases the number of trees in a dataset by a large factor—typically 100—which makes embedding all the distances by standard CMDS computationally challenging (table 4.1). The approximations require significantly less computation than the standard CMDS approach, and achieve similar or better accuracy than CMDS according to the

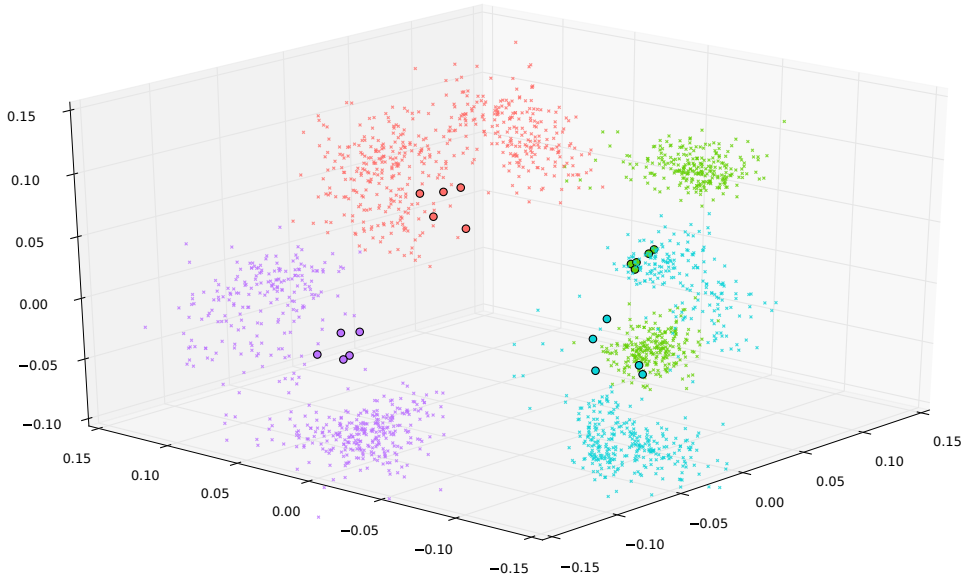


Figure 4.4. *As figure 4.1, except locus trees were fitted to the space using CMDS, and bootstraps added using non-linear least squares optimisation.*

root-mean-square deviation between the distances computed in tree-space and those induced by the embedding.

The embeddings allow the distribution of trees in tree-space to be readily visualised in two or three dimensions, which allows for rapid assessment of a phylogenetic dataset by eye. The bootstrap clouds surrounding the locus trees provide a ready indication of the degree of uncertainty in the locus tree estimation. As the best fitting approximation, as determined by RMSD, non-linear least squares, also looks somewhat distorted in the visualisation, it seems there is further investigation needed before we can satisfactorily represent tree-space in a space that can be understood intuitively. It may be that the distortion derives from the non-Euclidean nature of tree-space, as it is known that CMDS becomes distorted when distances depart from being Euclidean (Torgerson 1952). Other methods, such as nonmetric multidimensional scaling, may be more robust to non-Euclidean inputs, and also compatible with non-linear least squares approximations.

4.2 OPTIMISING A PHYLOGENETIC CLUSTERING

The data clustering technique I have presented so far has been successful on both simulated (Chapter 2) and empirical (Chapter 3) data, but it should

be noted that it suffers from an important drawback when used to generate a data partition. Namely, the clustering depends on inferred quantities (distances between trees) and treats these derived quantities as if they were actual observations. This is a heuristic approach, for although the clustering procedure ostensibly finds the best partition of the input data, it does so under the somewhat arbitrary criteria to which that clustering procedure is sensitive. What I really wish to address is what is the best evolutionary interpretation of a set of (potentially) incongruent data. Even if the clustering algorithm perfectly generates the optimum partition of the distance data (as hopefully it does), there is no guarantee that this also corresponds to the most useful evolutionary interpretation of the underlying biological data. While the results presented in Chapter 3 suggest that a good clustering result may correspond to an evolutionary hypothesis, this is not necessarily the case.

Furthermore, the clustering algorithms presented deliver a static result: they have no way of incorporating evolutionary assessments and updating the partition to give a better fit. The closest that the presented techniques come is that the results of those with some degree of stochasticity will vary: for example, those in which the final assignment into clusters is made by *k*-means may show random variation as typically *k*-means algorithms make multiple inferences from many randomly chosen starting points (see section 2.3.3). I can choose among these whichever gives the highest partition likelihood. I could expand this approach to running several clustering algorithms independently — even using different distance metrics — and choosing the best result, but this undirected approach is likely to be quite inefficient, and complicates statistical interpretation due to multiple testing.

An alternative is to use likelihood-based optimisation protocols to partition the data. The probabilistic framework that has been developed for phylogenetic inference uses models of sequence evolution to describe the evolution of sequences in terms of probabilities (Felsenstein 1973; Yang 2000), and can be used to accommodate the clustering procedure within estab-

lished mathematical contexts, such as Expectation-Maximisation (EM). This is a procedure in which parameters of a probabilistic model are estimated, and then improved, iteratively until they converge to a maximum likelihood solution. A version of EM has been developed for use in clustering problems, called the Expectation-Classification-Maximisation (ECM) algorithm (Celeux and Govaert 1992), which is an application of EM to the discrete space of data partitioning.

4.2.1 EXPECTATION-CLASSIFICATION-MAXIMISATION

The general procedure of the EM algorithm is to optimise the parameters of a model by repeatedly applying two steps: an expectation step (E-step) in which a probability distribution is calculated using an estimate of the model parameters, and a maximisation step (M-step) that updates the parameter estimate according to the probability distribution (Do and Batzoglou 2008). These two steps operate in a ratchet-like fashion, only allowing the likelihood to increase, which guarantees that the algorithm is convergent (Dempster, Laird, and Rubin 1977). In the ECM algorithm of Celeux and Govaert (1992), the parameter of interest is the assignment of data points to one of a fixed number of clusters. The assignment is updated by introducing a classification step (C-step) between expectation and maximisation. The expectation step is used to generate a table of probabilities of each data point's membership to each cluster: classification assigns each data point to the cluster with the highest probability of membership, and maximisation updates the estimates of the remaining model parameters.

In the general case, the ECM algorithm proceeds as follows:

- Quantities updated through the algorithm:
  - N data points.
  - K clusters.
  - An assignment vector, $\boldsymbol{c}^{(m)}$, of length N. Each element $c_i^{(m)}$ is an index mapping data point x_i ($i \in \{1, \ldots, N\}$ to cluster k ($k \in \{1, \ldots, K\}$, at iteration m.
  - A matrix, $\boldsymbol{\theta}^{(m)}$, of the parameter values for each cluster, at iteration m.

  - A vector of mixing weights, $\boldsymbol{p}^{(m)}$ of length K; each element is the proportion of the N data points in each cluster.
- Initialise:
  - Fill the vector $\boldsymbol{c}^{(0)}$ to the initial cluster assignments such that all clusters are occupied by at least one data point.
  - Set initial parameter values $\boldsymbol{\theta}^{(0)}$.
  - Maximise the parameter values $\boldsymbol{\theta}^{(0)}$ (see M-step).
  - Fill in $\boldsymbol{p}^{(0)}$ according to the number of points in each cluster.
- E-step: Compute the posterior probability that the ith data element, x_i, belongs to the kth cluster at the mth iteration using the current cluster assignment and parameter values:

$$P_k^{(m)}(x_i \mid c_i^{(m)} = k) = \frac{p_k^{(m)} f(x_i \mid \theta_k^{(m)})}{\sum_{k=1}^{K} p_k^{(m)} f(x_i \mid \theta_k^{(m)})} \tag{4.14}$$

  f is the likelihood function, which specifies the likelihood of $\theta_k^{(m)}$ given the observations x_i in cluster k.
- C-step: Update the assignment vector, $\boldsymbol{c}^{(m+1)}$ by assigning each x_i to the cluster that maximises its posterior probability, computed during the E-step. Update the mixing weights, $\boldsymbol{p}^{(m+1)}$, according to the new assignments.
- M-step: Using the new assignments determined in the C-step, compute maximum likelihood estimates of $\theta_k^{(m+1)}$ by maximising the likelihood function:

$$L(\theta_k^{(m)} \mid \boldsymbol{x}) = \prod_{i,\, c_i^{(m)}=k} f(x_i \mid \theta_k^{(m)}) \tag{4.15}$$

- Repeat E-, C- and M-steps until convergence.

4.2.2 ECM IN PHYLOGENETICS

ECM has previously been used in phylogenetic analysis as a way to cluster sites in a sequence alignment, to look for variation in the evolutionary process among regions of a gene (Bao et al. 2008). I have expanded this ap-

proach to a multilocus context and have applied ECM to clustering loci. In this formulation θ_k are the likelihood model parameters (including the tree topology) estimated when searching for cluster trees, and $f(x_i \mid \theta_k)$ computes the likelihood of the cluster tree and model parameters inferred for cluster k given locus alignment x_i.

The algorithm is initialised by assigning the loci to K clusters — they may be randomly assigned, or may be clustered, for example using a procedure from Chapter 2. Using this initial assignment, a separate phylogenetic model and tree is optimised for each cluster (the precise nature of the model is arbitrary). Then, ECM proceeds as follows:

- E-step: Calculate the likelihood for each locus using the pruning algorithm of Felsenstein (1973). The likelihood model parameters and the tree are those determined during the inference of the cluster tree from the cluster the locus belongs to. These likelihoods replace the $f(x_i \mid \theta_k^{(m)})$ term in equation (4.14), and are used to calculate the posterior probabilities of cluster membership. To avoid errors due to numerical underflow, the calculations are carried out in log-space.
- C-step: Cluster assignments are updated according to the posterior probabilities. Specifically, each locus is assigned to the cluster for which the posterior probability is highest.
- M-step: If a new partition has been generated, the cluster trees are re-estimated. The resulting cluster trees and likelihood model parameters will be used in the next iteration of the algorithm. If there is no change in partition, the algorithm terminates.

4.2.3 IMPLEMENTATION

I implemented the ECM algorithm and the pruning algorithm for computing phylogenetic likelihoods as part of treeCl. I refer to the ECM version of treeCl as treeCl-ECM, to distinguish the partitioning results obtained through ECM from those obtained through the clustering method of Chapter 2. To reduce computation time, during the M-step the cluster tree topologies are estimated using the BIONJ algorithm (Gascuel 1997), a version of the neighbour-joining method of hierarchical clustering, rather than

doing a full maximum-likelihood inference and tree search on each iteration. The free model parameters are estimated using maximum-likelihood. Both BIONJ and the parameter optimisation is done using the program PhyML (Guindon et al. 2010).

I used the ECM algorithm to explore the yeast dataset from section 3.1, as this is a well understood dataset for which I already had a cluster-related interpretation. I partitioned the dataset into 3 clusters, as in Chapter 3. I ran treeCl-ECM 40 times; 20 times starting from a random partition, 20 times starting from the partition found using spectral clustering and geodesic distances (see section 3.1.3).

The partition with the highest likelihood found over the 40 runs was used to perform full maximum likelihood tree inference using PhyML with the WAG model and four categories of gamma-distributed rates.

4.2.4 RESULTS AND DISCUSSION

The progress of each treeCl-ECM run is shown in figure 4.5. In this plot the lines show the sum of the per-locus likelihood values calculated in the E-step of the algorithm for each iteration. The red lines show the runs that were initialised by a random assignment. The blue lines show the runs that were initialised with the spectral clustering result for three clusters. These are all overlapping, and almost horizontal: very little improvement was possible from this starting configuration, and all the runs followed the same trajectory. The red lines show that the algorithm is able to make improvements from random starting partitions, but the improvements all occur within a small number of iterations. It is possible that the search strategy, as implemented in the C-step, may be prone to getting stuck in local areas of the search space.

The highest-likelihood result produced clusters of 224, 91 and 29 loci (table 4.2). Full maximum-likelihood tree estimation was applied to these clusters to produce optimised partition likelihoods (higher than those obtained from the hybrid BIONJ-ML estimates used during optimisation, shown in figure 4.5). The optimised partition likelihood for this result was -1930029.9, while the partition likelihood obtained for the spectral cluster-

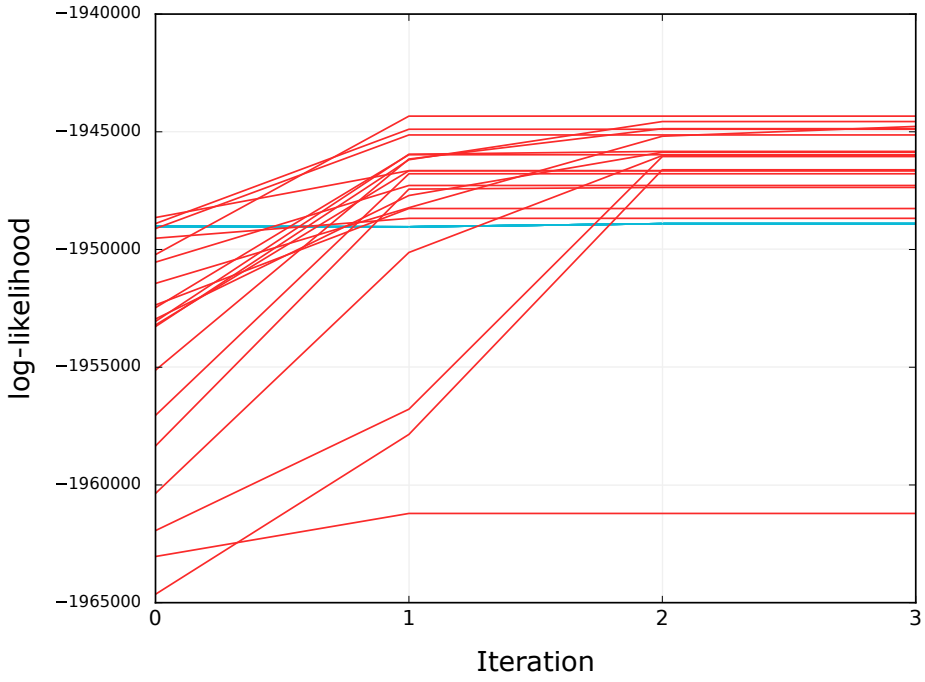


Figure 4.5. *Log-likelihood values obtained during treeCl-ECM optimisation of yeast data. Lines show the likelihood values obtained during the iterative optimisation procedure — these are conditioned on BIONJ topologies, which is used during iteration to reduce computation time. Lines in red show runs that were initialised with a random partition. Lines in blue show runs that were initialised with the partition found by spectral clustering. All runs of the algorithm converged after at most three iterations.*

ing result was −1945206.2. The interpretation of the treeCl-ECM clusters is less clear than for those produced using the methods of Chapters 2 and 3. Not all the loci containing non-orthologous sequences were identified using treeCl-ECM; in addition, the loci for which orthology was not in doubt were divided into separate clusters.

The smallest cluster consists entirely of loci that were identified as containing non-orthologous sequences in Chapter 3. It contains all the loci for which the *S. kudriavzevii* sequence was an incorrectly identified ortholog — all the loci that were in what was referred to as "cluster 2" in Chapter 3, and three members of Chapter 3's "cluster 3". The eight loci containing non-orthologous sequences that were not captured by the 29 locus cluster are distributed over both the larger clusters. The maximum-likelihood trees for the three clusters are shown in figure 4.6.

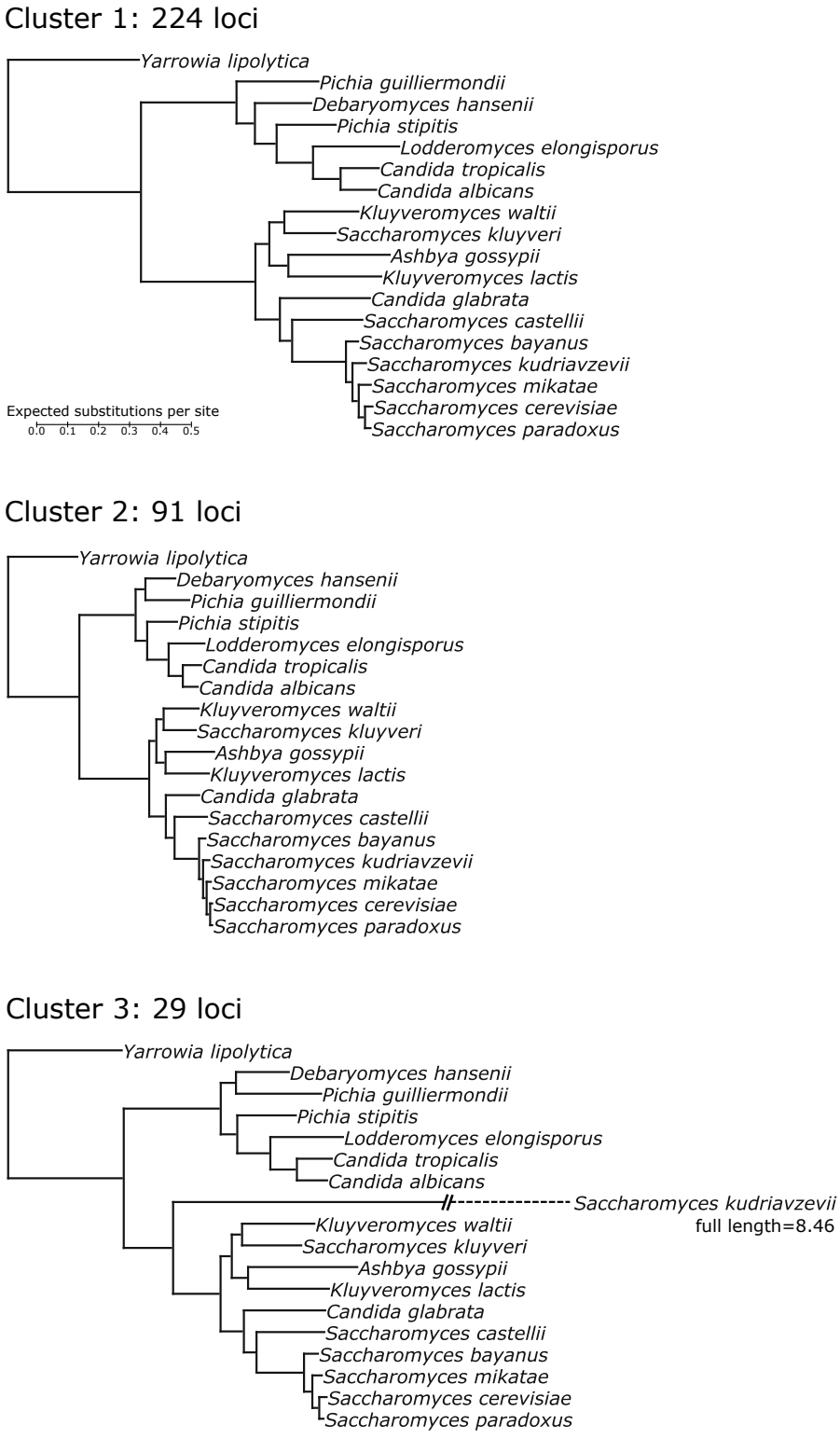


Figure 4.6. *Yeast cluster trees optimised using treeCl-ECM. All trees are drawn to the same scale. Support values (aBayes) for all branches are >0.99*

.

Of the larger clusters, the 91 locus cluster shows the same topology as the largest cluster in both the treeCl and treeCl-ECM two-cluster results (see figure 3.3. The tree for the 224 locus cluster, however, shows *D. hansenii* in a slightly altered position; no longer sister-taxa with *P. guilliermondii*, it is now in a position basal to *P. stipitis*, *L. elongisporus*, *C. tropicalis* and *C. albicans*. This is a small local rearrangement, consistent with explanations of ILS or hybridisation. In a study that looked for evidence of ILS among the Saccharomycetaceae, none was found (Rosenfeld, Payne, and DeSalle 2012). However, Jacques et al. (2015) report a high rate of hybridisation among the *Debaryomyces* genus, and the species *Pichia sorbitophila* is suggested to be a recent hybrid of *D. hansenii* and *Pichia farinosa*, so it is possible that this cluster tree shows the result of hybridisation effects. Further study with joint hybridisation and ILS models, such as in Sang and Zhong (2000) or Kubatko (2009), may help to distinguish between scenarios of hybridisation, ILS or non-mechanistic stochastic variation.

There is also a noticeable difference in overall tree length between the 224 and 91 locus clusters. This suggests that the partitioning is occurring at least partially on evolutionary rate, as the 224 locus cluster seems to consist of fast-evolving loci, and the 91 locus cluster of slow-evolving loci.

The WGD that is considered to have occurred in the ancestor of the *Saccharomyces* yeasts has left no detectable trace on the evolution of the loci considered in these analysis.

CLUSTER	LOCI ASSIGNED TO CLUSTER				
1	YAL009W	YAL010C	YAL032C	YAL044C	YAR003W
	YAR007C	YBR061C	YBR095C	YBR101C	YBR111C
	YBR170C	YBR193C	YBR217W	YBR243C	YBR248C
	YBR251W	YBR260C	YBR271W	YBR282W	YCL016C
	YCL031C	YCL034W	YCL052C	YDL033C	YDL043C
	YDL045C	YDL087C	YDL098C	YDL116W	YDL205C
	YDL207W	YDL212W	YDR023W	YDR036C	YDR049W
	YDR101C	YDR121W	YDR140W	YDR152W	YDR175C
	YDR204W	YDR265W	YDR292C	YDR298C	YDR301W
	YDR306C	YDR315C	YDR361C	YDR362C	YDR364C
	YDR405W	YDR449C	YDR454C	YDR465C	YDR468C

YDR470C YDR496C YDR529C YEL037C YEL038W
YEL062W YER016W YER049W YER078C YER156C
YER183C YFL017C YFR052W YGL017W YGL018C
YGL085W YGL110C YGL122C YGL129C YGL137W
YGL192W YGL243W YGR057C YGR078C YGR080W
YGR095C YGR149W YGR150C YGR171C YGR178C
YGR193C YGR208W YGR215W YGR255C YGR262C
YHL004W YHL013C YHR007C YHR011W YHR059W
YHR062C YHR114W YHR132C YHR144C YHR187W
YHR191C YHR201C YJL006C YJL011C YJL025W
YJL046W YJL085W YJL115W YJL180C YJL203W
YJL208C YJR006W YJR014W YJR050W YJR052W
YJR062C YJR073C YJR102C YJR113C YJR119C
YJR122W YKL045W YKL047W YKL119C YKL138C
YKL149C YKL175W YKL179C YKL193C YKL205W
YKR038C YLR015W YLR023C YLR051C YLR059C
YLR084C YLR195C YLR209C YLR239C YLR253W
YLR288C YLR292C YLR330W YLR370C YLR409C
YLR418C YML004C YML021C YML030W YML036W
YML080W YML096W YML127W YMR009W YMR013C
YMR026C YMR038C YMR055C YMR061W YMR150C
YMR188C YMR193W YMR197C YMR201C YMR208W
YMR211W YMR218C YMR281W YNL005C YNL010W
YNL072W YNL177C YNL252C YNL291C YNL292W
YNL306W YNL308C YNL310C YNL315C YNR017W
YNR039C YNR052C YNR054C YOL005C YOL022C
YOL041C YOL093W YOL124C YOL135C YOL142W
YOL145C YOR067C YOR070C YOR077W YOR095C
YOR111W YOR160W YOR164C YOR166C YOR236W
YOR243C YOR249C YOR250C YOR251C YOR289W
YPL001W YPL002C YPL030W YPL065W YPL104W
YPL107W YPL109C YPL126W YPL149W YPL157W
YPL172C YPL183C YPL210C YPL239W YPR031W
YPR056W YPR073C YPR139C YPR143W

2 YBL015W YBR135W YBR254C YCL054W YCL059C
YCR047C YDL051W YDL100C YDL165W YDL198C
YDL201W YDL215C YDR041W YDR148C YDR339C
YDR373W YDR448W YDR460W YEL024W YEL050C
YER009W YER012W YER023W YER043C YER086W
YER107C YER136W YFL046W YFR044C YFR050C

	YGL040C	YGL043W	YGR005C	YGR083C	YGR172C
	YGR285C	YHR025W	YHR058C	YHR088W	YHR193C
	YJL030W	YJL033W	YJL072C	YJL104W	YJL121C
	YJL166W	YJR010W	YJR063W	YJR121W	YKL003C
	YKL016C	YKL028W	YKL041W	YKL058W	YKL080W
	YLR017W	YLR026C	YLR078C	YLR244C	YML110C
	YML121W	YMR002W	YMR015C	YMR131C	YMR236W
	YMR241W	YMR260C	YMR276W	YMR290C	YMR314W
	YNL025C	YNL071W	YNL220W	YNL287W	YNR007C
	YOL010W	YOR065W	YOR128C	YOR142W	YOR150W
	YOR168W	YOR211C	YOR262W	YOR271C	YOR361C
	YOR370C	YPL094C	YPR103W	YPR110C	YPR133C
	YPR166C				
3	YBL080C	YBR094W	YBR290W	YCR068W	YDL104C
	YEL053C	YFR051C	YGL236C	YHR019C	YHR020W
	YHR024C	YHR075C	YJL054W	YJL071W	YJR141W
	YKL060C	YMR224C	YNL219C	YNL232W	YNL256W
	YNL325C	YNR029C	YOL097C	YOR125C	YOR201C
	YPL188W	YPL244C	YPR025C	YPR118W	

Table 4.2. *This table lists all 344 loci in the yeast dataset. The loci are listed according to their cluster membership when data are partitioned into 3 clusters using treeCl-ECM.*

4.2.5 SUMMARY

Here I have implemented the ECM algorithm of Celeux and Govaert (1992) to cluster loci as treeCl-ECM. I tested its clustering performance on the yeast dataset introduced in Chapter 3. I was able to find partitions with higher likelihood than the ones obtained using the treeCl clustering method. However, the resulting clusters had a less clear interpretation. The clusters found in Chapter 3 were consistent with a simple explanation: the clustering structure had identified loci for which some orthology assignments had been made in error, and even conveyed something of the details of that error (in that it identified a cluster of loci in which the error affected a particular species). Applying treeCl-ECM distinguished between loci with and without correctly established orthology relationships, but did so less cleanly. It identified a subset of the non-orthologous loci in one cluster, and divided the remaining loci into two large clusters. The inferred phylo-

genies show that one of the larger clusters has a topological departure from the established species tree that may indicate an evolutionary process such as hybridisation, or may simply reflect stochastic variation. Determining whether there is an underlying evolutionary explanation for the different topology is a matter for further investigation.

The traces presented in figure 4.5 suggest that the search procedure used in treeCl-ECM may not be efficiently exploring the parameter space. The search procedure, which takes place in the C-step, is quite simplistic, and there are several possible modifications that may be useful. For example, the classification step makes a batch reassignment — on each iteration it assigns every locus to the cluster for which its membership probability is highest. It may be that when many loci are reassigned at once this operates as a large move in the explored space. Limiting the number of reassignments made in each classification step would allow for a finer-grained exploration, at the potential expense of greater runtime, due to requiring more iterations. Periodic use of other operations, such as splitting a cluster into two, or merging two clusters into one, could be tried as an alternative to the classification step, to create a wider range of ways to navigate the space of all partitions.

Computing phylogenetic likelihood is computationally expensive (Flouri et al. 2015), and this restricts the number of partitions treeCl-ECM can explore. An alternative approach would be to use distance-based methods as faster, though statistically less efficient, means of estimating and assessing phylogenetic trees (Yang 2014). Least-squares estimation builds a tree using a matrix of the evolutionary distances between pairs of sequences in an alignment (Hillis et al. 1996). The discrepancy between the pairwise distances and the distances induced by the tree is measured as the sum of squared residuals. Modelling these values as draws from a particular probability distribution (e.g. Normal) allows us to interpret them as a likelihood, thereby allowing them to be incorporated in treeCl-ECM.

4.3 SUMMARY

In this chapter I addressed the issue of uncertainty in both tree estimation, and cluster assignment. I outlined a way in which tree estimation uncertainty, as expressed by the bootstrap, could be used to inform us about the distribution of trees in tree-space. This could be used in a simple extension to the treeCl clustering method Chapters 2 and 3 by augmenting the input distance matrix with bootstrap distances using one of the rapid approximations described in section 4.1. One could then cluster loci based on the distances among loci, and take into account the uncertainty in the tree estimation through the use of bootstrap replicates. By introducing constraints in the clustering procedure, locus trees can be constrained (with high probability) to fall into the same clusters as their bootstrap counterparts, the error in clustering could be reduced, or at least made less sensitive to the noise in tree estimation. This could be achieved by using a semi-supervised constrained clustering method in place of the direct clustering techniques so far considered, such as non-negative matrix factorisation (Zitnik and Zupan 2012), or by including a prior on whether locus trees and bootstrap trees should be clustered together in mixture models (Georgi, Costa, and Schliep 2010). The use of probabilistic mixture models is particularly appealing, as the degree to which the bootstraps affect how granularly the locus trees are clustered could be tuned through the prior probability that locus trees cluster with their bootstraps.

An iterative, likelihood-based procedure derived from expectation-maximisation can be used to try to improve the cluster assignments given to a set of loci. This allows cluster assignment to be linked back to widely used probabilistic phylogenetic models of sequence evolution. This injects an amount of rigour back into what is otherwise a somewhat heuristic method. However, further work is required to see whether this is effective, or if it is tractable, with both search complexity and computational burden proving to be a concern.

5

OTHER WORK

There were a number of other lines of work that I pursued during my PhD research in addition to the treeCl clustering project that was my main coherent research project. I give details of them in this short chapter. The publications that arose as a consequence of this work are reproduced in appendices A to D.

5.1 REVIEW OF MULTIPLE SEQUENCE ALIGNMENT BENCHMARKS

In the first year of my PhD I took the University of Cambridge post-graduate course 'Reviews in Computational Biology: Assimilate, Write and Evaluate Reviews' (`http://christophe.dessimoz.org/revcompbiol/`). For this course I jointly authored a review of multiple sequence alignment (MSA) methods that was ultimately published as 'Who watches the watchmen: an appraisal of benchmarks for multiple sequence alignment', a chapter of the book *Methods in Molecular Biology* (Iantorno et al. 2014). This review discussed the intentions behind MSA. In the review, we consider the purpose of multiple sequence alignment to be to infer related residues among biological sequences, but recognise that different research fields may have different interpretations of what it means for residues to be related. For example, for a structural biologist related residues occupy the same structural position or role within a protein, while for an evolutionary biologist related residues are those descending from a common ancestor. Although there are these subtle differences in interpretation of the method, MSA is a fundamental and ubiquitous technique in bioinformatics. Thus accuracy of alignment is a crucial consideration. A number of benchmarking strategies have been pursued to compare the performance of different aligners and

help detect systematic errors in alignments. In the review we present an overview of the main strategies — based on simulation, consistency, protein structure, and phylogeny (figure 5.1) — and discuss their different advantages and associated risks. We outline a set of desirable characteristics for effective benchmarking, and evaluate each strategy in light of them. We conclude that there is currently no means of benchmarking MSA that is universally applicable across research fields, and that developers and users of alignment tools should base their choice of benchmark depending on the context of application — with a keen awareness of the assumptions underlying each benchmarking strategy.

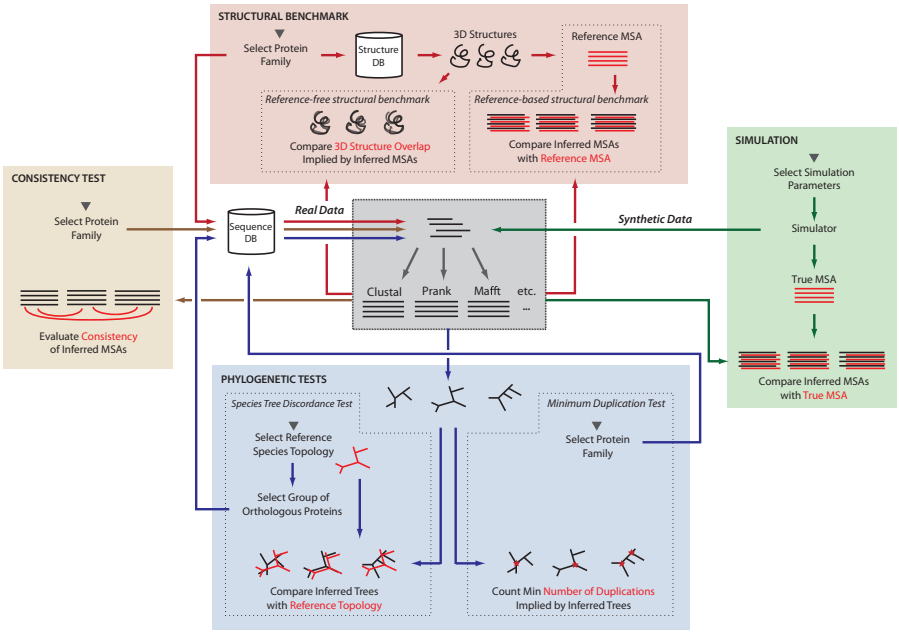


Figure 5.1. *Schematic of the four main MSA benchmarking strategies described in the review Who watches the watchmen: an appraisal of benchmarks for multiple sequence alignment. For each approach, the benchmarking process starts from the corresponding downward-pointing arrow and involves alignment by different multiple sequence alignment methods (grey box in centre, illustrating example aligners that may be benchmarked). This figure was used as Figure 1 in Iantorno et al. (2014).*

5.2 CONTRIBUTIONS TO THE FERRET GENOME PROJECT

In the second year of my PhD I contributed to the project that produced the draft genome sequence of the ferret *Mustela putorius furo*, published in Peng et al. (2014). The ferret is used as a model organism for the study of

human respiratory disease, and one of the main objectives of the project was to compare the ferret's usefulness in this capacity with the mouse, another frequently used model organism in many aspects of human health. My contribution was to perform comparative genomics between the ferret sequences and several whole-genome sequences taken from the Ensembl (release 70) database (Flicek et al. 2014), and Genbank (Benson et al. 2005). The results of my work were to produce a phylogenetic tree of 34 organisms including the ferret to place the ferret in its evolutionary context (figure 5.2), and to perform three-way comparisons between one-to-one orthologs among human, ferret and mouse, to contrast the levels of sequence divergence between human and ferret and between human and mouse (figure 5.3). I inferred the orthologous relationships underlying the tree and the three-way comparisons using the OMA pipeline (Altenhoff et al. 2011). The tree was built from the concatenated alignments of 789 orthologous groups that covered at least 31 of the 34 species. Divergence among triplets of orthologous protein sequences was computed in point accepted mutation (PAM) units estimated by pairwise maximum likelihood distance estimation using Gonnet matrices (Gonnet, Cohen, and Benner 1992).

Although mice and humans are taxonomically more closely related than humans and ferrets — i.e. mice and humans share a more recent common ancestor than do humans and ferrets — branch lengths in the tree indicate rapid evolution in the rodent clade, which has resulted in less genetic divergence between humans and ferrets than between humans and mice. In comparing protein sequences between the species, I found that for 75% of all orthologous triplets, ferret proteins are closer than mouse proteins to human proteins. For example, the ferret cystic fibrosis transmembrane conductance regulator (CFTR) protein is considerably closer to the human protein than is its mouse counterpart (percentage identities [PAM distance] for ferret to human = 92% [8.1]; mouse to human = 79% [23.3]). In summary, the overall high sequence similarity between ferret and human proteins shown by these genome-level analyses indicates that many ferret proteins have likely evolved to conserve similar molecular functions to their human

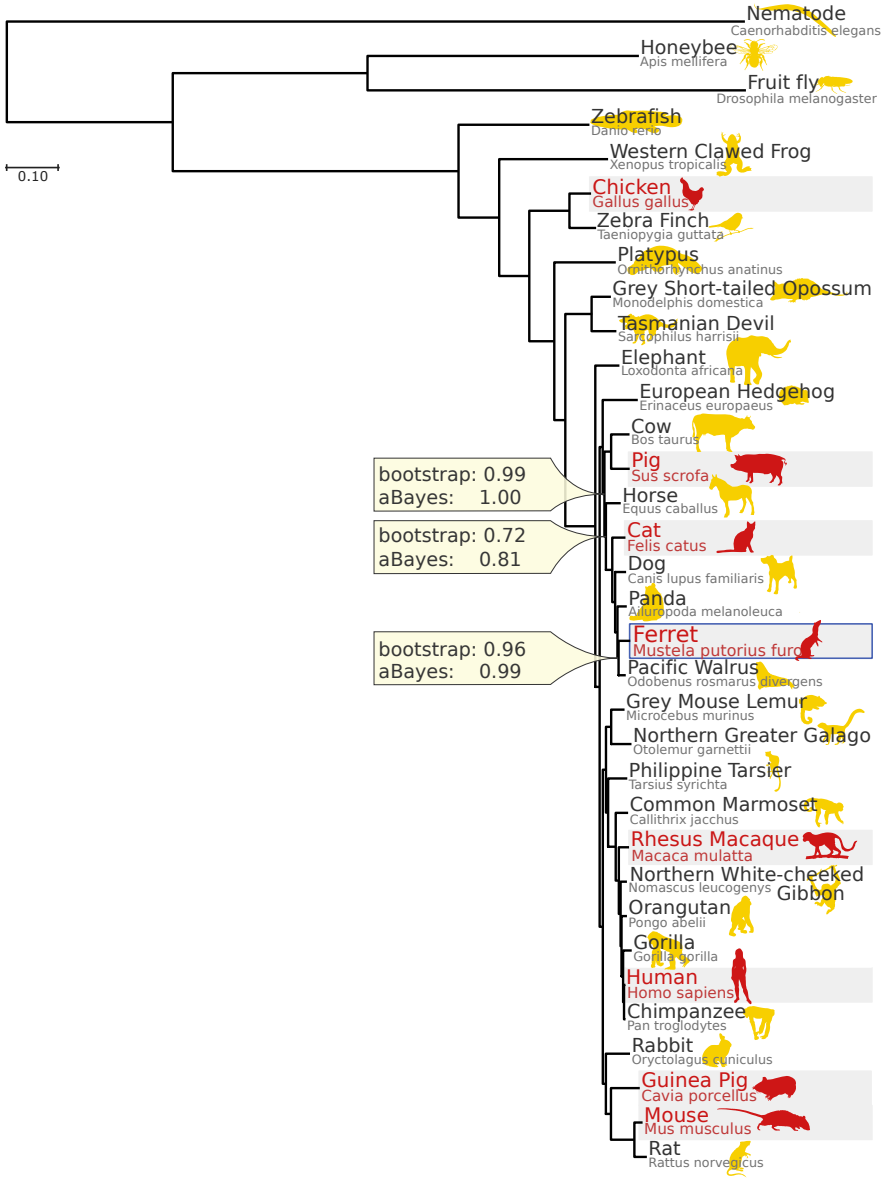


Figure 5.2. *Phylogenetic tree based on protein sequences of 789 orthologous groups with representation in at least 31 of the 34 species. Species commonly used for respiratory models of human health and disease are highlighted in red. Bootstrap and approximate Bayesian (aBayes) support values take their maximum possible value of 1.0 at all nodes except in the three indicated. Images are from* `http://www.phylopic.org`. *This figure was used as Supplementary Figure 2 in Peng et al. (2014).*

protein orthologs, which supports the use of ferret as a suitable model organism for studying human respiratory health.

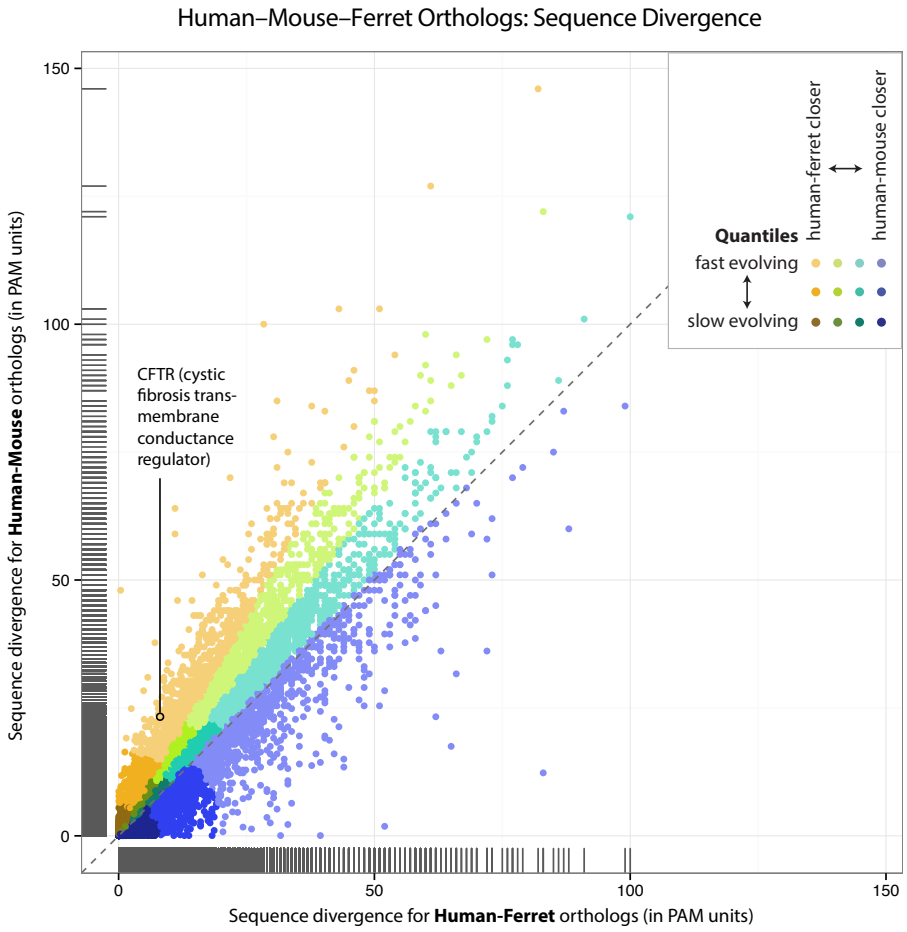


Figure 5.3. *Scatter plot of human vs. mouse protein divergence measured in point accepted mutation (PAM) units (*y*-axis) against the corresponding human vs. ferret protein divergence (*x*-axis). Proteins appear above the 45° diagonal (grey dashes) when the ferret sequence is closer to the human sequence than the corresponding mouse sequence. The angle of the line to each protein from the origin is directly related to the ratio of mouse divergence from human sequence and ferret divergence from the human sequence. A greater angle from the origin indicates greater divergence. The quartiles of the distribution of these ratios are displayed in different colours (blue being the least conserved in ferret relative to mouse, and orange-brown being the most conserved). Hatched lines on the axes show the marginal densities of the scattered points. This figure was used as Figure 1a in Peng et al. (2014).*

5.3 CONTRIBUTION TO THE OMA ORTHOLOGY INFERENCE TOOLS

I am a lead contributor to the FamilyAnalyzer tool `https://github.com/DessimozLab/familyanalyzer`, part of the software toolbox associated with the OMA orthology database (Altenhoff et al. 2015). The OMA algorithm is used to assign orthology and paralogy relationships between biological sequences (Altenhoff et al. 2013). FamilyAnalyzer maps these relationships onto a phylogenetic tree, allowing them to be interpreted as *de novo* gene gain, duplication and loss events, summarising the way gene families evolve over time (see figure 5.4 for an example). FamilyAnalyzer also allows estimation of the ancestral origin of genes and gene families, which potentially allows it to be used in the field of phylostratigraphy (Domazet-Lošo, Brajković, and Tautz 2007).

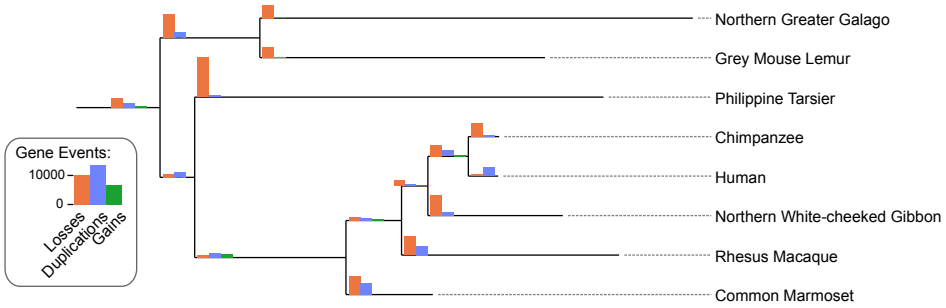


Figure 5.4. *Gene duplications, losses and gains on the primate lineage inferred from OMA hierarchical orthologous groups. Numbers of inferred gene evolutionary events are displayed as bars; these events were mapped onto the tree using FamilyAnalyzer. This figure was used as Figure 6 in Altenhoff et al. (2015). Figure credit: Steven Müller.*

6

CONCLUSION

Incongruence, in which different sources of evidence lead to the inference of different evolutionary trees for a set of taxa, is a well-known problem in molecular phylogenetics. While it was expected that multi-locus phylogenetics, or phylogenomics, would be able, through volume of data, to overcome incongruence and enable the inference of clear, unambiguous evolutionary trees, this has not been borne out. The reason for this is that incongruence is not only a reflection of uncertainty, but also of the heterogeneous processes of evolution that act on the genomes of living things.

It is possible to incorporate diverse evolutionary processes explicitly into the analysis of evolution, by designing models of evolution that take into account various mechanisms of evolution. It is also viable to approach evolutionary analysis from what I have termed a 'process-agnostic' standpoint, in which incongruence is accommodated, but not attributed to specific processes. The approach I have taken in this thesis has been process-agnostic. In the work presented here, I have described a method, which I have called the treeCl method, for partitioning sets of biological sequences (loci) into clusters so that the phylogenetic incongruence among the loci is minimised within clusters. The treeCl method works via a process of clustering loci by the similarity of the individually inferred evolutionary trees, and assigning them to clusters that share a common topology, within which incongruence is minimised. The loci within the resulting clusters are then concatenated, and analysed using the standard tools of phylogenetic inference. Through doing this I am able to identify the common themes of the evolutionary processes that have acted on sequences, aiding identification of likely processes that are at play. I have shown that a process-agnostic approach to evolutionary inference is complementary to mechanistic approaches.

The treeCl clustering method involves inferring per-locus trees, and then, based on the distances between them, partitioning the loci into clusters of similar trees. In my implementation I require both a distance metric to use to compare trees, and a method to produce clusters using the distance information. In Chapter 2 I reviewed several distance metrics and clustering methods. By extensive simulation experiments I measured the performance of combinations of these distance metrics and clustering methods at reproducing known partitions of simulated loci. Distance metrics that take branch length information into account, as opposed to measuring distances based only on tree topology, were shown to be most effective. The branch-length-sensitive metric that was most effective was the geodesic distance, that provides a measure of distance that reflects the non-Euclidean nature of tree-space, as described by Billera, Holmes, and Vogtmann (2001).

The perennial problem with clustering is ascertaining the appropriate number of clusters. Too few leads to overlooking structural features present in the data, and introducing bias by forcing data that have differing characteristics together. Too many and the picture presented becomes difficult to interpret. In treeCl the number of clusters is estimated by applying a novel stopping criterion that involves resampling and reanalysing data, and using the statistical principle of likelihood to determine whether the improvement in goodness of fit observed when increasing the number of clusters is statistically significant. In Chapter 2 two variants of the stopping criterion were tested; both proved to be better at estimating the number of clusters than a general purpose heuristic, the silhouette measure, which was included for comparison. Although the results show that the stopping criterion proved effective—and robust to missing data—both variants are computationally expensive. Of the two variants the one based on permutation is the least onerous.

I developed the treeCl method by testing it on simulated data, but my aim was to apply it to real, large-scale datasets, to see if it could reveal something about the actual process of evolution, rather than an idealised case. As described in Chapter 3 I investigated two empirical datasets in detail, one

from yeast and one from *Chiastocheta* flies. These datasets contain sequence alignments numbered in the hundreds, placing them on a similar scale to data that is routinely produced by high-throughput sequencing.

Both datasets showed a high degree of phylogenetic incongruence. The yeast dataset had no missing data: for each species there were sequences data available for every locus. A high degree of incongruence was detected. For this dataset, the principle discovery was that misannotated orthology caused the greatest amount of incongruence, with no detectable incongruence being witnessed for those loci for which the orthology was accurately established.

For the *Chiastocheta*, there was a large amount of missing data, and what data there was for any individual locus was not enough to establish a fully resolved tree. Cluster trees largely recovered monophyletic species, though the branching order of the species varied between clusters. ILS is likely to be the leading cause of incongruence.

Due to the process-agnostic nature of treeCl, I was able to apply the method in the same way to both datasets, and learn something about the incongruent signals in the data. This allowed me to speculate as to the likely processes at play, and prioritise different types of follow-up analysis — stringent orthology identification in the first case, and analysis under a mechanistic ILS model in the second. In this way a process-agnostic approach is complementary, rather than in opposition, to using mechanistic models of incongruence.

Up to this point, what had been missing in the development of treeCl was any treatment of uncertainty in either tree estimation or cluster assignment. I tried to address these limitations in Chapter 4. I outlined a way in which uncertainty in tree estimation could be incorporated into the analysis of tree distribution in tree-space by including bootstrap replicate trees. There is substantial computational overhead in both estimating bootstrap trees, and in obtaining their distances in tree-space, so I developed approximate methods to reduce the amount of computation required. These were shown

to make faithful representations of the distribution of trees in tree-space while substantially reducing the amount of computation required.

I also implemented an iterative, likelihood-based procedure derived from expectation-maximisation to estimate and refine the cluster assignments given to a set of loci, using probabilistic phylogenetic models of sequence evolution. Application of this method to the yeast dataset introduced in Chapter 2 led to the discovery of a partition of the data with a much higher likelihood. This was perhaps less successful at identifying the errors that had been discovered in the dataset, but hinted at the possibility of there being some biological basis to the incongruence observed in these data, if the signal detected turns out to be more than noise.

I tested the approaches to dealing with uncertainty displayed in Chapter 4 to a 'proof-of-concept' extent. Further investigation is needed to test the approaches more thoroughly, which may lead to better understanding of the problem and help to develop more refined approaches.

The aim of this thesis was to investigate a process-agnostic approach to tackling phylogenetic incongruence. I have shown that clustering approaches coupled with measures of the distance between inferred evolutionary trees can be used successfully to tease apart the influences of disparate evolutionary forces, and generate novel hypotheses.

A

WHO WATCHES THE WATCHMEN

I shared first author status with Stefano Iantorno on this review, which was published as a book chapter (see chapter 5 for details). Although I contributed throughout, I had particular responsibility for the sections on structural benchmarks and phylogenetic tests of alignments.

Chapter 4

Who Watches the Watchmen? An Appraisal of Benchmarks for Multiple Sequence Alignment

Stefano Iantorno, Kevin Gori, Nick Goldman, Manuel Gil, and Christophe Dessimoz

Abstract

Multiple sequence alignment (MSA) is a fundamental and ubiquitous technique in bioinformatics used to infer related residues among biological sequences. Thus alignment accuracy is crucial to a vast range of analyses, often in ways difficult to assess in those analyses. To compare the performance of different aligners and help detect systematic errors in alignments, a number of benchmarking strategies have been pursued. Here we present an overview of the main strategies—based on simulation, consistency, protein structure, and phylogeny—and discuss their different advantages and associated risks. We outline a set of desirable characteristics for effective benchmarking, and evaluate each strategy in light of them. We conclude that there is currently no universally applicable means of benchmarking MSA, and that developers and users of alignment tools should base their choice of benchmark depending on the context of application—with a keen awareness of the assumptions underlying each benchmarking strategy.

Key words Multiple sequence alignment, Benchmarking, Phylogenetic, Protein structure, Sequence evolution, Consistency, Homology

1 Introduction

Multiple sequence alignment (MSA) has become a common first step in the analysis of sequence data for downstream applications such as comparative genomics, functional analysis and phylogenetic reconstruction. Given their importance, MSA methods need to be objectively validated in order to ensure their output is both accurate and reproducible. Benchmarking is a crucial tool in the assessment of sequence alignment programs, as it allows their developers and users to compare the performance of different aligners objectively, identify strengths and weaknesses and help detect systematic errors in alignments. In recent years, there has been a growing

Stefano Iantorno and Kevin Gori contributed equally to this work.

David J. Russell (ed.), *Multiple Sequence Alignment Methods*, Methods in Molecular Biology, vol. 1079,
DOI 10.1007/978-1-62703-646-7_4,

appreciation of the importance of benchmarking measures and datasets to evaluate and critically examine the performance of different MSA software packages, as underscored by a number of recent articles addressing the subject [1–5].

At the same time, and despite these positive developments, the standard approach adopted by the great majority of scientists dealing with sequence alignment has remained reliance on aligners that have long been outperformed in benchmarks [6], or even manual and therefore inevitably subjective intervention in the alignment process [7]. It is unclear whether this is due to the simplicity of use and convenience of long-standing aligners ("historical inertia" [7]), reluctance to move away from customary practice, or unawareness or even distrust of newer, lesser-tested technologies. This trend is particularly worrying in light of the rapid spread of high-throughput technologies and the associated need for automation of analysis pipelines [8]. A reason for this state of affairs might lie upon the absence of a straightforward alignment benchmarking procedure and interpretation. In this chapter, we contribute to overcoming this problem by reviewing present alignment benchmarks, aiming to clarify their strengths and risks for MSA evaluation with a view towards having better (and better-trusted) benchmarks in the future. But before considering benchmarking strategies, we first need to review the alignment objectives we expect them to gauge.

1.1 What Should Sequence Aligners Strive for?

A conceptual complication lies in the fact that MSAs have multiple and potentially conflicting goals, depending on the biological question of interest [9]. Commonly, the residues aligned are those inferred to be related through homology, i.e., common ancestry. In other contexts, however, the emphasis might be more on functional or structural concordance among residues. A strictly evolutionary interpretation of homology in these cases could be counterproductive, as recognized also by Kemena and Notredame [1], since regions of the protein that carry out the same function or that occupy the same position in the three-dimensional conformation of the protein may have arisen independently by evolutionary convergence. For example, an alignment that pairs structurally analogous, but nonhomologous, residues would be informative and therefore "correct" to the structural biologist, although not so to the phylogeneticist. It should however be noted that functional and structural objectives are considerably less precise than the evolutionary objective: while common ancestry is an absolute, binary attribute, similarity in functional or structural role are context-dependent, continuous attributes, thus rendering any reduction to the aligned/unaligned dichotomy subjective at best, ill-defined at worst.

At the same time, the unambiguous nature of the evolutionary objective does not make it automatically easy to pursue (or, as we shall see below, ascertain). Indeed, the evolutionary history of biological sequences is mostly unknown and can only be inferred

from present data under the (explicit or implicit) assumption of a model of sequence evolution.

In practice, most MSA methods muddle the distinction among homology-, structure-, or function-motivated alignment by employing strategies anchored in inconsistent objectives. Indeed, almost all well-established aligners assume and exploit evolutionary relationships among the sequences (e.g., by constructing the alignment using an explicitly phylogenetic guide tree and alignment scores derived from models of sequence evolution). Yet many use at the same time structural criteria in their parameters or heuristics, for example by training their parameters using structure-derived reference alignments [10, 11]. The complications of the strategies different aligners employ can however be divorced from the measurement of their success, and we wish to make no assumption that an aligner employing one strategy necessarily performs better when assessed according to criteria consistent with its internal methods. In the present context of alignment benchmarking, we therefore treat aligners as "black boxes" and refer the reader interested in the specifics of alignment methods to later chapters.

1.2 Aims and Desirable Properties of Alignment Benchmarks

As mentioned in the introduction, benchmarks provide ways of evaluating the performance of different MSA packages on standardized input. The output produced by the different programs is compared to the "correct" solution, the so-called gold standard, that is defined by the benchmark. The extent of similarity between the two then defines the quality of the aligner's performance.

Proper benchmarking is advantageous to both the user and the developer community: the former obtains standardized measures of performance that can be consulted in order to pick the most appropriate MSA tools to address a particular alignment problem, and the latter gains important insight into aspects of the software that need improvement, or new features to be implemented, thus promoting advancement of the field [2].

Which characteristics do benchmarks and the gold standard reference dataset need to satisfy in order to be useful to the user and developer community? Benchmarks can be critically examined by looking at their ability to yield performance measures that reflect the actual biological accuracy (whether defined in terms of shared evolutionary history or structural or functional similarity of the aligned sequence data) of the MSA method. This can most easily be done by defining a set of predetermined criteria for good benchmarking practice. We follow Aniba et al. [2] in their list of desirable properties of benchmarks, which states that a benchmark should be:

- *Relevant*, in that a benchmark should be reflective of actual MSA applications, i.e., tasks carried out by MSA in practice and not in an artificial or hypothetical setting.

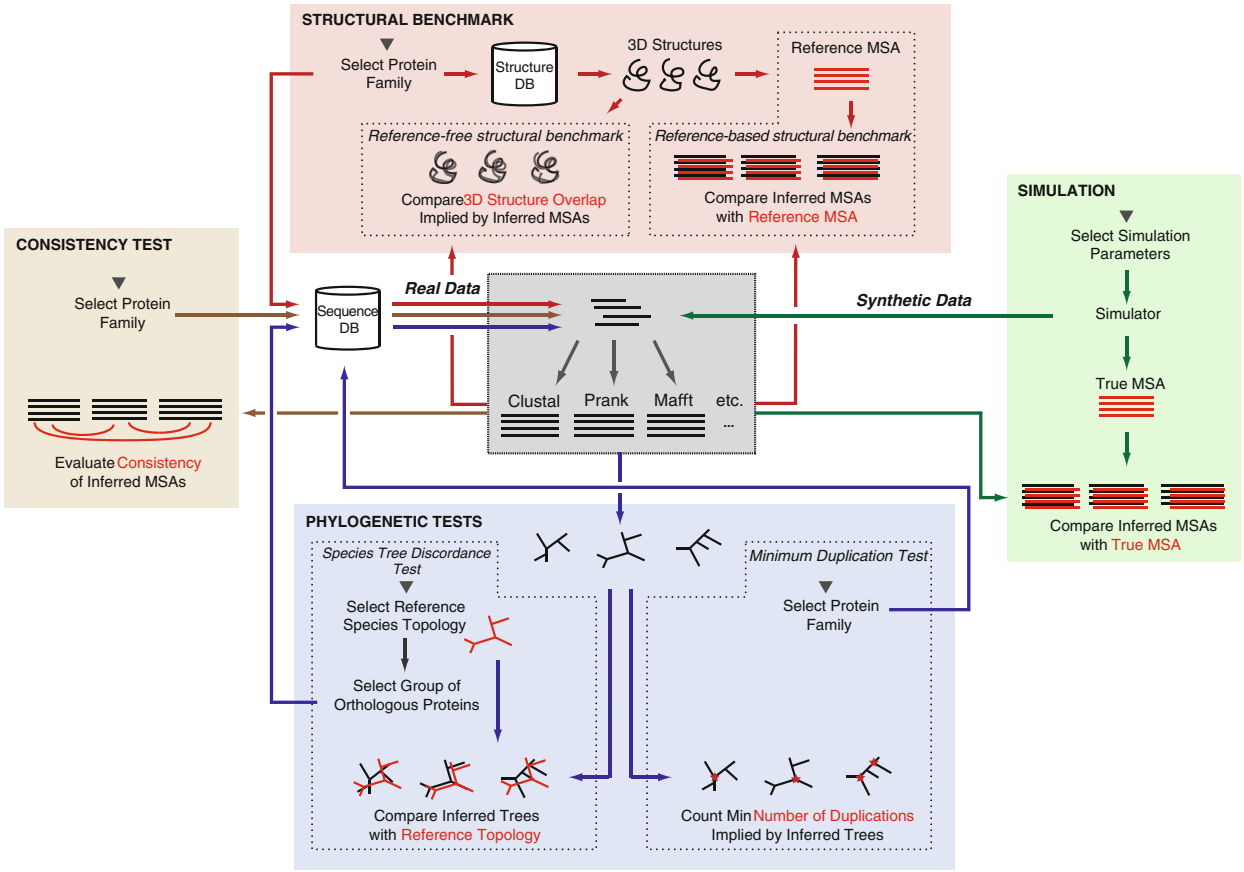


Fig. 1 Schematic of the four main MSA benchmarking strategies of this review: for each approach, the benchmarking process starts from the corresponding *downward-pointing arrow* and involves alignment by different MSA methods (*gray box* in center, illustrating example aligners that may be benchmarked)

- *Solvable*, in that it provides sufficient challenge to differentiate between poor and good performances, while remaining a tractable problem.
- *Scalable*, so that it can grow with the development of MSA programs and sequencing technologies.
- *Accessible*, in order to be widely used by developers and users.
- *Independent* from the methods used by programs under test, as benchmark datasets should avoid any overlap with the heuristics chosen for construction of MSA in order to constitute an objective reference.
- *Evolving*, to reduce the possibility of developers adapting their programs to a particular test set over time, thus artificially inflating their scores.

Although MSA methods employ different computational solutions to reconstruct sequence alignments, their performance needs to be assessed on the same benchmarks in order to be objectively evaluated and compared. In this chapter, we consider four broad MSA benchmarking strategies (Fig. 1):

1. Benchmarks based on simulated evolution of biological sequences, to create examples with known homology.

2. Benchmarks based on consistency among several alignment techniques.
3. Benchmarks based on the three-dimensional structure of the proteins encoded by sequence data.
4. Benchmarks based on knowledge of, or assumption about, the phylogeny of the aligned biological sequences.

In the remainder of this chapter, we analyze each of these benchmarking approaches to point out their pros and cons, and determine how well they satisfy the criteria defined above and summarized in Table 1.

2 Simulated Sequences

Given that a major objective of MSA is to identify residues that evolved from a common ancestor, i.e., to optimize for homology in the alignment, one approach to benchmarking involves generating families of artificial sequences by a process of simulated evolution along a known tree. Such simulation-based approaches adopt a probabilistic model of sequence evolution to describe nucleotide substitution, deletion, and insertion rates, while keeping track of "true" relationships of homology between individual residue positions. Since these are known, a "true" reference alignment and a test alignment based on the simulated sequence data, assembled by a particular MSA tool of choice, can be compared and measures of accuracy estimated (see below). There are many packages that will perform simulated sequence evolution, including Rose [12], DAWG [13], EvolveAGene3 [14], INDELible [15], PhyloSim [16], REvolver [17], and ALF [18].

To quantify the agreement between the reconstructed alignment and the true alignment (known from the simulation), two measures of accuracy are commonly employed: the sum-of-pairs (SP) and the true column (TC) scores [19]. The former is defined as the fraction of aligned residue pairs that are identical between the reconstructed and true alignment, averaged over all pairwise comparisons between individual sequences; the latter is defined as the fraction of correctly aligned columns that are reproduced in the reconstructed alignment. Given that the TC score considers whole columns in the alignment as comparable units, a single misaligned sequence can reduce the TC score to zero. For this reason, when considering numerous or divergent sequences, the finer-grained SP score is usually used. Yet even the SP score is not without problems. For instance, pairwise comparisons ignore correlations among sequences, meaning that closely related sequences contribute disproportionately more to the SP score than they do to the total phylogenetic information contained in the alignment; this can be misleading in phylogenetic applications. More generally, SP and TC

Table 1
The advantages and risks of the four approaches to MSA benchmarking. Examples are given of relevant software packages, benchmark databases and tests

Approach	Advantages	Risks	Examples	References
Simulation-based	Solvability: "true" homology is known	Relevance: simulated data might strongly differ from real biological data	Rose	[12]
	Evolving: different scenarios can be modelled	Independence: MSA parameters might resemble those used in simulation	DAWG	[13]
	Scalability: new data can be generated ad libitum		EvolveAGene3	[14]
			iSGv2.0	[48]
			INDELible	[15]
			PhyloSim	[16]
			ALF	[18]
Consistency-based	Scalability: not constrained to a particular reference set	Relevance: consistent MSA methods may be collectively biased	MUMSA	[26, 49]
	Accessibility: tests are easy and quick	Independence: similar scores might be used in MSA inference	HoT	[27]
Structure-based	Relevance: closely matches a major biological objective of MSA	Relevance: limited to structurally conserved regions; biological objective of MSA may vary	HOMSTRAD	[10, 30]
	Independence: empirical data is used as input	Scalability: only applicable to small subset of protein sequences	OXBench	[40]
			PREFAB	[33]
			SABMARK	[32]
			BAliBASE 3.0	[11, 31]
			STRIKE	[50]
Phylogeny-based	Relevance: closely matches a major biological objective of MSA	Relevance: biological objective of MSA may vary from phylogenetic reconstruction	Species-tree discordance test	[44]
	Independence: empirical data is used as input		Minimum duplication test	[44]
	Scalability: broad array of sequence data can be used as input			

are not proper metrics (they do not satisfy the conditions of symmetry or triangle inequality), which has motivated the recent development of better-founded alternatives [20].

Besides the advantage of knowing the true alignment, the fact that the parameters for simulated sequence evolution are user-defined directly translates into great flexibility to address specific questions or to investigate the effect of individual factors in

isolation of others, which is particularly useful to gain insights into the behavior of complex alignment pipelines. For instance, Löytynoja and Goldman used simulated sequences to expose the systematic underrepresentation of the number of insertions by many aligners, which is especially true as sequence divergence and the number of sequences increases [21].

At the same time, the high level of flexibility afforded by simulation ties in with its biggest drawback: all observations drawn from simulated data depend on the assumptions and simplifications of the model used to generate these data. The vague notion of "realistic simulation" is often used to justify reliance on simulations capturing relevant aspects of real data, but simulations cannot straightforwardly, if at all, account for all evolutionary forces. The risk thus becomes the benchmarking of MSA programs in scenarios of little or no relevance to real biological data. For instance, Golubchik et al. investigated the performance of six aligners by simulating sequences in which gaps of constant size were placed in a staggered arrangement across all sequences [22]; although this scenario might be useful to emphasize a more general problem in aligning regions adjacent to gaps, its very artificial nature makes it a poor choice to gauge the extent of that problem on real data.

A further potential risk is the use of simulation settings more favorable to some packages than others [23]. For instance, the selected model of sequence evolution might resemble the underlying model of a particular aligner and thus provide it with an "unfair" advantage (i.e., presumably unrepresentative of typical situations) in the evaluation. Even when the evaluation is conducted in good faith, the high complexity of many MSA aligners—particularly in terms of implicit assumptions and heuristics—can make it challenging to design a fair simulation.

3 Consistency Among Different Alignment Methods

The key idea behind consistency-based benchmarks is that different good aligners should tend to agree on a common alignment (namely, the correct one) whereas poor aligners might make different kinds of mistakes, thus resulting in inconsistent alignments. Confusingly, this notion of consistency among aligners is different from that of consistency-based aligning, which is an alignment strategy that favors MSAs consistent with pairwise alignments [24, 25]. In the context of benchmarking, the relevant notion is the former—referred to by Lassmann and Sonnhammer as "inter-consistency," cf. "intra-consistency" for the latter [26].

Practically, benchmarking by consistency among aligners can be implemented using measures such as the overlap score [26], a symmetric variant of sum-of-pairs. From a set of input alignments,

all paired aligned residues are determined over all sequences in every alignment. The overlap score for two alignments is calculated by counting the aligned pairs present in both alignments, and dividing by the average number of pairs in the alignments. Hence, two almost identical alignments have an overlap score close to one, while two very different alignments have an overlap score close to zero. Two additional scores based on this concept are the average overlap score, and the multiple overlap score. The average overlap score is simply the mean of the overlap scores measured over all pairs of input alignments, and represents the difficulty of the alignment problem. The multiple overlap score is a weighted sum of all pairs present in a single alignment, with the weight determined by the number of times each pair appears in the whole set of alignments. It is assumed that a high multiple overlap score, gained by an alignment with a high proportion of commonly observed pairs, corresponds to a good performance.

Another score that allows an internal control measure to estimate the consistency of different aligners is the heads-or-tails (HoT) score [27]. This consistency test is based on the assumption that biological sequences do not have a particular direction, and thus that alignments should be unaffected whether the input sequences are given in the original or reversed order. The agreement between the alignments obtained from the original and reversed sequences can be quantified with the overlap measures outlined above.

Both these consistency approaches—consistency among aligners and HoT score—are attractive because they assume no reference alignment or model of sequence of evolution, and thus can be readily and easily employed. Furthermore, high consistency is a necessary quality of a set of accurate aligners, thus making it desirable. The consistency criterion also appeals to the intuitive idea of "independent validation"—although most aligners have many aspects in common and are thus hardly "independent."

The biggest weakness of consistency is that it is no guarantee of correctness: methods can be *consistently wrong*. More subtly, consistency is sensitive to the choice of aligners in the set. This can be partly mitigated by including as many different alignments as possible [26]; nevertheless, it is easy to imagine cases where an accurate alignment, outnumbered by inaccurate, but similar, alignments, will be rated poorly. For instance, a new method solving a problem endemic to existing aligners will have low consistency scores.

Likewise, while low HoT scores can be indicative of considerable alignment uncertainty, the converse is not necessarily true. Hall reported that on simulated data at least, HoT scores tend to overestimate alignment accuracy [28]. That being said, considering the simplicity of HoT's scheme, the correlation Hall observed between HoT and simulation-based measures of alignment

accuracy is strikingly high (depending on methods, Pearson ρ of 87–98 %). It remains to be seen whether this will remain the case over time—new aligners might be tempted to exploit HoT's idea in their inference algorithms or parameter optimization procedures, thus compromising its independence as a benchmarking criterion. For instance, a trivial way of "gaming" the HoT score is to align sequences with "centre-justification" (adding a gap character in the middle of sequences of even-numbered length). Such obviously flawed alignment procedure is nevertheless insensitive to joint sequence reversals, consistently obtaining a perfect HoT score.

4 Structural Benchmarks

Benchmarks have also been developed starting from protein structure data. Structural benchmarks are by far the most widely adopted type [2]. Most commonly these employ the superposition of known protein structures as an independent means of alignment, to which alignments derived from sequence analysis can then be compared using the sum-of-pairs and true-column measures discussed earlier.

Structural benchmarks are naturally highly relevant when sequence alignments are sought to identify structural concordance among amino-acid residues. Yet they are also relevant to an evolutionary interpretation of alignments. Indeed, the biological observation that forms the basis of using structure in the latter context is that homologous proteins often retain structural similarity even when sequence divergence is large [29]. Thus, at high levels of divergence, a greater degree of confidence may be placed on alignments based on structural conservation than on sequence similarity. If residues from different proteins can be shown to overlap in three-dimensional space, it is likely (though not certain) that they are homologous. An important advantage of structural benchmarks is that they provide a truly independent, empirically derived standard to test different alignment algorithms.

A number of structurally derived benchmark datasets exist. One of the oldest is HOMSTRAD [10, 30]. Although not originally intended for benchmarking, this dataset has been extensively used to rate the quality of alignments. The first purpose-built, large-scale structural benchmark was BAliBASE [11, 31], which was based on similarity of known protein structures. It is divided into a number of datasets, each suited to test a different alignment problem—for example, greater or lesser sequence diversity, the presence of large insertions or extensions or the presence of repeated elements. Each BAliBASE dataset was constructed by accessing information in structural databases, and alignments were verified by hand, at both the level of individual residues and of overall secondary structure. Other purpose-built structural benchmarks include SABMARK [32] and PREFAB [33], which

differ from BAliBASE in that they are derived by automatic means, rather than by manual annotation of protein alignments. Reference sets also exist for RNA structures [34]. For further discussion of these datasets, we direct the reader to reviews by Aniba et al. [2], Edgar [3], Kim and Sinha [35], and Thompson et al. [4].

Regarding the desirable criterion of independence, although alignment algorithms incorporating structural aspects of sequence data do exist, such as Dynalign [36] and Foldalign [37]—for a more exhaustive discussion of RNA structural alignments, see Gardner et al. [34]—the parameters that go into constructing structure-based reference datasets are usually completely detached from the considerations that go into the development of MSA workflows.

Despite the degree of confidence structural alignment confers, it has been observed that sequence alignments used in BAliBASE and PREFAB are not always consistent with known annotations from external sources such as the CATH and SCOP databases, thus calling into question their biological accuracy [3]. Both manual and automated structural benchmark construction face considerable challenges. Manually curated structural benchmarks, while usually believed to generate more biologically accurate results than automated procedures, might also introduce subjective bias in the alignment. Automated procedures ensure reproducibility, but cannot avoid the existence of debatable parameter choices (e.g., the choice of the minimum spatial distance for two residues to be considered in the same fold) and potential systematic errors.

The nontrivial relationship between structural similarity of residues and alignment, however, is not the only cause of concern in structural benchmarks. Specifically, structure superpositions used for creating structural benchmarks are often not only based on experimentally derived structures, but also on primary sequence-based procedures such as BLASTP [38] and NORMD [39] which themselves employ amino acid substitution matrices and gap penalty scores, and thus make modelling assumptions about the sequences to be aligned [3]. If these parameters overlap with the parameters employed in MSA methods under test, then reference alignments obtained this way will be biased towards MSA-derived alignments that used those same parameters.

Problems arising from the use in benchmarking of reference alignments derived from structural comparisons can partially be overcome by the direct use of structural measures that are independent of any reference alignment. To evaluate the structure superposition implied by an MSA, Raghava et al. [40] adopted scores from a sequence-based multiple structure alignment algorithm [41]. Such structure similarity scores approximate the location of an amino acid in a test alignment by the location of its α-carbon (backbone carbon to which the amino acid side-chain attaches). Two aligned amino acid are then compared by the distance between

their chains of α-carbon atoms, estimated by least squares over translations and rotations of their respective 3D protein structures (which are known a priori). A simple score is given by the root-mean-square deviations between superposed α-carbon atoms, whereas a more refined score also takes into account the orientation of these atoms [48].

Two final aspects of structural benchmarks further complicate their application in MSA assessment. The fact that reliable annotations exist only for structurally conserved sequences means that MSA of any region of the genome other than structured protein coding regions—be it intronic, regulatory, natively disordered, or simply poorly annotated—cannot be effectively assessed using existing structural benchmarks [4, 35]. This is particularly important given that only a very small fraction of genome sequences encode globular, folded protein domains, and that both structural benchmarks and MSA tools focus mainly on alignment of this very small portion of sequences. The current state of sequencing technologies also means that sequence data come with many artifacts due to sequencing errors, short read length, and/or poor gene prediction models [4, 8, 42, 43] which are only very recently starting to be accounted for in benchmarks [4].

Considering all these complications, it becomes apparent that the map between structure and alignment is neither straightforward nor unequivocal. And indeed, by annotating known domains in reference datasets (or estimating superfamilies when the domain was unavailable), and then comparing annotation agreement in the reference alignments by use of column scores, Edgar found inconsistencies in the assignment of aligned residues to specific secondary structure in both PREFAB and BAliBASE [3].

5 Phylogenetic Tests of Alignment

Our last type of benchmark is phylogenetic tests of alignment. Dessimoz and Gil [44] have recently introduced such tests, developing an MSA assessment pipeline that explicitly takes into consideration phylogenetic relationships within the input sequence data to evaluate the validity of alignment hypotheses generated by different MSA methods.

This approach to benchmarking involves deriving alignments of the test data from different MSA packages as the starting point for tree building. The principle of the tests is simple: the more accurate the resulting tree, the more accurate the underlying alignment is assumed to be. The quality of the tree is measured by its compliance with an auxiliary principle or model; auxiliary in the sense that the additional knowledge introduced be independent of sequence data. So far, two methods have been devised. In the first, referred to as the “species tree discordance test,” test alignments are

built from putative orthologous sequences, so that the resulting test trees can be expected to have the same topology as the underlying species tree. Each resulting tree is then compared to a reference species tree, comprising sufficiently divergent species that its branching order is deemed uncontroversial. The best performing aligners are taken to be those that most consistently generate alignments that yield test trees congruent with the species tree. Indeed, it can be expected that averaged over many hundreds or thousands of families, discordance due to non-orthology among the input sequences will affect the performance of all aligners equally, whereas discordance due to alignment error will vary among aligners.

The second method, termed the "minimum duplication" invokes a parsimony argument to interpret trees built from alignments of both orthologous and paralogous sequences, favoring trees which require fewer gene duplications to explain the data as more likely to reflect the true evolutionary history of the sequences.

One key advantage of phylogenetic benchmarks is that they provide a way of evaluating gap-rich and variable regions, regions for which structural benchmarks are often not applicable and simulation benchmarks lack realism [44]. In particular, the limited applicability of structural benchmarks to conserved protein core regions has quite possibly caused developers of alignment methods to focus their efforts on improving the performance of their tools on conserved regions at the expense of gap-rich or variable regions. Yet focusing on conserved regions can result in a loss of potentially informative data for multiple sequence alignment [21]. Adopting a simple tree inference method that looks only at presence or absence of gaps as a binary character within a maximum parsimony framework, Dessimoz and Gil reported that gap-only trees are sometimes even more accurate than nucleotide-based trees, thus highlighting the signal lost in neglecting variable or gap-rich regions [44].

At present, phylogeny-based benchmarks are the only ones that can be interpreted to be directly evaluating homology on real data. The premise of this interpretation is that more accurate trees on average necessarily ensue from a higher proportion of homologous positions in alignments on average, and therefore that the former is a good surrogate for the latter. Yet although we view the premise as highly plausible (and indeed fail to see how one could argue the opposite), there is no proof for it. If dismissed altogether, the interpretation has to be weakened so that these phylogeny tests only measure the effect of alignment on phylogenetic inference. In this case, phylogeny-based benchmarks are less meaningful even for other homology-based applications of alignments, such as detecting sites under positive selection [45].

6 Conclusions

Benchmarks for MSA applications have arisen in recent years as a crucial tool for bioinformaticians to keep a critical eye on existing software packages and reliably diagnose areas that need further development. The implementation of benchmarks to routinely assess the efficacy and accuracy of MSA methods has clearly provided important insights, and has pointed out to the developer community very serious shortcomings of existing methods that would not otherwise have been so apparent [4, 26, 44, 46]. Each benchmarking solution examined in this chapter—whether simulation-, consistency-, structure-, or phylogeny-based—entails risks of bias and error, but each is also useful in its own right when applied to a relevant problem. It is interesting to note that simulation benchmarks rank MSA methods differently from empirical benchmarks [21, 46, 47]. It is clear that no single benchmark can be uniformly used to test different MSA methods. Instead, due to both the computational and biological issues raised by the problem of sequence alignment optimization, a multiplicity of scenarios need to be modelled in benchmark datasets.

A telling symptom of the current state of affairs is the fact that subjective manual editing of sequence alignments remains widespread, reflecting perhaps an overall lack of confidence in the performance of automated multiple alignment strategies. The criteria used when editing sequence alignments "by eye" are vague and may introduce individual biases and aesthetic considerations into sequence alignment [9, 21].

In order to ensure reproducibility of experimental results, one of the most important goals of scientific practice, this trend needs to change. Context-specific, effective benchmarking with well-defined objectives represents a sensible way forward.

Acknowledgments

The authors thank Julie Thompson for helpful feedback on the manuscript. CD is supported by SNSF advanced researcher fellowship #136461. This article started as assignment for the graduate course "Reviews in Computational Biology" at the Cambridge Computational Biology Institute, University of Cambridge.

References

1. Kemena C, Notredame C (2009) Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinformatics 25(19):2455–2465
2. Aniba MR, Poch O, Thompson JD (2010) Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. Nucleic Acids Res 38(21):7353–7363

3. Edgar RC (2010) Quality measures for protein alignment benchmarks. Nucleic Acids Res 38(7):2145–2153
4. Thompson JD, Linard B, Lecompte O, Poch O (2011) A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. PLoS One 6(3):e18093
5. Löytynoja A (2012) Alignment methods: strategies, challenges, benchmarking, and comparative overview. Methods Mol Biol 855:203–235
6. Thompson JD, Higgins DG, Gibson TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res 22(22):4673–4680
7. Morrison DA (2009) Why would phylogeneticists ignore computerized sequence alignment? Syst Biol 58(1):150–158
8. Mardis ER (2008) The impact of next-generation sequencing technology on genetics. Trends Genet 24(3):133–141. doi:10.1016/j.tig.2007.12.007
9. Anisimova M, Cannarozzi G, Liberles D (2010) Finding the balance between the mathematical and biological optima in multiple sequence alignment. Trends Evol Biol 2(1):e7
10. Stebbings LA, Mizuguchi K (2004) HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. Nucleic Acids Res 32(Database issue): D203–D207
11. Thompson JD, Koehl P, Ripp R, Poch O (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins 61:127–136
12. Stoye J, Evers D, Meyer F (1998) Rose: generating sequence families. Bioinformatics 14(2):157–163
13. Cartwright RA (2005) DNA assembly with gaps (Dawg): simulating sequence evolution. Bioinformatics 21(Suppl 3):iii31–iii38
14. Hall BG (2008) Simulating DNA coding sequence evolution with EvolveAGene 3. Mol Biol Evol 25(4):688–695
15. Fletcher W, Yang Z (2009) INDELible: a flexible simulator of biological sequence evolution. Mol Biol Evol 26(8):1879–1888
16. Sipos B, Massingham T, Jordan GE, Goldman N (2011) PhyloSim – Monte Carlo simulation of sequence evolution in the R statistical computing environment. BMC Bioinformatics 12(1):104
17. Koestler T, Av H, Ebersberger I (2012) REvolver: modeling sequence evolution under domain constraints. Mol Biol Evol 29(9): 2133–2145
18. Dalquen DA, Anisimova M, Gonnet GH, Dessimoz C (2012) ALF-a simulation framework for genome evolution. Mol Biol Evol 29(4): 1115–1123
19. Thompson JD, Plewniak F, Poch O (1999) A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res 27(13):2682–2690, gkc432 [pii]
20. Blackburne BP, Whelan S (2012) Measuring the distance between multiple sequence alignments. Bioinformatics 28(4):495–502. doi:10.1093/bioinformatics/btr701
21. Löytynoja A, Goldman N (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. Science 320(5883):1632–1635. doi:10.1126/science.1158395
22. Golubchik T, Wise MJ, Easteal S, Jermiin LS (2007) Mind the gaps: evidence of bias in estimates of multiple sequence alignments. Mol Biol Evol 24(11):2433–2442
23. Huelsenbeck JP (1995) Performance of phylogenetic methods in simulation. Syst Biol 44(1): 17–48
24. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. Genome Res 15(2):330–340. doi:10.1101/gr.2821705
25. Notredame C, Higgins DG, Heringa J (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol 302 (1):205–217. doi:10.1006/jmbi.2000.4042
26. Lassmann T, Sonnhammer ELL (2005) Automatic assessment of alignment quality. Nucleic Acids Res 33(22):7120–7128
27. Landan G, Graur D (2007) Heads or tails: a simple reliability check for multiple sequence alignments. Mol Biol Evol 24(6):1380–1383
28. Hall BG (2008) How well does the HoT score reflect sequence alignment accuracy? Mol Biol Evol 25(8):1576–1580
29. Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. EMBO J 5(4):823
30. Mizuguchi K, Deane CM, Blundell TL, Overington JP (1998) HOMSTRAD: a database of protein structure alignments for homologous families. Protein Sci 7(11):2469–2471. doi:10.1002/pro.5560071126
31. Thompson JD, Plewniak F, Poch O (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics 15(1):87–88, btc017 [pii]

32. Van Walle I, Lasters I, Wyns L (2005) SABmark – a benchmark for sequence alignment that covers the entire known fold space. Bioinformatics 21(7):1267–1268. doi:10.1093/bioinformatics/bth493
33. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 32(5):1792–1797. doi:10.1093/nar/gkh340
34. Gardner P, Wilm A, Washietl S (2005) A benchmark of multiple sequence alignment programs upon structural RNAs. Nucleic Acids Res 33(8):2433–2439
35. Kim J, Sinha S (2010) Towards realistic benchmarks for multiple alignments of non-coding sequences. BMC Bioinformatics 11:54
36. Mathews DH (2005) Predicting a set of minimal free energy RNA secondary structures common to two sequences. Bioinformatics 21(10):2246–2253. doi:10.1093/bioinformatics/bti349
37. Havgaard JH, Lyngso RB, Stormo GD, Gorodkin J (2005) Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. Bioinformatics 21(9):1815–1824. doi:10.1093/bioinformatics/bti279
38. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410
39. Thompson JD, Fdr P, Ripp R, Thierry J-C, Poch O (2001) Towards a reliable objective function for multiple sequence alignments1. J Mol Biol 314(4):937–951. doi:10.1006/jmbi.2001.5187
40. Raghava GP, Searle SM, Audley PC, Barber JD, Barton GJ (2003) OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinformatics 4:47. doi:10.1186/1471-2105-4-47
41. Russell RB, Barton GJ (1992) Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. Proteins 14(2):309–323. doi:10.1002/prot.340140216
42. Pop M, Salzberg SL (2008) Bioinformatics challenges of new sequencing technology. Trends Genet 24(3):142–149. doi:10.1016/j.tig.2007.12.006
43. Berger SA, Stamatakis A (2011) Aligning short reads to reference alignments and trees. Bioinformatics 27(15):2068–2075. doi:10.1093/bioinformatics/btr320
44. Dessimoz C, Gil M (2010) Phylogenetic assessment of alignments reveals neglected tree signal in gaps. Genome Biol 11(4):R37
45. Jordan G, Goldman N (2011) The effects of alignment error and alignment filtering on the sitewise detection of positive selection. Mol Biol Evol 29:1125. doi:10.1093/molbev/msr272
46. Blackshields G, Wallace IM, Larkin M, Higgins DG (2006) Analysis and comparison of benchmarks for multiple sequence alignment. In Silico Biol 6(4):321–339
47. Lassmann T, Sonnhammer EL (2002) Quality assessment of multiple alignment programs. FEBS Lett 529(1):126–130, S0014579302031897 [pii]
48. Strope CL, Abel K, Scott SD, Moriyama EN (2009) Biological sequence simulation for testing complex evolutionary hypotheses: indel-Seq-Gen version 2.0. Mol Biol Evol 26(11):2581–2593. doi:10.1093/molbev/msp174
49. Lassmann T, Sonnhammer EL (2006) Kalign, Kalignvu and Mumsa: web servers for multiple sequence alignment. Nucleic Acids Res 34 (Web Server issue):W596–W599. doi:10.1093/nar/gkl191
50. Kemena C, Taly JF, Kleinjung J, Notredame C (2011) STRIKE: evaluation of protein MSAs using a single 3D structure. Bioinformatics 27(24):3385–3391. doi:10.1093/bioinformatics/btr587

B

THE DRAFT GENOME SEQUENCE OF THE FERRET (*MUSTELA PUTORIUS FURO*) FACILITATES STUDY OF HUMAN RESPIRATORY DISEASE

The ferret genome paper is reproduced in this appendix. My contribution to this paper is described in chapter 5. The phylogenetic tree shown I produced appeared in the supplementary materials only, so is not reproduced in this appendix, instead it appears as figure 5.2.

nature biotechnology

OPEN

The draft genome sequence of the ferret (*Mustela putorius furo*) facilitates study of human respiratory disease

Xinxia Peng[1], Jessica Alföldi[2], Kevin Gori[3], Amie J Eisfeld[4], Scott R Tyler[5,6], Jennifer Tisoncik-Go[1], David Brawand[2,22], G Lynn Law[1], Nives Skunca[7,8], Masato Hatta[4], David J Gasper[4], Sara M Kelly[1], Jean Chang[1], Matthew J Thomas[1], Jeremy Johnson[2], Aaron M Berlin[2], Marcia Lara[2,22], Pamela Russell[2,22], Ross Swofford[2], Jason Turner-Maier[2], Sarah Young[2], Thibaut Hourlier[9], Bronwen Aken[9], Steve Searle[9], Xingshen Sun[5], Yaling Yi[5], M Suresh[4], Terrence M Tumpey[10], Adam Siepel[11], Samantha M Wisely[12], Christophe Dessimoz[3,13,14], Yoshihiro Kawaoka[4,15–18], Bruce W Birren[2], Kerstin Lindblad-Toh[2,19], Federica Di Palma[2,22], John F Engelhardt[5,20], Robert E Palermo[1] & Michael G Katze[1,21]

The domestic ferret (*Mustela putorius furo*) is an important animal model for multiple human respiratory diseases. It is considered the 'gold standard' for modeling human influenza virus infection and transmission[1–4]. Here we describe the 2.41 Gb draft genome assembly of the domestic ferret, constituting 2.28 Gb of sequence plus gaps. We annotated 19,910 protein-coding genes on this assembly using RNA-seq data from 21 ferret tissues. We characterized the ferret host response to two influenza virus infections by RNA-seq analysis of 42 ferret samples from influenza time-course data and showed distinct signatures in ferret trachea and lung tissues specific to 1918 or 2009 human pandemic influenza virus infections. Using microarray data from 16 ferret samples reflecting cystic fibrosis disease progression, we showed that transcriptional changes in the CFTR-knockout ferret lung reflect pathways of early disease that cannot be readily studied in human infants with cystic fibrosis disease.

We performed whole-genome sequencing with DNA from an individual female sable ferret (*M. putorius furo*) and created a genome assembly using ALLPATHS-LG[5]. The draft assembly is 2.41 Gb including gaps, has a contig N50 size of 44.8 kb, a scaffold N50 size of 9.3 Mb and quality metrics comparable to other genomes sequenced using Illumina technology (**Table 1** and **Supplementary Note**). RNA-seq data for annotation were obtained from polyadenylated transcripts using RNA from 24 samples of 21 tissues from male and female ferrets, including both developmental and adult tissues (**Supplementary Table 1**). The genome assembly was annotated using the Ensembl gene annotation system[6] (Ensembl release 70). Protein-coding gene models were annotated by combining alignments of Uniprot[7] mammal and other vertebrate protein sequences and the aforementioned RNA-seq data. The ferret genome can be viewed as unanchored scaffolds along with the Ensembl genome models in both the University of California Santa Cruz and Ensembl genome browser interfaces. We also used the tool liftOver to map the coordinates from the ferret assembly onto the well-finished genome sequence of its phylogenetic neighbor, the domestic dog (*Canis familiaris*; V3.1); this mapping is a useful resource for a genome based on short-read sequencing and facilitates browsing the ferret genome in the surrogate context of dog chromosomes (**Supplementary Fig. 1**).

Using the annotated ferret protein sequences, we constructed a highly resolved phylogenetic tree (**Supplementary Fig. 2**). As expected,

[1]Department of Microbiology, University of Washington, Seattle, Washington, USA. [2]Broad Institute of MIT and Harvard, Cambridge, Massachusetts, USA. [3]European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, Cambridge, UK. [4]Department of Pathobiological Sciences, School of Veterinary Medicine, University of Wisconsin, Madison, Wisconsin, USA. [5]Department of Anatomy and Cell Biology, Carver College of Medicine, University of Iowa, Iowa City, Iowa, USA. [6]Molecular and Cellular Biology Program, Carver College of Medicine, University of Iowa, Iowa City, Iowa, USA. [7]Department of Computer Science, Swiss Federal Institute of Technology (ETH Zurich), Zurich, Switzerland. [8]Swiss Institute of Bioinformatics, Zurich, Switzerland. [9]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK. [10]Centers for Disease Control and Prevention, Atlanta, Georgia, USA. [11]Department of Biological Statistics and Computational Biology, Cornell University, Ithaca, New York, USA. [12]Department of Wildlife Ecology and Conservation, University of Florida, Gainesville, Florida, USA. [13]Department of Genetics, Evolution and Environment, University College London, London, UK. [14]Department of Computer Science, University College London, London, UK. [15]ERATO Infection-Induced Host Responses Project, Japan Science and Technology Agency, Saitama, Japan. [16]Division of Virology, Department of Microbiology and Immunology, Institute of Medical Science, University of Tokyo, Tokyo, Japan. [17]Department of Special Pathogens, International Research Center for Infectious Diseases, Institute of Medical Science, University of Tokyo, Minato-ku, Tokyo, Japan. [18]Laboratory of Bioresponses Regulation, Department of Biological Responses, Institute for Virus Research, Kyoto University, Kyoto, Japan. [19]Science for Life Laboratory, Department of Medical Biochemistry and Microbiology, Uppsala University, Uppsala, Sweden. [20]Center for Gene Therapy, Carver College of Medicine, University of Iowa, Iowa City, Iowa, USA. [21]Washington National Primate Research Center, Seattle, Washington, USA. [22]Present addresses: MRC Functional Genomics Unit, University of Oxford, Oxford, UK (D.B.); Biogen Idec, Cambridge, Massachusetts, USA (M.L.); Division of Biology, California Institute of Technology, Pasadena, California, USA (P.R.); Vertebrate and Health Genomics, The Genome Analysis Center, Norwich, UK (F.D.P.). Correspondence should be addressed to M.G.K. (honey@uw.edu) or R.E.P. (palermor@uw.edu).

Received 20 May; accepted 22 October; published online 17 November 2014; doi:10.1038/nbt.3079

Table 1 Summary details of the ferret genome assembly and associated Ensembl annotation

Mustela putorius furo genome	
Assembly	MusPutFur1.0
Date	June 2011
Total assembly length	2.41 Gb
Total sequence	2.28 Gb
Short read coverage	162 ×
Organization	7,783 unplaced scaffolds
Scaffold N50	9.3 Mb
Ensembl annotation	
Database version	70.1
Date	August 2012
Coding genes	19,910
Noncoding genes	3,614
Pseudogenes	287
Gene transcripts	23,963

Supporting evidence: 1.1×10^9 mRNA-seq reads (Paired end 2 × 100; 220 Gb) from 21 individual tissues including developmental stages, and respiratory tissues from 2009 SOIV infected ferrets.

ferret falls within the *Caniformia* suborder of the Carnivores, as represented by the domestic dog, cat, giant panda and walrus, and the support values are high for most clades (**Supplementary Tables 2** and **3** and Methods). Although the clade containing the ferret diverged from a common ancestor before the divergence of the rodent and human/primate lineages, branch lengths in the tree indicate rapid evolution in the rodent clade, which has resulted in less genetic divergence between humans and ferrets than between humans and mice. Indeed, in comparing protein sequences between the species, we found that for 75% of all orthologous triplets, ferret proteins are closer than mouse proteins to human proteins (**Fig. 1a** and **Supplementary Table 4**). For example, the ferret cystic fibrosis transmembrane conductance regulator (CFTR) protein is considerably closer to the human protein than is its mouse counterpart (percentage identities [PAM distance] for ferret to human = 92% [8.1]; mouse to human = 79% [23.3]). Overall, Gene Ontology (GO) terms related to basic cell physiology tend to be enriched among the genes residing in the angular sector, representing the top 25% of genes for which the ferret sequence is closer to human than the mouse ortholog. The enriched GO terms include nucleic acid metabolism, nuclear division, regulation of expression, and protein modification and localization (**Supplementary Fig. 3** and **Supplementary Tables 5** and **6**). Extending this comparison from CFTR to 106 CFTR-interacting proteins, we found that the ferret-to-human protein sequence similarity is significantly greater than the corresponding mouse ortholog (Wilcoxon test P value = 3.1×10^{-6}; **Fig. 1b**). In additional comparisons, we examined gene sets pertinent to cystic fibrosis disease processes including inflammation, lung and pancreatic remodeling, and the regulation of insulin and diabetes; in all cases we found the encoded human proteins to be better conserved in ferret than in mouse (**Fig. 1b** and **Supplementary Fig. 4**). In contrast, proteins encoded by some nervous system–related genes seem to be more divergent from human in ferret than in mouse (**Fig. 1b**). In summary, the overall high sequence similarity between ferret and human proteins shown by these genome-level analyses indicates that many ferret proteins have likely evolved to conserve similar molecular functions as their human protein orthologs.

Next, we investigated whether ferret and human genes exhibit similar tissue expression. We compared the patterns of relative transcript abundance across seven tissues in common between our data set and previously reported human RNA-seq data[8] (**Fig. 1c**). First, we determined the genes with highest relative abundance across all tissues within each species and found that the intersection of these tissue-specific sets between human and ferret was highly significant (chi-squared test P values < 10^{-186}; **Supplementary Table 7**). To refine the sets of tissue-specific genes, we clustered genes with similar expression patterns across the 14 tissue samples (7 from ferret and 7 from human) into 7 disjoint clusters (**Fig. 1c** and **Supplementary Table 8**). This clustering analysis revealed that many ferret and human genes exhibited highly concordant, tissue-specific expression patterns. The assignment of a gene cluster to a specific tissue was evident by its significantly increased expression in that tissue relative to the rest of the tissues of the same species for all comparisons except between skeletal muscle and heart (**Supplementary Table 9**). The similarity between skeletal muscle and heart may be attributed to the presence of striated muscle cells in both tissues. The clusters include transcription factors (TFs) with known tissue specificities in human samples, such as *OLIG2* and *NEUROD2* (brain), *OVOL1* (testis), *MYF6* (heart, skeletal muscle), and lung-specific Iroquois-class homeodomain TFs *IRX2*, *IRX3* and *IRX5*[9]. The same specificity was seen for the ferret tissues. Sequence comparisons showed that ferret TFs in brain, skeletal muscle/heart, lung and kidney gene clusters exhibited even greater similarity to human orthologs than the rest of the ferret genome, suggesting strong conservation of functional regulation (**Supplementary Tables 10** and **11**). Some genes related to immune and inflammatory functions, including TFs associated with Th17 cells (*BATF*, *IRF4*, *AHR*)[10], showed increased expression in ferret and human lungs, which is likely a consequence of the greater proportion of immune cells in this compartment and the possible presence of bronchus-associated lymphoid tissue. The broad similarity between ferret and human tissue-specific gene expression suggests the regulation of gene expression in tissue compartments is also highly conserved.

Ferrets are frequently used as a model for human influenza virus infection, in part due to the similar distribution of viral attachment receptors in the respiratory tract of humans and ferrets[2,11]. We used our genome sequence to profile the transcriptional response of ferrets to pandemic influenza virus. To this end, we infected ferrets with either of two human pandemic influenza viruses—the H1N1 2009 pandemic virus A/CA/04/2009 (CA04) or the reconstructed H1N1 1918 pandemic virus (1918)—and collected samples from both the upper (trachea) and the lower (lung) respiratory tract at 1, 3 and 8 days postinfection (dpi) for transcriptome analysis (**Supplementary Table 12** and **Supplementary Figs. 5** and **6**). To increase the coverage of our transcriptome analysis beyond standard Ensembl annotated genes, including non-polyadenylated transcripts, we performed RNA-seq on total RNA after ribosomal RNA depletion (Methods). To augment standard Ensembl annotation, we predicted additional transcript models using the RNA-seq data collected from both lung and trachea samples from these infected animals and the tissue samples described in the previous paragraph (Online Methods and **Supplementary Table 13**). Additional analyses indicate that the transcripts derived by RNA-seq are enriched with novel protein-coding isoforms and polyadenylated and non-polyadenylated intergenic noncoding RNAs (**Supplementary Figs. 7** and **8**). To make these genomic resources more accessible for gene expression profiling, we also designed and validated two versions of ferret-specific oligonucleotide microarrays: version 1 interrogates 23,582 Ensembl annotated transcripts plus 13,368 intergenic transcripts derived from RNA-seq analysis of ferret mRNAs; version 2 provides broader coverage with probes for an additional 27,288 intergenic transcripts from RNA-seq analysis of ferret total RNAs (**Supplementary Table 14**, **Supplementary Figs. 9–16** and **Supplementary Note**).

As quantified by RNA-seq, host transcriptional changes were much more extensive in infected trachea (9,869 differentially expressed [DE]

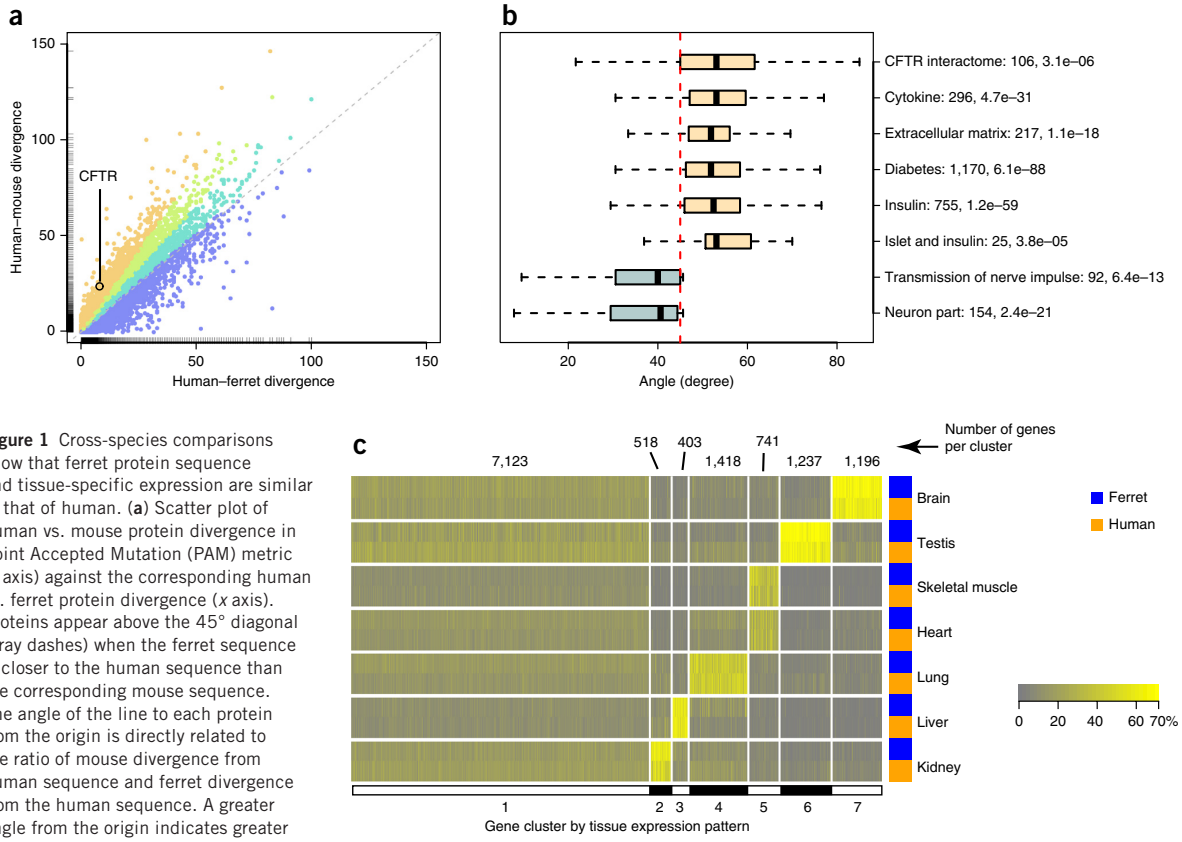
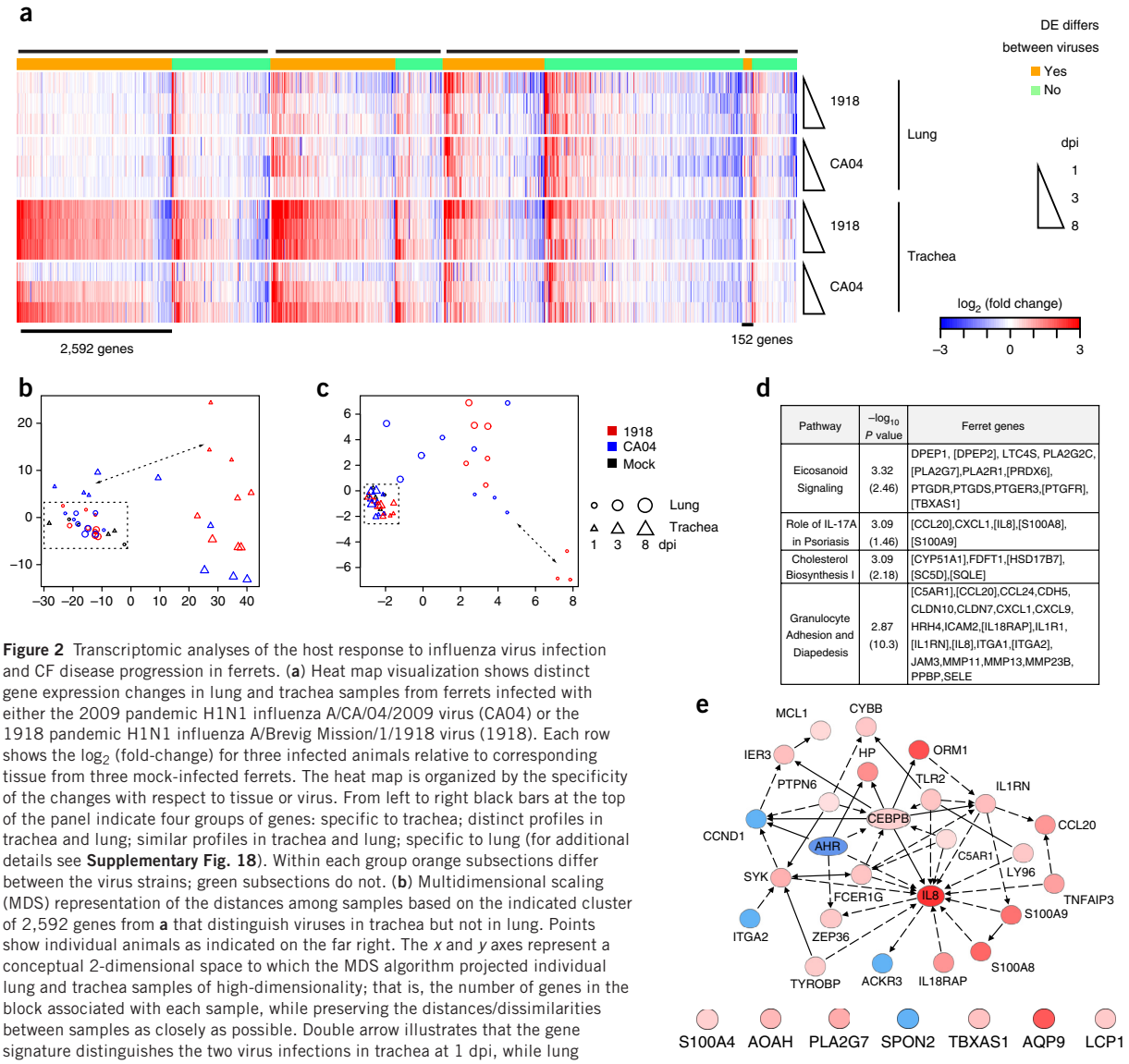


Figure 1 Cross-species comparisons show that ferret protein sequence and tissue-specific expression are similar to that of human. (**a**) Scatter plot of human vs. mouse protein divergence in Point Accepted Mutation (PAM) metric (*y* axis) against the corresponding human vs. ferret protein divergence (*x* axis). Proteins appear above the 45° diagonal (gray dashes) when the ferret sequence is closer to the human sequence than the corresponding mouse sequence. The angle of the line to each protein from the origin is directly related to the ratio of mouse divergence from human sequence and ferret divergence from the human sequence. A greater angle from the origin indicates greater divergence. The quartiles of the distribution of these ratios are displayed in different colors (blue being the least conserved in ferret relative to mouse, and orange-brown being the most conserved). Hatched lines on the axes show the metric distributions for the individual species (**Supplementary Table 4**). (**b**) Boxplots of the angles represented in **a** for proteins in eight selected biological functions. For gene sets related to CF (light yellow), human protein sequence is better conserved in ferret than in mouse. The gene sets highlighted in light yellow are relevant to CFTR function, lung inflammation and remodeling, and insulin secretory defects in CF-related diabetes. For two nervous system–related gene sets (light blue; neuron part (GO:0097458) and transmission of nerve impulse (GO:0019226)), human protein sequence tended to be more conserved in mouse. Next to each function is the number of proteins in the function and the *P* value from one-sided Wilcoxon signed rank test comparing the human-ferret (*x* axis in **a**) vs. human-mouse (*y* axis in **a**) divergence in PAM metric. (**c**) K-means clustering of ferret-human orthologous genes by their tissue expression patterns reveals similarities in tissue specificity. The color scale represents relative abundance across all tissues within each species and is saturated at 70%. Vertical partitions correspond to the seven clusters of genes from the optimal clustering, with numbers of genes per cluster appearing on the top. Horizontal groupings are organized by tissue with ferret and human pairings denoted by the color bar at the side, and highlight the tissue specificity of clusters 2 through 7.

genes, adjusted *P* value < 0.01) than in infected lung (4,646 DE genes), and the kinetics of the response differed by virus and compartment (**Supplementary Fig. 17**). In the trachea, the 1918 virus induced a pronounced transcriptional response, both in the number of DE genes and in the magnitude of their changes, that commenced at 1 dpi and was largely sustained through day 8; in contrast, infection with the CA04 virus resulted in a gradual escalation of overall transcriptional changes in these same genes, resulting in peak expression by 8 dpi. Different kinetics occurred in the lung, where both viruses induced a similar number of DE genes at 1 dpi, followed by a decline to far fewer DE genes by day 8. A detailed tissue-by-virus comparison revealed distinct transcriptional signatures that differentiated the response to the 1918 and CA04 viruses in the two respiratory tissues (**Fig. 2a**, **Supplementary Fig. 18** and **Supplementary Table 15**). Within the trachea-specific host response, a subset of 2,592 ferret genes distinguished the two viruses, with extensive perturbation at 1 dpi in response to the 1918 virus and minimal alteration in response to CA04 (**Fig. 2b**). This gene set has an over-representation of diverse biological processes such as Apoptosis Signaling, NGF Signaling and Ceramide Signaling (one-sided Fisher exact test *P* values of 4.28×10^{-7}, 1.61×10^{-6} and 3.76×10^{-6}, respectively; **Supplementary Table 16**). Related lipid-receptor signaling systems, such as sphingosine-1-phosphate receptor signaling, can protect the host from influenza virus-induced "cytokine storm" by inhibiting pro-inflammatory responses[12,13]. Similarly, some DE transcripts were exclusively observed within the lung compartment, with a subset of 152 ferret genes that differentiated the two virus infections (**Fig. 2c**). Within this subset, we observed enrichment of Prothrombin Activation Pathway and differential expression of *Il13* and *Il20*, associated with Role of Cytokines in Mediating Communication between Immune Cells (*P* value of 1.53×10^{-2}), which are produced by pulmonary innate lymphoid cells[14] and maturing dendritic cells[15], respectively. In summary, the ferret genomic resources described here enabled a side-by-side comparison of ferret transcriptional responses to two human pandemic influenza viruses. The results revealed that the host response to the two pandemic viruses differs in a tissue compartment–dependent manner.



Pathway	$-\log_{10}$ *P* value	Ferret genes
Eicosanoid Signaling	3.32 (2.46)	DPEP1, [DPEP2], LTC4S, PLA2G2C, [PLA2G7],PLA2R1,[PRDX6], PTGDR,PTGDS,PTGER3,[PTGFR], [TBXAS1]
Role of IL-17A in Psoriasis	3.09 (1.46)	[CCL20],CXCL1,[IL8],[S100A8], [S100A9]
Cholesterol Biosynthesis I	3.09 (2.18)	[CYP51A1],FDFT1,[HSD17B7], [SC5D],[SQLE]
Granulocyte Adhesion and Diapedesis	2.87 (10.3)	[C5AR1],[CCL20],CCL24,CDH5, CLDN10,CLDN7,CXCL1,CXCL9, HRH4,ICAM2,[IL18RAP],IL1R1, [IL1RN],[IL8],ITGA1,[ITGA2], JAM3,MMP11,MMP13,MMP23B, PPBP,SELE

Figure 2 Transcriptomic analyses of the host response to influenza virus infection and CF disease progression in ferrets. (**a**) Heat map visualization shows distinct gene expression changes in lung and trachea samples from ferrets infected with either the 2009 pandemic H1N1 influenza A/CA/04/2009 virus (CA04) or the 1918 pandemic H1N1 influenza A/Brevig Mission/1/1918 virus (1918). Each row shows the $\log_2$ (fold-change) for three infected animals relative to corresponding tissue from three mock-infected ferrets. The heat map is organized by the specificity of the changes with respect to tissue or virus. From left to right black bars at the top of the panel indicate four groups of genes: specific to trachea; distinct profiles in trachea and lung; similar profiles in trachea and lung; specific to lung (for additional details see **Supplementary Fig. 18**). Within each group orange subsections differ between the virus strains; green subsections do not. (**b**) Multidimensional scaling (MDS) representation of the distances among samples based on the indicated cluster of 2,592 genes from **a** that distinguish viruses in trachea but not in lung. Points show individual animals as indicated on the far right. The *x* and *y* axes represent a conceptual 2-dimensional space to which the MDS algorithm projected individual lung and trachea samples of high-dimensionality; that is, the number of genes in the block associated with each sample, while preserving the distances/dissimilarities between samples as closely as possible. Double arrow illustrates that the gene signature distinguishes the two virus infections in trachea at 1 dpi, while lung samples show no separation (dotted rectangle). (**c**) As in **b**, for the indicated cluster of 152 genes that is differentially regulated in lung but not trachea tissues and separates the two virus strains on 1 dpi. (**d**,**e**) Differential transcriptional responses in an experiment comparing lung samples from 15-day-old CF ferrets ($n = 3$) vs. non-CF ferrets ($n = 5$). (**d**) Similar pathways enriched in genes differentially expressed in 15-day-old CF ferret lung samples and CF human bronchial brushings, derived from Ingenuity Pathway Analysis. The values in parentheses are the enrichment *P* values for the corresponding pathways in the genes differentially expressed in CF human bronchial brushing[19]. In brackets are genes that were differentially expressed in both ferret and human CF/non-CF comparisons. (**e**) Network illustration of 32 genes of the function "inflammatory response", which were differentially expressed in the same direction in ferret and human CF data sets (for additional details see **Supplementary Fig. 20**). Red and blue shading reflects the extent of increased or decreased expression, respectively, in CF relative to non-CF individuals. A solid line between two genes indicates direct interaction(s) among them and a dotted line for indirect interaction(s), as documented in the literature.

Genetically engineered cystic fibrosis ferrets model two key components of disease not observed in cystic fibrosis mice, lung disease[16,17] and diabetes[18]. To investigate cystic fibrosis disease progression in the ferret model, we carried out microarray expression analysis on lung specimens from newborn and 15-day-old CFTR knockout (CF) and normal (non-CF) ferrets. In newborn animals, genotypic differences in transcriptomes were limited; 472 DE protein-coding genes were identified using a relaxed threshold (absolute fold-change ≥ 1.4, *P* value ≤ 0.1). Nonetheless, functional analyses of these DE genes showed disturbances in several canonical pathways, including Coagulation System, Primary Immunodeficiency Signaling, Serotonin Receptor Signaling and Signaling in T Helper Cells (**Supplementary Table 17**). Genotype-dependent gene expression differences between 15-day-old animals were much more extensive (1,468 DE protein-coding genes, absolute fold-change ≥ 1.5, false discovery rate ≤ 0.05) and included expression changes in genes from

pathways involved in Cholesterol Biosynthesis, Eicosanoid Signaling, Granulocyte Adhesion and Diapedesis, and IL8 regulation (**Fig. 2d**,**e** and **Supplementary Table 17**). In a previous study, gene expression in these pathways was also significantly perturbed in human bronchial brushings from adults with cystic fibrosis[19] (**Supplementary Table 17**). Further, in CF ferrets, changes in the expression of most of these genes were highly positively correlated (ANOVA P value 4.9×10^{-5}, Pearson correlation coefficient 0.63) with that in human cystic fibrosis samples, consistent with the overall positively correlated changes in expression between day 15 CF ferret and human cystic fibrosis samples (**Supplementary Fig. 19**). The exception of some cholesterol biosynthesis pathway genes may be the result of variation in epithelia sampled in the ferret (intact lung) and human (conducting larger airways), or differences in cystic fibrosis disease status between infants and adults.

Similar expression changes in the CF ferret and human cystic fibrosis data sets were also evident at the level of broader biological functions such as Chronic Inflammatory Disorder, Cell Movement of Phagocytes and Inflammatory Response (**Supplementary Table 18**). As anticipated, *IL8* was one of the most significantly increased inflammatory genes in older CF ferret (14.6-fold upregulated) and human cystic fibrosis (11.7-fold upregulated) samples (**Supplementary Table 17**), consistent with a dominant role of *IL8* in cystic fibrosis lung disease[20]. Indeed, many genes associated with *IL8* regulation, including *CCL20*, *S100A8*, *S100A9*, *IL18RAP*, *IL1RN* and *ITGA2*, were differentially regulated concordantly in CF ferret and human cystic fibrosis lung samples (**Fig. 2e** and **Supplementary Fig. 20**). Although these findings suggest commonalities in the pathways of cystic fibrosis inflammation between the two species, it is worth noting that the dominant bacterial pathogens of the lung are distinctly different between CF ferret and humans with this disease—*Pseudomonas aeruginosa*[21] in humans and enteric pathogens in both young and old CF ferrets[16,17]. Thus, the predominant gene pathways involved in cystic fibrosis inflammatory responses seem to be conserved across ferrets and humans, and to be largely independent of the pathogen's taxa. Of the DE genes for which expression changed in opposite direction between ferret and human data sets, one of the most significant functional pathways included 19 genes associated with Cell Movement (**Supplementary Fig. 20**). This suggests that there are differences in the extent of injury, repair and/or migratory inflammatory cell infiltrates between the ferret and human data sets. Such differences are not surprising given the larger number of DE genes associated with Granulocyte Adhesion and Diapedesis (**Supplementary Table 17**) and inflammation (**Supplementary Table 18**) in the older human cystic fibrosis samples. Despite these differences, the overall positively correlated expression changes, especially the high concordance in key cystic fibrosis-related pathways and functions between 15-day-old CF ferret and adult human cystic fibrosis samples, suggest that many disease changes associated with adult cystic fibrosis in humans may begin in infancy. Thus, the CF ferret represents a tractable model by which to systemically address disease progression–related changes in gene expression at anatomical sites not possible in humans.

Ferrets are extensively used to study human diseases such as influenza virus infection and cystic fibrosis, but the lack of genome sequence information has limited the ability to understand ferret transcriptional responses. Our transcriptomic analyses of the host response to human pandemic influenza virus infection and of cystic fibrosis disease progression in ferrets illustrate how the availability of the ferret genome sequence can enhance the sophistication of ferret respiratory-disease models. The analyses revealed high protein-sequence similarity and shared tissue-expression patterns between ferret and human, suggesting the potential utility of ferret models in a broader set of diseases. The ferret genome will also prove valuable to investigators exploring the conservation genomics of the highly imperiled North American black-footed ferret (*Mustela nigripes*), which is a congener to *M. putorius furo*[22]. The black-footed ferret underwent a population bottleneck in the 1980s leading to greatly diminished genetic diversity, and the resulting congenital defects include reduced immune capacity and anomalies in male fertility[23]. The genomic resources presented here can aid genetic analysis of these defects and the ongoing captive breeding program necessary for the survival of the species[24].

METHODS

Methods and any associated references are available in the online version of the paper.

Accession codes. The *M. putorius furo* genome, GenBank, AEYP00000000; access to assembled contigs and the derived unplaced genomic scaffolds. The assembly can also be found at http://uswest.ensembl.org/Mustela_putorius_furo/Info/Index. Trachea RNA-seq data, SRA, SRP033621; connected to the following BioProjects: genomic reads, PRJNA59869; mRNA-seq reads for tissue survey and lung samples from the influenza model, PRJNA78317. Microarray data sets, GEO: GSE49060 (influenza arrays) and GSE49061 (CF arrays). Agilent array using the developed designs (IDs 048471 and 048472) can be ordered via the Agilent eArray utility (https://earray.chem.agilent.com). Additional community resources related to the paper, including the genomic coordinates for the intergenic and nonpolyadenylated transcripts, and results for the ferret to dog liftOver can be found at http://ucsc.viromics.washington.edu/genomes/ferretGenome.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

This project was funded in whole or in part with federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health (NIH), Department of Health and Human Services, under Contract Nos. HHSN272200800060C and HHSN272201400005C and Public Health Service Grant P51OD010425 (M.G.K.). For the Broad Institute of MIT and Harvard, this project was funded in whole or in part with federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN272200900018C. D.J.G. also received training grants 9T32OD010423-06 and 5T32RR023916-05 from the NIH Office of the Director. J.F.E., S.R.T., X.S. and Y.Y. (University of Iowa) were supported under the NIH National Institute of Diabetes and Digestive and Kidney Diseases grants R37 DK047967, R24 DK096518 and P30 DK054759, and National Heart, Lung, and Blood Institute grant R01 HL108902. C.D. acknowledges support by SNSF advanced researcher fellowship (#136461). A.S. receives support under NIH National Institute of General Medical Sciences R01 GM102192. For S.S., B.A. and T.H., the Wellcome Trust Sanger Institute is operated by Genome Research Limited, a charity registered in England with number 1021457 and a company registered in England with number 2742969. K.L.-T. also receives support under EURYI ERC. We thank Marshall Farms, New York, for providing three adult sable female ferrets (*M. putorius furo*; 421 days old).

AUTHOR CONTRIBUTIONS

X.P., R.E.P., J.F.E., J.T.-G., A.J.E. and S.M.W. wrote the paper with input from other authors. F.D.P., J.A., B.W.B. and K.L.-T. oversaw genome and transcriptome sequencing, and related computational efforts at the Broad Institute. D.B., P.R. and J.T.-M. performed transcriptome analysis, initial genome annotation and initial expression analysis. J.J. assisted in coordinating samples and sequencing data at the Broad Institute. M.L. and R.S. performed library construction for sequencing. A.M.B. and S.Y. generated the genome assembly. T.M.T. and Y.K. provided influenza infected ferret tissues listed in **Supplementary Table 1**. T.H., B.A. and S.S. were responsible for Ensembl annotation pipeline.

Y.K. and A.J.E. oversaw the influenza model at the University of Wisconsin. M.H. implemented the ferret infection protocol and generated primary virological data. D.J.G. and M.S. did immunohistochemical staining for influenza A virus antigen. Y.K., A.J.E. and D.J.G. interpreted overall biological outcomes of the influenza model. J.F.E., X.S. and Y.Y. provided normal ferret tissues for genome and transcriptome sequencing, and RNA samples from the CF ferret model. S.M.K. isolated and characterized RNA from normal ferret tissues. J.C. and M.J.T. isolated RNA from samples from the ferret influenza infection model and generated ribosomally depleted RNA. M.J.T. generated total RNA-seq libraries for RNA from ferret trachea and coordinated the generation of trachea total RNA-seq data. G.L.L. oversaw sample handling and coordinated data generation for the ferret influenza model. K.G. and C.D. performed the phylogenetic and ferret-to-human and mouse-to-human comparisons. N.S. generated the GO term enrichments for the differing angular quadrants of **Figure 1a**. S.R.T. evaluated relative conservation of human genes between ferret or mouse for gene sets associated with biomedical models. A.S. advised on comparative genomics analyses. X.P. did the final analysis of all the RNA-seq data, including generating the expanded annotation, evaluating the tissue specificity, and performing the differential expression analysis for the ferret influenza model. J.T.-G. and R.E.P. provided the associated functional interpretation. X.P. also generated the designs for the ferret microarrays. G.L.L. coordinated sample handling and data generation with the ferret microarrays; J.C. generated the microarray data; X.P. performed the statistical comparisons; R.E.P., X.P., J.F.E. and S.R.T. performed the functional interpretation of transcriptional changes between CF ferret and human samples. R.E.P. and M.G.K. coordinated contributions between the collaborating laboratories.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

1. Tripp, R.A. & Tompkins, S.M. Animal models for evaluation of influenza vaccines. *Curr. Top. Microbiol. Immunol.* **333**, 397–412 (2009).
2. van Riel, D. *et al.* Human and avian influenza viruses target different cells in the lower respiratory tract of humans and other mammals. *Am. J. Pathol.* **171**, 1215–1223 (2007).
3. Belser, J.A., Katz, J.M. & Tumpey, T.M. The ferret as a model organism to study influenza A virus infection. *Dis. Model. Mech.* **4**, 575–579 (2011).
4. Schrauwen, E.J. *et al.* Possible increased pathogenicity of pandemic (H1N1) 2009 influenza virus upon reassortment. *Emerg. Infect. Dis.* **17**, 200–208 (2011).
5. Gnerre, S. *et al.* High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. USA* **108**, 1513–1518 (2011).
6. Curwen, V. *et al.* The Ensembl automatic gene annotation system. *Genome Res.* **14**, 942–950 (2004).
7. UniProt Consortium. Update on activities at the Universal Protein Resource (UniProt) in 2013. *Nucleic Acids Res.* **41**, D43–D47 (2013).
8. Cabili, M.N. *et al.* Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes Dev.* **25**, 1915–1927 (2011).
9. Su, A.I. *et al.* A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc. Natl. Acad. Sci. USA* **101**, 6062–6067 (2004).
10. Ciofani, M. *et al.* A validated regulatory network for Th17 cell specification. *Cell* **151**, 289–303 (2012).
11. van Riel, D. *et al.* H5N1 virus attachment to lower respiratory tract. *Science* **312**, 399 (2006).
12. Walsh, K.B. *et al.* Suppression of cytokine storm with a sphingosine analog provides protection against pathogenic influenza virus. *Proc. Natl. Acad. Sci. USA* **108**, 12018–12023 (2011).
13. Marsolais, D. *et al.* A critical role for the sphingosine analog AAL-R in dampening the cytokine response during influenza virus infection. *Proc. Natl. Acad. Sci. USA* **106**, 1560–1565 (2009).
14. Neill, D.R. *et al.* Nuocytes represent a new innate effector leUK.ocyte that mediates type-2 immunity. *Nature* **464**, 1367–1370 (2010).
15. Wolk, K. *et al.* Maturing dendritic cells are an important source of IL-29 and IL-20 that may cooperatively increase the innate immunity of keratinocytes. *J. Leukoc. Biol.* **83**, 1181–1193 (2008).
16. Sun, X. *et al.* Disease phenotype of a ferret CFTR-knockout model of cystic fibrosis. *J. Clin. Invest.* **120**, 3149–3160 (2010).
17. Sun, X. *et al.* Lung phenotype of juvenile and adult cystic fibrosis transmembrane conductance regulator-knockout ferrets. *Am. J. Respir. Cell Mol. Biol.* **50**, 502–512 (2014).
18. Olivier, A.K. *et al.* Abnormal endocrine pancreas function at birth in cystic fibrosis ferrets. *J. Clin. Invest.* **122**, 3755–3768 (2012).
19. Ogilvie, V. *et al.* Differential global gene expression in cystic fibrosis nasal and bronchial epithelium. *Genomics* **98**, 327–336 (2011).
20. Bonfield, T.L. *et al.* Inflammatory cytokines in cystic fibrosis lungs. *Am. J. Respir. Crit. Care Med.* **152**, 2111–2118 (1995).
21. Ciofu, O., Hansen, C.R. & Hoiby, N. Respiratory bacterial infections in cystic fibrosis. *Curr. Opin. Pulm. Med.* **19**, 251–258 (2013).
22. Ouborg, N.J., Pertoldi, C., Loeschcke, V., Bijlsma, R.K. & Hedrick, P.W. Conservation genetics in transition to conservation genomics. *Trends Genet.* **26**, 177–187 (2010).
23. Wisely, S.M., Buskirk, S.W., Fleming, M.A., McDonald, D.B. & Ostrander, E.A. Genetic diversity and fitness in black-footed ferrets before and during a bottleneck. *J. Hered.* **93**, 231–237 (2002).
24. Wisely, S.M., McDonald, D.B. & Buskirk, S.W. Evaluation of the genetic management of the endangered black-footed ferret (Mustela nigripes). *Zoo Biol.* **22**, 287–298 (2003).

ONLINE METHODS

Animal usage was performed under protocols approved by the Institutional Animal Care and Use Committees (IACUCs) at the University of Wisconsin School of Veterinary Medicine or the University of Iowa. Appropriate biosafety containment was used in the course of infections with the indicated influenza strains.

Genome sequencing and assembly. Three adult sable female ferrets (*M. putorius furo*; 421 days old) obtained from Marshall Farms (via John Engelhardt, Iowa) were sacrificed and specimens sent to the Broad Institute for heterozygosity testing. The individual ID#1420 was selected for sequencing due to its low heterozygosity. The ferret DNA was sequenced to 90× total coverage by Illumina sequencing technology and was composed of 45× coverage using 180 bp fragment libraries, 42× coverage using 3 kb sheared jumping libraries, 2× coverage using 6-14 kb sheared jumping libraries and 1× coverage using ShARC jumping libraries. The reads were assembled into MusPutFur1.0 (accession number AEYP00000000.1) using ALLPATHS-LG[5]. The *M. putorius furo* genome has been reported to have a karyotype of 40 chromosomes[25]. The draft assembly is 2.41 Gb in size and is composed of 2.28 Gb of sequence plus gaps between contigs. The ferret genome assembly has a contig N50 size of 48.8 kb, a scaffold N50 size of 9.3 Mb and quality metrics comparable to other Illumina genomes.

RNA sequencing and assembly. A panel of 24 ferret samples from multiple tissues was RNA sequenced to aid with genome annotation. Developmental (3 staged embryos) and uninfected adult tissues (the individual used for genome sequencing) were obtained from sable ferrets (obtained by the Engelhardt laboratory, Iowa). Two pooled RNA samples were prepared from ferrets that had been infected with strains of 2009 pandemic H1N1; one pool was generated from lung and trachea specimens collected in the laboratory of T. Tumpey (CDC) from two ferrets sacrificed 3 days after infection with A/Mexico/4482/2009 (H1N1). The other RNA pool was generated from materials from laboratory of Y. Kawaoka (Wisconsin), using spleens harvested at days 3 and 6 following infection of two ferrets with A/Wisconsin/WSLH049/2009 (H1N1). All RNAs were extracted at the University of Washington and the RNA-seq libraries were then produced at the Broad Institute by the strand-specific dUTP method from oligo dT polyA-isolated RNA[26]. The libraries were sequenced by Hi-Seq Illumina machines, producing 101 bp reads (3-6 Gb of sequence/tissue). All 24 RNA-seq data sets were assembled via the genome-independent RNA-seq assembler Trinity[27].

Ensembl gene annotation. The genome assembly was annotated by the Ensembl gene annotation system[6] (Ensembl release 70, August 2012). Protein-coding gene models were annotated by combining alignments of Uniprot[7] mammal and other vertebrate protein sequences and RNA-seq data. RNA-seq models were generated from a survey of adult ferret and embryonic ferret tissues, including tissues from an influenza-infected ferret. This pipeline produced 23,963 transcripts arising from 19,910 protein coding genes and 3,614 short noncoding genes. The ferret gene annotation is available on the Ensembl website (http://www.ensembl.org/Mustela_putorius_furo/), including orthologs, gene trees and whole-genome alignments against human, mouse and other mammals. Also included are the tissue-specific mRNA-seq transcript models, indexed BAM files and complete set of splice junctions identified by our pipeline. Further information about the annotation process can be found at http://www.ensembl.org/Mustela_putorius_furo/Info/Annotation, which includes a summary as well as a PDF giving a detailed description of the ferret genebuild.

Comparative genomics. Orthology inference: orthology among the ferret genome and 33 other genomes was inferred using the OMA pipeline[28]. This yielded pairwise orthologs between all species and orthologous groups. The latter were used for 3-way human/mouse/ferret comparisons and species tree inference. The number of genes conserved across mammals and carnivores was computed from hierarchical orthologous groups identified using GETHOGs[29].

Species tree. The 789 orthologous groups covering at least 31 of the 34 species considered were aligned individually and then concatenated. Missing data were represented as "X" characters. Alignment was performed using MAFFT's local-alignment based L-INS-i algorithm. Phylogenetic inference was performed using PhyML, under the JTT, WAG and LG models, and also as a partitioned analysis in RAxML[30,31]. Support values were calculated using bootstrapping (RAxML and PhyML) and approximate Bayesian support (PhyML aBayes).

Scatterplot analysis. To contrast the divergence between human-ferret genes and their human-mouse counterparts, we extracted triplets of orthologs between the three species from all OMA groups computed above. Divergence was computed using two measures: (i) point accepted mutation (PAM) unit estimated by pairwise maximum likelihood distance estimation using Gonnet matrices[32]; and (ii) nucleotide divergence calculated in PhyML from triplet alignments using the general time-reversible model.

Gene ontology (GO) annotations and enrichment analyses. GO terms were assigned to OMA groups by propagating experimental GO annotations (GO evidence codes EXP, IDA, IPI, IMP, IGI, IEP) of any group member to the rest of the group. This procedure assigned 13,509 GO terms, most of them from the Biological Process ontology, to 9,117 OMA groups, resulting in 541,220 GO annotations.

To perform the GSEA analysis[33], we created a list by ordering our data points in **Figure 1b** according to the angle to the *x* axis and determined, for each GO term separately, whether data points are randomly distributed throughout the list or are primarily found at the top or at the bottom. To perform a two-tailed Fisher's exact test, we partitioned the data points in the scatterplot (**Fig. 1b**) into 4 quantiles according its angle to the *x* axis, thereby accounting for the relative evolutionary distance between ferret and mouse. For each GO term, we contrasted each quartile with the other three fourths of the data. We used a two-tailed Fisher's exact test as implemented in R. In both statistical analyses we adjusted the *P* values for multiple testing using the Benjamini & Hochberg method as implemented in R. To organize the enriched GO terms, the terms that passed the enrichment criteria were processed with REVIGO to remove redundant GO terms and cluster semantically similar terms. Very generic GO terms (for example, "macromolecular complex", "organelle part") were excluded, as were singleton terms that were not aggregated into clusters.

Comparison of gene sets specific to models of human health and disease. Increased conservation in gene subsets (**Fig. 1b**) were determined by Wilcoxon signed-rank test in R (version 2.14.1) by the wilcox.test function. Interactions with CFTR (gene subset CFTR interactome in **Fig. 1b**) were obtained from a previously published CFTR IP–mass spec data set[34]. Other five CF-related gene subsets in **Figure 1b** were obtained from http://www.genecards.org[35] using relevancy score cutoffs at the point of greatest Euclidean distance. Two nervous system related GO terms in **Figure 1b** were from the enrichment analysis of proteins with mouse sequences closer to human when tested with Fisher's exact test and Gene Set Enrichment Analysis (false-discovery rate < 0.05) (**Supplementary Tables 5** and **6**).

Expanded custom ferret genome annotation for differential expression analysis. We assembled ferret transcript contigs *de novo* using Trinity[27] from ferret RNA-seq data with default parameters. The RNA-seq data used included mRNA-seq data of the panel of 24 tissue samples for 21 different tissues or tissue mixes (each sample was assembled separately), as well as both mRNA-seq and Total RNA-seq data from the influenza study (all 21 virus infected or control lung samples (each condition and protocol was assembled separately). All assembled transcript contigs were aligned to the ferret reference genome (MusPutFur1.0) using GMAP[36] with default parameters. For those uniquely aligned ferret transcript contigs, their alignments across all mRNA-seq data (transcript contigs from lung Total RNA-seq data were not included here) were merged using cuffmerge (Cufflinks version 2.0.2)[37] to remove redundant alignments and to predict novel genes and transcripts. Predicted transcripts were checked against Ensembl annotation (version 69) to identify: 1) putative intergenic transcripts—those did not overlap with any Ensembl annotated transcripts directly (with class code 'u') and indirectly through any predicted transcripts overlapping Ensembl annotated transcripts, and 2) putative novel isoforms of Ensembl annotated transcripts (with class code 'j', at least one splice junction is shared with a reference transcript). We removed all single exon transcripts or any transcript with the alignment to reference genome shorter than 200 nt to minimize unspliced precursor fragments. For putative

intergenic transcripts, we also removed a small number of predicted transcripts located within introns of other predicted transcripts. For putative novel isoforms, we removed those predicted transcripts that spanned two or more Ensembl annotated genes to minimize mis-assembled transcripts. Similarly, we predicted intergenic transcripts from lung Total RNA-seq data, which were then filtered against both Ensembl annotation and the newly predicted transcripts from mRNA-seq data. We did not require intergenic transcripts from Total RNA-seq to be spliced, but the length of alignment to reference genome had to be at least 120 nt, which was intended to capture ncRNAs that would be longer than small RNAs like miRNAs. For Total RNA-seq data we did not predict novel isoforms to avoid unspliced transcripts. After filtering we combined all predicted transcripts with Ensembl annotation into one annotation, which was used for all downstream gene quantification and differential expression analysis. This expanded annotation was used for Agilent ferret microarray design.

To investigate if Total RNA-seq captured non-polyadenylated transcripts, we performed both Total RNA-seq and mRNA-seq analysis of 21 ferret lung samples. We reasoned that, for the same gene in the same sample, if Total RNA-seq analysis collected many more short reads than mRNA-seq analysis, that gene likely transcribed non-polyadenylated transcripts, since by polyT priming mRNA-seq analysis selected against non-polyadenylated transcripts. To facilitate the comparison, the raw gene read counts were first preprocessed as follows: i) any gene with less than 50 raw read counts in all 42 RNA-seq measurements were removed to ensure genes to be compared were robustly detected at least once in the samples used here, and ii) all gene raw read counts were scaled by the total read counts of remaining genes in each RNA-seq analysis for each sample. Next, for each gene we counted the number of samples (out of 21 samples in total) in which the scaled read count from Total RNA-seq analysis was much larger (1.5-fold or more) than that from the corresponding mRNA-seq analysis.

Comparative analysis of tissue expression patterns. The panel of ferret tissue RNA-seq data generated for genome annotation was used to quantify ferret gene expressions in each tissue and was processed the same way as described in the influenza study section below; depending on the tissue type, the data was from a single individual or from 2–4 individuals. For human data set, alignment files of RNA-Seq read alignment of 24 human tissues and cell types were downloaded from Human lincRNA Catalog website (http://www.broadinstitute.org/genome_bio/human_lincrnas/?q=home).

These data were derived from specimens collected from 9 individuals, with 1–2 contributing individuals per tissue. Similarly we quantified the expression of human genes (Ensembl 69) in each tissue using HT-seq (http://www-huber.embl.de/users/anders/HTSeq/doc/overview.html). We selected the set of 7 tissues (brain, testis, skeletal muscle, heart, lung, liver and kidney) that were common between two data sets for further comparative analysis. We limited the comparison to 15,597 ferret-human gene pairs that had 1:1 ortholog relationships as defined by Ensembl 69. The normalized read count in counts per million (cpm) for each tissue was obtained using edgeR[38]; in instances when there was more than one tissue donor, the resulting data are averages. To focus on genes that tended to be robustly detected in both ferret and human data sets, we applied two *ad hoc* filters on genes based on the observed abundances. First, before read count normalization we instituted a threshold for genes with a raw read count of at least 100 in at least one of the ferret tissues and at least one of the human tissues (not necessarily the same tissue between ferret and human). Second, to account for the differences in sequencing depth, in the normalized read count we further required each gene to have at least 5 cpm in at least one of the ferret tissues and at least one of the human tissues (again, not necessarily the same tissue between ferret and human). The final working set was 12,636 genes. For each gene within each species, we calculated the relative abundance in each tissue as the ratio between its cpm in the tissue and the sum of cpms across all tissues. To evaluate if orthologous genes tend to exhibit concordant tissue-specific expression across ferret and human, we determined the number of genes with highest relative abundance across all tissues of the same species, as well as the intersection of these tissue-specific sets between human and ferret, and assessed the significance of the intersections using a chi-squared test.

To identify groups of genes with similar expression patterns across tissues and species, we clustered genes using k-means partitioning. We iteratively assessed the number of centers (k) to be used for clustering as following: for a given k, we calculated the difference between the converged, total within-cluster sum of squares vs. that from 500 random data sets generated by random permutation of the actual data matrix; for the series of k tested, $k = 7$ had the maximum difference for the final gene clustering. For the results of the $k = 7$ clustering, we used a Mann-Whitney test to evaluate if the overall expression of a cluster of genes was significantly higher in one tissue relative to the rest tissues of the same species, based on the relative abundances in the normalized count matrix.

Cells and influenza viruses. The 2009 pandemic influenza A/California/04/2009 (H1N1) virus, referred to as CA04, and the 1918 pandemic influenza A/Brevig Mission/1/1918 (H1N1) virus, referred to as 1918, were generated by reverse genetics, as described[39–41]. Madin-Darby canine kidney (MDCK) cells for virus titer measurements were from ATCC, and were grown in Eagle's minimum essential medium (MEM) with 5% newborn bovine calf serum (HyClone, Thermo Fisher Scientific) and penicillin/streptomycin. Cell stocks are periodically restarted from early passage aliquots and routinely monitored for mycoplasma contamination.

Ferret infections. Twenty-one 4- to 8-month-old female ferrets were obtained from Triple F Farms Inc. (Sayre, PA, USA), confirmed serologically negative by hemagglutination inhibition assay for currently circulating influenza viruses and randomly assigned to experimental groups. Individual animals were intramuscularly anesthetized with ketamine and xylazine (5 mg and 0.5 mg per kg of body weight, respectively), followed by intranasal inoculation with 500 µl of phosphate-buffered saline (PBS; $n = 3$) alone, PBS containing 1×10^6 plaque forming units (PFU) of the CA04 virus ($n = 9$), or PBS containing 1×10^6 PFU of the 1918 virus ($n = 9$). On day 1 post-infection (p.i.), 3 animals from each infection group were euthanized, and tracheal and lung tissues were harvested for virological, immunohistochemical staining for influenza A virus antigen and RNA sequencing analysis. Tissues were similarly harvested from 3 additional CA04- or 1918-infected ferrets on days 3 and 8 p.i. Tracheal tissues harvested for each individual analysis were derived from the same general region in each ferret, and lung tissues for all analyses were derived from the same lung lobe. We previously examined pathologic lesions in 1918 virus-infected ferret lung, observing macroscopic pathologic changes by day 3 p.i. that included severe lesions and hemorrhage[40]. Since the primary purpose of the present study was to measure gene expression changes in regions of the lung where macroscopic lesions are known to develop, we carefully selected the lung lobe and lung region based on our previous study and then collected samples from the same region for all animals in the study to be consistent. Sample sizes of 3 animals per condition were in keeping with prior reports for exploratory animal models to characterize influenza infection when models require serial sacrifice. While the sample sizes were not the result of a power analysis for a prespecified effect size, the evaluation of the gene expression differences between conditions is performed with statistical stringencies suitable for exploratory assessment, hypothesis generation and reproducibility by alternate techniques such as qPCR. All procedures with ferrets were approved by the University of Wisconsin School of Veterinary Medicine Animal Care and Use Committee, and were performed in an enhanced biosafety level 3 Agriculture (BSL3-Ag) containment suite. All samples derived from influenza virus-infected ferret tissues and containing infectious virus were manipulated in BSL3-Ag containment. Ensuing analysis of samples from the influenza model was performed without blinding, with the exception of histopathology scoring.

Virus quantification and immunohistochemical staining. Ferret tracheal and lung tissues, frozen at −80 °C at the time of excision, were thawed and homogenized in PBS containing penicillin/streptomycin. Cleared supernatants were titrated on MDCK cells using standard methods. For virus antigen immunohistochemical (IHC) analysis, tissues were preserved by immersion in 10% phosphate-buffered formalin (Sigma-Aldrich). Preserved tissues were paraffin embedded and several 5-µm-thick sections were cut for ferret tracheal and lung tissues. Sections were stained with standard hematoxylin and eosin

doi:10.1038/nbt.3079

and then processed for IHC staining with an in-house rabbit anti-influenza virus polyclonal antibody (R309) raised against influenza A/WSN/1933 (H1N1) virus[40].

Quantitative reverse transcription (RT-PCR). Quantitative RT-PCR was performed to assess viral mRNA transcripts from infected ferret tracheal and lung samples used for sequencing. Total RNAs were treated with DNase using DNA-free DNase Treatment and Removal Reagents (Ambion, Inc., Austin, TX). cDNAs from total RNAs were generated using the QuantiTect reverse transcription kit (Qiagen Inc.). A custom-designed TaqMan gene expression assay for influenza Matrix (M) sequence (MPCONS2010), with primers that have complete homology with both 1918 and CA04 M sequences, was ordered from Applied Biosystems, Inc. Taqman experiments were performed on the ABI 7500 Real-Time PCR System platform and each sample was run in quadruplicate. Ribosomal RNA (18S) was used as endogenous control to normalize quantification of each target within tissues using Applied Biosystems Sequence Detections Software version 1.3. The relative amount of viral mRNA ($\log_{10}$) is presented in the final results.

RNA extraction and library preparation. Tissues used for RNA sequencing were excised and immediately immersed in RNALater (Ambion) for 24 h at 4 °C, subsequently frozen at −80 °C, and later thawed and homogenized in TRIzol (Life Technologies); RNA was isolated using QIAGEN miRNeasy protocols.

Total RNA from each sample was divided into two pools for whole transcriptome and mRNA library construction. RNA for whole transcriptome analysis was depleted of rRNA using the Epicentre RiboZero Gold protocol (Epicentre) designed for human, mouse and rat samples, but effective in reducing rRNA amounts for ferret total RNA samples. The presence of 18S and 28S rRNA peaks was checked using the Agilent 2100 Bioanalyzer instrument (Agilent). The rRNA depleted RNA was then used to make strand-specific whole transcriptome libraries[26]. Strand-specific mRNA libraries were constructed using the Illumina TruSeq RNA Preparation Kit (Illumina) according to the manufacturer's guide. Both libraries were quality controlled and quantitated using the Agilent 2100 Bioanalyzer instrument and qPCR (Kapa Biosystems).

Transcriptome sequencing, read mapping and differential expression analysis. Constructed libraries were sequenced using Illumina platform with stranded paired end reads, 2 × 100 nt for all mRNA-seq data, 2 × 100 nt for lung total RNA-seq data and 2 × 50 nt for trachea total RNA-seq data. Lung data sets were assembled via the genome-independent RNA-seq assembler Trinity, with each set of three biological replicates assembled into a single transcriptome assembly. However, the quantitative analysis used the ferret genome as a reference, mapping short reads to the ferret genome using RNA-seq aligner STAR[42] with default parameters. The index used for STAR included splicing junctions from the expanded custom annotation constructed below, genome sequences of influenza viruses used in this study, and human and mouse ribosomal RNA sequences. Gene level quantification was based on the Ensembl gene annotations combined with the expanded list of transcribed genomic regions that were identified using the 63 RNA-seq data sets generated from the influenza model (cf. Supplemental Materials). Quantification was performed using HT-seq. The differential expression analysis was performed using edgeR[38]. Clustering and other statistical analyses were performed using R (http://www.r-project.org/).

Influenza model - transcriptomic analysis details. The expression data from both tissues were combined and processed together, using the generalized linear model approach provided by edgeR. Stages in the analysis are outlined in **Supplementary Figure 18**. Genes were differentially regulated vs. mock in any of the infection conditions were partitioned into disjointed clusters reflecting their tissue or virus specificity. Both Ensembl annotated genes and the expanded list of transcribed genomic regions were quantified using mapped RNA-seq reads. Differential analysis using count data was called significant for adjusted P values ≤ 0.01.

Functional analysis of differential gene expression data. Functional analysis was performed using Ingenuity Pathway Analysis (IPA, Ingenuity Systems, Inc). The software tool analyzes the experimental data set in the context of known biological functions and pathways within the Ingenuity Pathways Knowledge Base, a curated repository of biological interactions and functional annotations. Analysis of the data sets used human annotations, based on the Ensembl listing of human-ferret orthologs. The P values associated with functions or pathways were calculated using the right-tailed Fisher's exact test.

Ferret microarray design and performance assessment. We designed two versions of oligonucleotide microarray using Agilent eArray Web portal (https://earray.chem.agilent.com/earray) to profile both Ensembl annotated transcripts and intergenic transcripts derived from ferret RNA-seq data as described above. In both cases, the longest isoform of each locus was selected for probe design with the 'Design with 3′ Bias' checked, and probe length was set to 60 nt. For first version of microarray (design ID: 048471) 36,950 probes were selected to target Ensembl annotated genes and intergenic transcripts uncovered from mRNA-seq data, and is intended to work with conventional experimental protocols using poly(A) priming for cDNA synthesis. The second version (design ID: 048472) has 64,238 probes selected to target Ensembl annotated genes as well as intergenic transcripts uncovered from both mRNA-seq and Total RNA-seq data. It is intended to work with experimental protocols using random priming for cDNA synthesis to capture both poly(A) and non-poly(A) transcripts. The performance of designed microarray was evaluated by comparing the microarray measurements vs. RNA-seq measurements on the same influenza-infected ferret samples.

Microarray measurements and data analysis. Cy3-labeled cRNA probes were prepared using standard approaches as provided by Agilent Technologies. For the ferret array design ID 048471, Agilent kit 5190-2305 yields probes derived from poly-adenylated RNAs in the starting sample. For design ID 048472, labeled probes were generated with the whole transcriptome labeling kit (part number 5190-2943). Hybridizations were performed as per manufacturer instructions and the slides read on an Agilent Technologies model G2565C high-resolution scanner with extended dynamic range. Image files were processed with Agilent Feature Extraction Software, yielding background-corrected fluorescence intensities with flags for those features deemed not significantly different from background. Statistical analyses to determine differentially regulated genes were performed with the Bioconductor package limma, as described above.

Transcriptional profiling of CF and non-CF ferrets. Homozygous CFTR knockout ferrets were generated and reared as described[16]. Non-CF ferrets were either homozygous or heterozygous for functional CFTR genes. Lung samples were collected after sacrifice at birth or at 15 days post partum and flash frozen. For RNA isolation, lung specimens were ground in liquid nitrogen and then immediately suspended in TRIzol (Life Technologies) and RNA isolated with QIAGEN RNeasy protocols. Animals were assigned to groups solely on the basis of age and genotype (i.e., CF or non-CF). At age 15 days, the comparisons used three CF ferrets (1 F, 2M) and 5 non-CF animals (3 F, 2 M); comparisons of newborn animals used four animals of each phenotype (sexes unknown). Experiments were performed under protocols approved by the Institutional Animal Care and Use Committee at the University of Iowa; statistical considerations for sample sizes were described earlier in the context of the influenza infection model. Array measurements for polyadenylated transcripts (design 048471) were performed as described above and statistical comparisons of CF vs. non-CF animals were done as t-tests as implemented in limma; procedures were performed without blinding. Differences between CF vs. non-CF newborn animal were quite limited and were determined without the use of a multiple test correction. The threshold for differential expression was absolute fold-change ≥ 1.4 and un-adjusted P value ≤ 0.1. Expression differences for CF vs. non-CF 15-day-old animals did use a multiple test correction (Benjamini-Hochberg false discovery rate). The threshold for differential expression was absolute fold-change ≥ 1.5 and false discovery rate ≤ 0.05. These analyses were limited to those array probes that interrogated protein-coding genes within the Ensembl annotation for the ferret genome, and functional interpretation used the corresponding human gene symbols based on the Ensembl mapping of ferret-to-human orthologs. See each

supplemental tables and figures for specifics of filtering criteria for the results of the statistical tests.

Comparative analysis of transcriptional changes in human cystic fibrosis bronchial epithelium. We downloaded the gene expression data on human cystic fibrosis (CF) CF and non-CF bronchial epithelium samples from ArrayExpress (E-MTAB-360)[19]. The Illumina HumanRef-8 Expression BeadChips summary expression data was statistically analyzed using limma with default settings. We filtered out two CF samples ("127 CF BBr", "129 CF BBr") and two non-CF samples ("112 control BBr", "113 control BBr") as potential outliers, due to their relative large deviations from other replicates upon inspecting multidimensional scaling (MDS) plots and the overall expression changes. The final data set included 17 non-CF samples and 10 CF samples. The statistical analysis of differential expression was done at the probe level, and we applied the same multiple test correction (Benjamini-Hochberg false discovery rate) as we did for day 15 ferret CF vs. non-CF comparison. The functional enrichment analysis of differentially expressed genes was performed using Ingenuity Pathway Analysis (IPA), similarly for differentially expressed ferret genes.

To highlight genes concordantly or discordantly differentially expressed in lung samples from day 15 CF ferrets and human CF bronchial epithelium samples, we first identified biological functions enriched in CF/non-CF differentially expressed genes, separately for day 15 ferret CF/non-CF comparison and human CF/non-CF comparison using IPA analysis. From one given function (or a subset of related functions) enriched in both comparisons, we gathered genes differentially expressed in either comparison to identify concordantly or discordantly differentially expressed genes between two comparisons. We applied the following steps to identify genes with concordant expression changes between two comparisons. First, the gene was significantly (fold change ≥ 1.5 and adjusted P value ≤ 0.05) differentially expressed in both comparisons. Second, the gene had the same direction of expression changes in two comparisons. Third, we ranked these genes selected from steps 2 and 3 by their absolute $\log_2$ fold changes within each comparison, i.e., the gene with the largest fold change had a rank of 1 and the gene with the smallest fold change had a rank equal to the total number of these selected genes. This way each gene was assigned with two corresponding ranks, one from the day 15 ferret CF/non-CF comparison and one from the human CF/non-CF comparison. Fourth, we ranked these selected genes by the sum of 1) the difference between their two ranks and 2) the maximum of two ranks. The top genes from this process tended to have both large expression changes in both comparisons and their expression changes tended to be close in magnitude. Similarly we applied the following steps to identify genes with discordant expression changes between two comparisons. First, the gene was significantly (fold change ≥ 1.5 and adjusted P value ≤ 0.05) differentially expressed in both comparisons. Second, the gene had the different direction of expression changes in two comparisons. Third, we ranked these genes selected from steps 2 and 3 by the absolute value of the difference between their two $\log_2$ fold changes, one from the day 15 ferret CF/non-CF comparison and one from the human CF/non-CF comparison. The top gene had the largest difference in fold changes between two comparisons, and with opposite expression changes.

25. Cavagna, P., Menotti, A. & Stanyon, R. Genomic homology of the domestic ferret with cats and humans. *Mamm. Genome* **11**, 866–870 (2000).
26. Levin, J.Z. *et al.* Comprehensive comparative analysis of strand-specific RNA sequencing methods. *Nat. Methods* **7**, 709–715 (2010).
27. Grabherr, M.G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biotechnol.* **29**, 644–652 (2011).
28. Altenhoff, A.M., Schneider, A., Gonnet, G.H. & Dessimoz, C. OMA 2011: orthology inference among 1000 complete genomes. *Nucleic Acids Res.* **39**, D289–D294 (2011).
29. Altenhoff, A.M., Gil, M., Gonnet, G.H. & Dessimoz, C. Inferring hierarchical orthologous groups from orthologous gene pairs. *PLoS ONE* **8**, e53786 (2013).
30. Stamatakis, A. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thoUSA.nds of taxa and mixed models. *Bioinformatics* **22**, 2688–2690 (2006).
31. Guindon, S. *et al.* New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.* **59**, 307–321 (2010).
32. Gonnet, G.H., Cohen, M.A. & Benner, S.A. Exhaustive matching of the entire protein sequence database. *Science* **256**, 1443–1445 (1992).
33. Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* **102**, 15545–15550 (2005).
34. Wang, X. *et al.* Hsp90 cochaperone Aha1 downregulation rescues misfolding of CFTR in cystic fibrosis. *Cell* **127**, 803–815 (2006).
35. Rebhan, M., Chalifa-Caspi, V., Prilusky, J. & Lancet, D. GeneCards: integrating information about genes, proteins and diseases. *Trends Genet.* **13**, 163 (1997).
36. Wu, T.D. & Watanabe, C.K. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* **21**, 1859–1875 (2005).
37. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* **28**, 511–515 (2010).
38. McCarthy, D.J., Chen, Y. & Smyth, G.K. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Res.* **40**, 4288–4297 (2012).
39. Ozawa, M. *et al.* Impact of amino acid mutations in PB2, PB1–F2, and NS1 on the replication and pathogenicity of pandemic (H1N1) 2009 influenza viruses. *J. Virol.* **85**, 4596–4601 (2011).
40. Watanabe, T. *et al.* Viral RNA polymerase complex promotes optimal growth of 1918 virus in the lower respiratory tract of ferrets. *Proc. Natl. Acad. Sci. USA* **106**, 588–592 (2009).
41. Neumann, G. *et al.* Generation of influenza A viruses entirely from cloned cDNAs. *Proc. Natl. Acad. Sci. USA* **96**, 9345–9350 (1999).
42. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).

doi:10.1038/nbt.3079

C

THE OMA ORTHOLOGY DATABASE IN 2015: FUNCTION PREDICTIONS, BETTER PLANT SUPPORT, SYNTENY VIEW AND OTHER IMPROVEMENTS

I worked on the FamilyAnalyzer tool, which was used to infer the numbers of gene duplication, gain and loss events that are displayed in Figure 6 of the paper reproduced in this appendix. Further details are given in chapter 5.

Nucleic Acids Research Advance Access published November 15, 2014

doi: 10.1093/nar/gku1158

The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements

Adrian M. Altenhoff[1,2,3], Nives Škunca[1,2,3], Natasha Glover[1,4,5], Clément-Marie Train[3], Anna Sueki[1], Ivana Piližota[1], Kevin Gori[6], Bartlomiej Tomiczek[1], Steven Müller[1], Henning Redestig[5], Gaston H. Gonnet[2,3] and Christophe Dessimoz[1,6,*]

[1]University College London, Gower Street, London WC1E 6BT, UK, [2]Swiss Institute of Bioinformatics, Universitätstr. 6, 8092 Zurich, Switzerland, [3]ETH Zurich, Computer Science, Universitätstr. 6, 8092 Zurich, Switzerland, [4]Institut National de la Recherche Agronomique (INRA) UMR1095, Genetics, Diversity and Ecophysiology of Cereals, 5 Chemin de Beaulieu, 63039 Clermont-Ferrand, France, [5]Bayer CropScience NV, Technologiepark 38, 9052 Gent, Belgium and [6]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

Received September 22, 2014; Revised October 24, 2014; Accepted October 29, 2014

ABSTRACT

The Orthologous Matrix (OMA) project is a method and associated database inferring evolutionary relationships amongst currently 1706 complete proteomes (i.e. the protein sequence associated for every protein-coding gene in all genomes). In this update article, we present six major new developments in OMA: (i) a new web interface; (ii) Gene Ontology function predictions as part of the OMA pipeline; (iii) better support for plant genomes and in particular homeologs in the wheat genome; (iv) a new synteny viewer providing the genomic context of orthologs; (v) statically computed hierarchical orthologous groups subsets downloadable in OrthoXML format; and (vi) possibility to export parts of the all-against-all computations and to combine them with custom data for 'client-side' orthology prediction. OMA can be accessed through the OMA Browser and various programmatic interfaces at http://omabrowser.org.

INTRODUCTION

The flood of newly sequenced genomes presents a daunting interpretation challenge. Fortunately, the common origin of all living beings implies that many genes are conserved across species—in some cases despite billions of years of intervening evolution. Elucidating evolutionary relationships amongst genes and genomes is thus a key step in the analysis of new data. Sequences that have a common ancestry—homologs—are typically refined into orthologs, which are pairs of genes that started diverging via speciation, and paralogues, which are pairs of genes that started diverging via gene duplication (1,2). This distinction is useful in a broad range of contexts, including multigene phylogenetic inference, propagation of experimental knowledge from model organisms to non-model organisms and the study of gene evolution and adaptation (reviewed in 3,4). The need for orthology inference has led to the development of numerous methods (reviewed in 5) and databases, notably including EggNOG (6), Ensembl Compara (7), Inparanoid (8), MBGD (9), OrthoDB (10), OrthoMCL (11), Panther (12), PhylomeDB (13), Plaza (14) and OMA (15).

The OMA (Orthologous MAtrix) project is a method and database for the inference of orthologs amongst complete proteomes (i.e. the protein sequences associated for every protein-coding gene in all genomes). Initiated in 2004, OMA has undergone 17 major releases, steadily increasing the number of proteomes under consideration from 150 to 1706 across all domains of life. Besides its large scope, the distinctive features of OMA are the high specificity of the inferred orthologs (e.g. 16–19), feature-rich web interface, availability of data in a wide range of formats and interfaces and frequent update schedule of two releases per year.

In this update paper, after providing a brief review of the OMA pipeline, we present major new features recently added to OMA: a new web interface and reorganization, integrated gene ontology function prediction, better support of plant genomes, a synteny viewer depicting orthology relationships in their genomic context, statically computed hierarchical orthologous groups (HOGs) and the possibility to export genomes including all-against-all computations and to combine them with custom genome/transcriptome data.

*To whom correspondence should be addressed. Tel: +44 20 7679 0079; Fax: +44 20 7679 7193; Email: c.dessimoz@ucl.ac.uk

OVERVIEW OF THE OMA INFERENCE PIPELINE

OMA's inference algorithm consists of three main phases:

(i) First, to infer homeologous sequences (sequences of common ancestry), we compute all-against-all Smith–Waterman alignments between every sequence and retain significant matches.
(ii) Second, to infer orthologous pairs (the subset of homologs related by speciation events), mutually closest homologs are identified based on evolutionary distances, taking into account distance inference uncertainty and the possibility of hidden paralogy due to differential gene losses (20,21).
(iii) Third, these orthologs are clustered in two different ways, which are useful for different purposes: (a) we identify cliques of orthologous pairs (OMA groups). Because all relations in one OMA group are orthologous, these are useful as marker genes for phylogenetic reconstruction and tend to be highly specific (18); (b) we identify HOGs, groups of genes defined for particular taxonomic ranges and identify all genes that have descended from a common ancestral gene in that taxonomic range (22).

OMA infers evolutionary relationships between genes from protein sequences, using one protein sequence per gene. If multiple splicing variants are possible, the best one in terms of matches with other genomes is selected, which is not necessarily the longest one (15).

NEW WEB INTERFACE WITH BETTER ORGANIZATION

The OMA browser has been reorganised and redesigned to make it user-friendlier. The menu bar provides a consistent and persistent overview of all main functionalities. The documentation and help pages have been restructured and extended. The new 'responsive' layout takes advantage of large contemporary screens whilst also accommodating small screens such as smartphones and tablets. The landing page now provides pointers to introductory explanations for new users and recent announcements for returning users (Figure 1).

GENE ONTOLOGY FUNCTION INFERENCE AS PART OF THE OMA PIPELINE

One key motivation for orthology inference is to computationally predict the roles that genes play in living organisms—e.g. Cellular Component, Molecular Function and Biological Process of the Gene Ontology (23). For many years, Gene Ontology (GO) annotations from the UniProt-GOA database (24) have been linked to all sequences in OMA. Additionally, we now provide inferred annotations based on orthology relationships: within the orthologous groups, we propagate GO annotations across different species.

To infer GO annotations, we start with curated annotations that are based on direct evidence from the literature: GO evidence codes EXP, IDA, IPI, IMP, IGI and IEP (http://geneontology.org/page/guide-go-evidence-codes). We then propagate them across OMA groups—sets of genes for which all members are inferred to be mutually orthologous—as these have been previously shown to be highly coherent in terms of functional annotations (25). Additionally, to avoid over-propagating clade-specific terms (e.g. 'nematode larval development' outside the nematodes), we require that propagated terms be used in at least one literature-based annotation in the clade in question. For example, the OMA group with fingerprint 'VWQCDTP' contains a *Caenorhabditis elegans* gene annotated with the GO term 'nematode larval development' (Figure 2); this term is not appropriate for genes outside of the Nematoda phylum. Therefore, when propagating this GO term to, for example, the poorly annotated *Arabidopsis thaliana* protein within the same OMA group, we only propagate those parent terms of 'nematode larval development' that are known to be associated with plant proteins; in this case, the most specific amongst those is 'post-embryonic development' (Figure 2). Indeed, the propagated annotation complements one of the known annotations for the *A. thaliana* protein, 'embryo sac development'.

Overall, the OMA database now provides 442 376 477 function annotations for 7 947 728 proteins (Figure 3). Amongst the available annotations, most are computationally inferred; our own predictions constitute about 20% of the available annotations.

Function annotations based on OMA orthologs are particularly valuable for proteins for which other computational annotation methods provide no annotations and the available annotations assigned by curators are relatively general and/or sparse. In the most recent OMA release, we provide annotations for 423 983 proteins for which there are no other electronic annotations. For example, at the time of writing the *A. thaliana* protein with UniProt identifier Q8VYZ5 had no electronically inferred GO annotations (evidence code IEA); it had five annotations based on evidence codes ISS or RCA, which are not used in our propagation pipeline; and the annotations from literature-based evidence were 'nucleolus' (IDA), 'rRNA processing' (IMP) and 'embryo sac development' (IMP). Using our OMA annotation pipeline, we assigned new annotations that complement these: for example, we inferred GO terms 'RNA 5'-end processing' and 'endonucleolytic cleavage involved in rRNA processing' that complement the known experimental annotation 'rRNA processing'; we inferred the GO term 'post-embryonic development' that complements the known experimental annotation 'embryo sac development' (Figure 3).

BETTER SUPPORT FOR PLANT GENOMES, INCLUDING HOMEOLOGY IN WHEAT

One research area where comparative genomics can make an important difference is modern crop science. Indeed, plant genomes tend to have highly redundant genomes as a result of their complex history of duplication and hybridisation events. With almost all genes being available in several copies on multiple sub-genomes, the use of comparative genomics is essential in order to map knowledge across different species. Several specialised plant resources already exist—such as Ensembl Plants (26), Gramene (27),

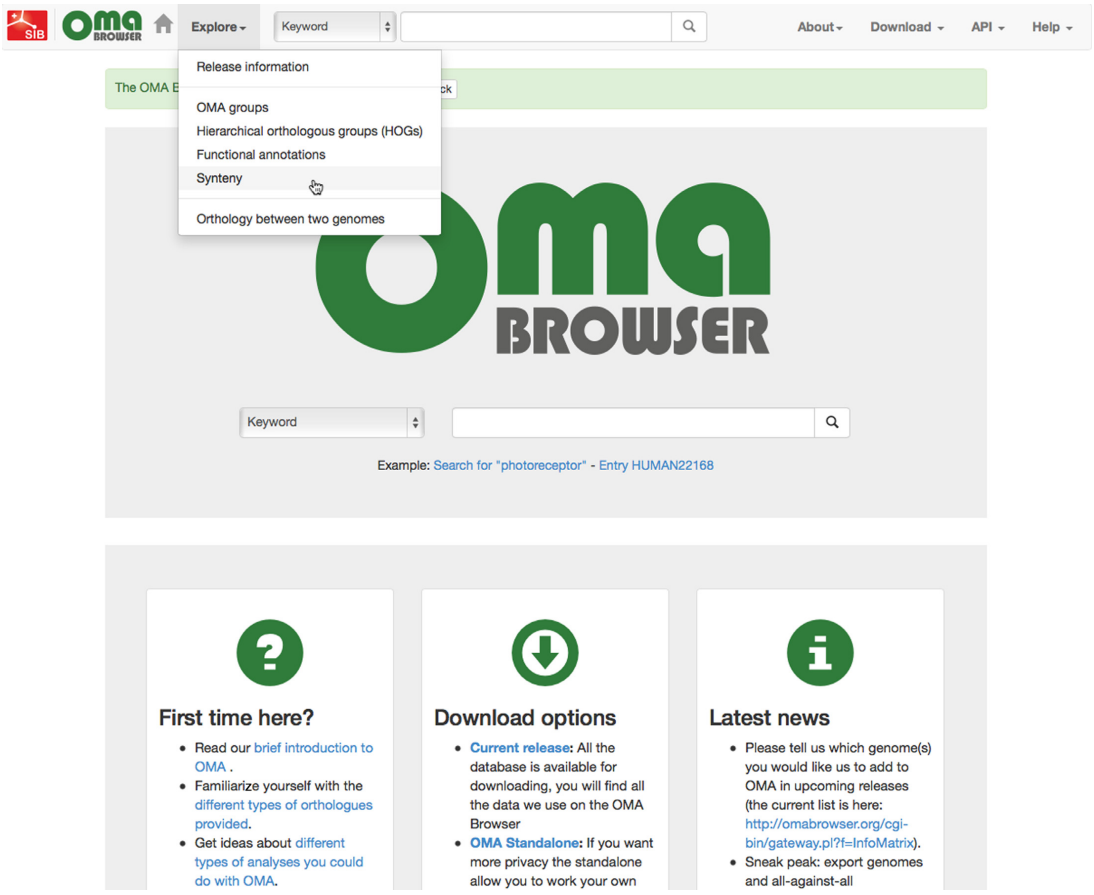


Figure 1. User-centric new design. The website has been redesigned with an emphasis on usability.

Greenphyl (28) and Plaza (29)—but there is value in providing plant support in resources inferring orthology across all domains of life. Also, plant-based analyses can benefit from the other distinctive features of OMA, such as its highly specific predictions and ability to infer HOGs. We have improved plant genome support in OMA by adding and updating more plant genomes and by inferring and annotating homeology—genes related through polyploidization—in the wheat genome.

The number of plant species in the OMA database has increased from 8 to 28 plants in recent years. In the latest release, we have added *Selaginella moellendorffii* (a lycophyte) as the deepest branching vascular plant and *Physcomitrella patens* (a bryophyte) as a representative of the non-vascular plants, thus widening the taxon set to cover ~450 million years of plant evolution (30). We have also added the important model grass species *Brachypodium distachyon* and *Aegilops tauschii*. Additionally, we have added a variety of crop species of practical and economic importance, which are especially useful to plant geneticists and breeders. These species include: banana (*Musa acuminata subsp. malaccensis*), potato (*Solanum tuberosum*), several rice species (*Oryza brachyantha*, *Oryza glaberrima*, *Oryza sativa subsp. indica*), foxtail millet (*Setaria italica*) and bread wheat (*Triticum aestivum*).

In particular, bread wheat is the staple food source for 30% of the human population, making it one of the world's most important cereal crops. However, its very large (17 Gb), highly repetitive, hexaploid (2n = 6x = 42) genome, has made studying its organization and evolution notoriously challenging due to the lack of a high-quality reference sequence. Wheat is a recent allopolyploid resulting from two recent (<0.8 MYA ago) hybridization events between three diploid progenitors, of which the most distant pair diverged an estimated 6.5 MYA ago (31). Following that hybridization event, there has seemingly been little or no recombination across the chromosomes derived from the three progenitor genomes (32). It is therefore helpful to think of these three sets of chromosomes as 'subgenomes'. This gives rise to the notion of homeologous (also spelled 'homoeologous') chromosomes—closely related pairs of chromosomes between two subgenomes. These homeologous chromosomes have maintained a high degree of conservation amongst them, with highly similar genes located on the same chromosomal group (1 to 7) of each subgenome. However, because there have been extensive gene duplications,

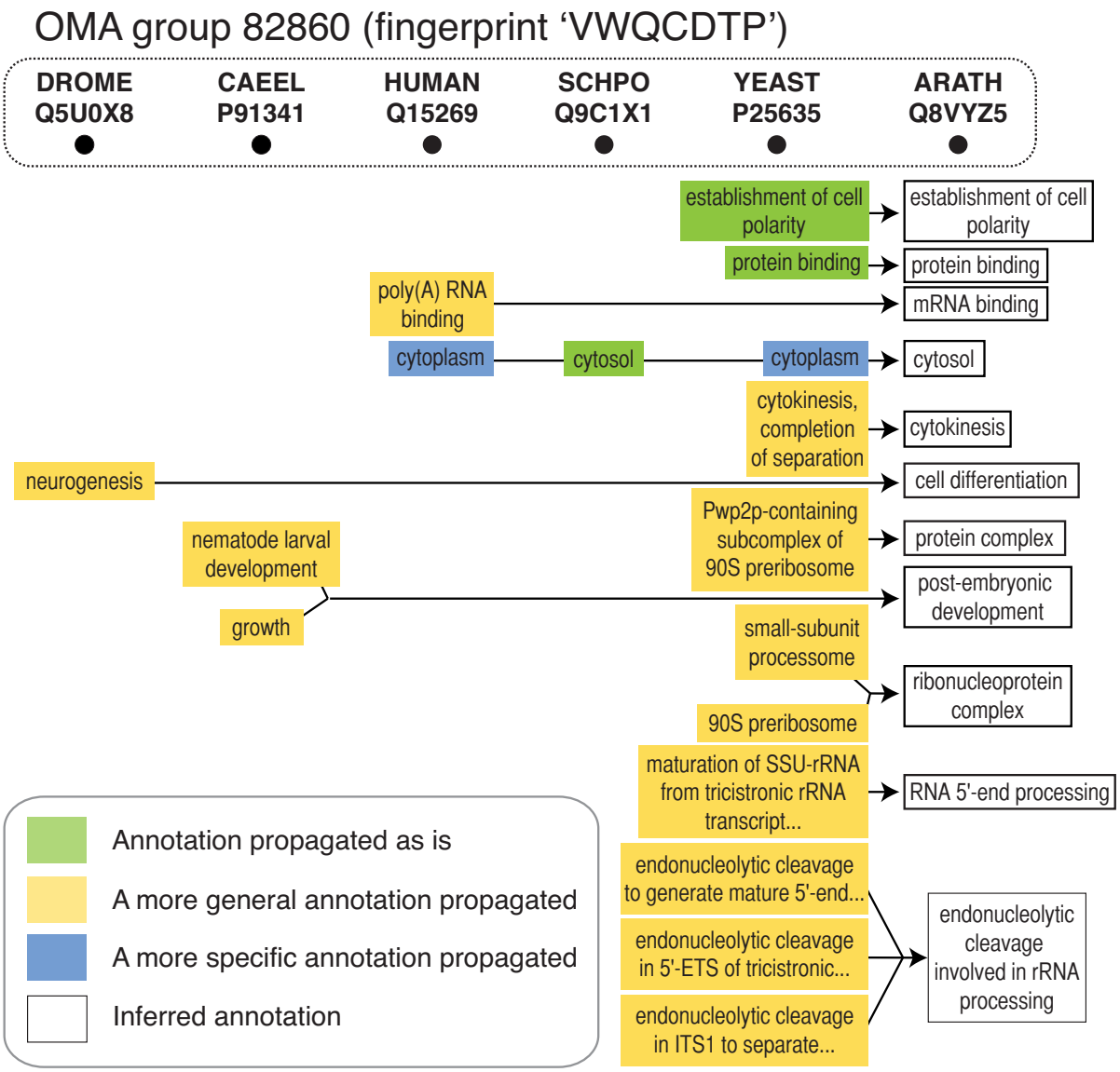


Figure 2. Gene Ontology propagation in the OMA pipeline. New Gene Ontology (GO) annotations for the sparsely annotated *Arabidopsis thaliana* protein Q8VYZ5 are inferred by propagating annotations from other members of the OMA group, taking into account implied parental terms and lineage-specific terms (see main text). For example, the inferred biological process Gene Ontology (GO) term 'post-embryonic development' is based on the more specific GO term 'nematode larval development'; the latter is in itself inappropriate to assign to a protein in the plant clade. Proteins are labelled with their SwissProt/UniProt identifiers. The abbreviations ARATH, CAEEL, SCHIPO, DROME, HUMAN and YEAST refer to species *Arabidopsis thaliana, Caenorhabditis elegans, Schizosaccharomyces pombe, Drosophila melanogaster, Homo sapiens* and *Saccharomyces cerevisiae*, respectively.

losses and rearrangements in the *Triticeae* lineage (32–35), the relationship across homeologs is not necessarily 1:1:1.

In OMA, we define homeologous genes as pairs of homologous genes that have started diverging through speciation between the progenitor genomes and then merged back into the same genome by hybridization. Thus, homeologs can be thought of as 'orthologs between subgenomes'. This suggests a simple way of adapting the OMA pipeline to infer homeologs: we first partitioned the predicted wheat proteins into the three subgenomes based on the annotation of the IWGSC (32), then inferred 'orthologs' between these subgenomes using our standard pipeline. Although conceptually straightforward, this procedure is complicated by the fragmentary nature of the current wheat survey genome, consisting of many contigs and resulting in numerous genes which are split, misannotated, or simply missing.

Dubious homeolog inferences are discarded in two steps. The first filter, part of the standard OMA algorithm, identifies instances of differential gene losses through witnesses of non-orthology in a third genome (21). This filter dis-

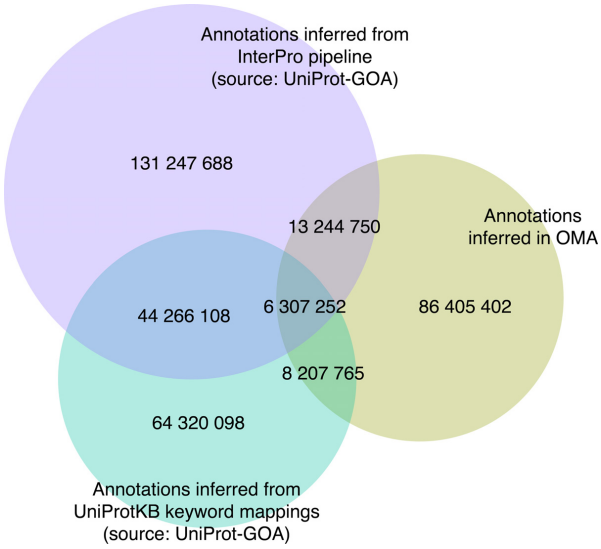


Figure 3. Numbers of electronic Gene Ontology annotations in the OMA database. Three major sources of electronic annotations are shown: annotations through the association of InterPro records with GO terms, annotations based on UniProtKB keyword mappings and annotations inferred in the OMA pipeline. The intersections show the numbers of annotations in common amongst the resources.

carded 4166 pairs. The second filter, developed specifically for homeology detection, considers the distribution of the evolutionary distances and removes outliers (defined as gene pairs with a distance higher than 2.5 standard deviations above the mean distance) from the set of reported homeologs. This discarded an additional 2212 pairs.

Two indicators suggest that the bulk of these discarded pairs are indeed unlikely to be homeologous. First, assuming that the majority of genes have remained in their ancestral position in the *Triticeae* lineage, most homeologous relationships should be between genes on corresponding chromosome groups. Yet only 14.7% of all the pairs discarded by witnesses of non-orthology and 34.7% of outliers are inferred to be between the same chromosome group (compared to 14.5% for random pairs). Second, because the three progenitor genomes diverged relatively recently (~6.5 MYA), most homeologs can be expected to be highly similar. Yet the evolutionary distance between discarded homolog pairs is on average much higher than for the retained pairs, even if we only consider pairs filtered in the first step (Figure 4A).

We applied the same indicators to the 62 910 retained homeolog inferences. The proportion of retained homeologs involving pairs of genes on corresponding chromosome groups was considerably higher (62.8% versus 14.7–34.7% for discarded pairs). Furthermore, as expected, the distribution of evolutionary distance between predicted homeologs was skewed towards low distances, with a mean of 12.6 PAM (0.126 substitutions per site) and a standard deviation of 20.6 PAM (Figure 4B). As an additional assessment, we selected a random subset of 20 homeologous gene pairs and performed a manual validation taking into account sequence quality, gene annotation, shared chromosome group, percentage identity and evolutionary distance between pairs. Fifty-five percent of the predictions could be confirmed, with the rest being either inconclusive or likely mistakes due to misannotations (transposons, chloroplast genes), missing true homeologous counterparts, etc. (Supplementary Table S1). Given that the process of flow sorting of the wheat chromosomes and arms resulted in on average 10% contamination with other chromosomes (32), a small proportion of bona fide homeolog pairs can be expected to be erroneously annotated as belonging to different chromosome group.

In the OMA browser, retained homeolog inferences are labelled as 'high confidence' if they involve genes belonging to consistent chromosome groups, and 'low confidence' if they do not. In the latest release, this resulted in 39 442 pairs (63.2%) of high-confidence homeology predictions and 23 468 (36.8%) low-confidence ones. The average percent identity for the 12 high confidence pairs is 95.4% compared to 90.5% for low confidence pairs. We chose not to be too stringent in the cut-off for evolutionary distance and/or percent identity because although most homeolog pairs have a high degree of conservation, this might not necessarily be true for certain genes that evolve quickly such as disease resistance genes (36), transcription factors (37) or pentatricopeptide repeat proteins (38).

NEW SYNTENY VIEWER PROVIDING THE GENOMIC CONTEXT OF ORTHOLOGS

In the absence of genome rearrangement, orthology relationships can be expected to be consistent across neighbouring genes—a concept commonly referred to as 'shared synteny'. Patterns of syntenic conservation or divergence can shed light on the evolutionary history of genomic loci of interest; they can also reveal sequencing artefacts, misannotations or orthology inference errors. Synteny visualization tools have been successfully developed in several comparative genomics databases such as Yeast Gene Order Browser (39), Genomicus (40) or GnpIS (41). The OMA Browser now features a synteny viewer as well.

The OMA synteny viewer uses a typical layout: genes are represented by boxes, with neighbouring genes displayed in adjacent columns and orthologous regions displayed in different rows. The reference syntenic block, centred on a query gene, is displayed in the first row. The other rows are centred on genes that are orthologous to the query gene, ordered by increasing taxonomic distance to the query gene species. Orthology relationships to each gene contained in the reference syntenic block are coded using different colours. To convey many-to-one and many-to-many relationships, we use stripes of the relevant colours. To aid clarity, hovering over a gene highlights all orthologs of the same colour including those with stripes. The data can be conveniently explored by clicking on any gene, which recentres the display on that gene as a new query.

To illustrate the usefulness of the new synteny viewer, consider the arrangement of alcohol dehydrogenase (ADH) genes around human *ADH1A* (Figure 5). The human ADH gene cluster ADH7 (class IV)-ADH1C (class I)-ADH1B (class I)-ADH1A (class I)-ADH6 (class V)-ADH4 (class

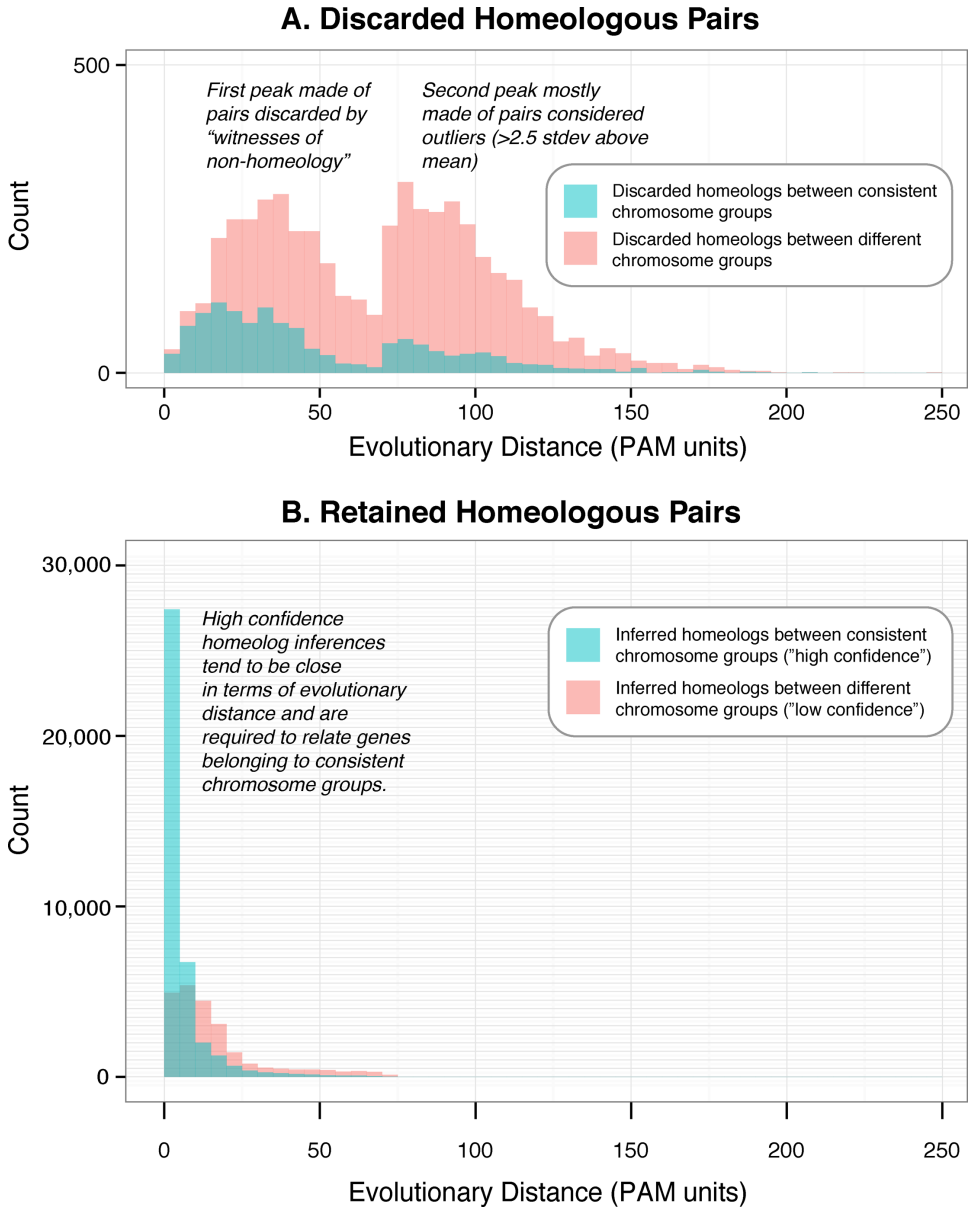


Figure 4. Distribution of evolutionary distances for homeologous pairs that were (**A**) discarded via witness of non-homeology or because they were outliers, or (**B**) retained as inferred homeologs. In both plots, the blue colour represents pairs where both homeologs are located on the same chromosome group and the red colour indicates pairs where homeologs are located on different chromosome groups. The y-axes are drawn at different scales but the grid is consistent across the two plots.

II)-ADH5 (class III) is displayed in the first row. Because the cluster sits on the complementary strand, it appears in reverse order—starting in column 3 (Gene ID 22172) and ending in column −3 (22163). The synteny viewer suggests that the neighbourhood of orthologous genes is well conserved amongst simians, but the conservation diminishes as we move to more distant lineages. Genes with stripes are in one-to-many or many-to-many orthologous relationships with human ADH1A (22168), human ADH1B (22169) and human ADH1C (22171). In particular, the presence of two orthologs in the bushbaby (OTOGA) suggests a separate duplication within the lemur lineage, yielding many-to-many orthology. These observations are all consistent with detailed analyses in the literature (42). Although positioned within well-conserved syntenic regions, genes 13367 in the chimp (PANTR) and 15069 in the gorilla (GORGO) have no human orthologous counterpart in this region. On account of their very short lengths—13 AA and 14 AA,

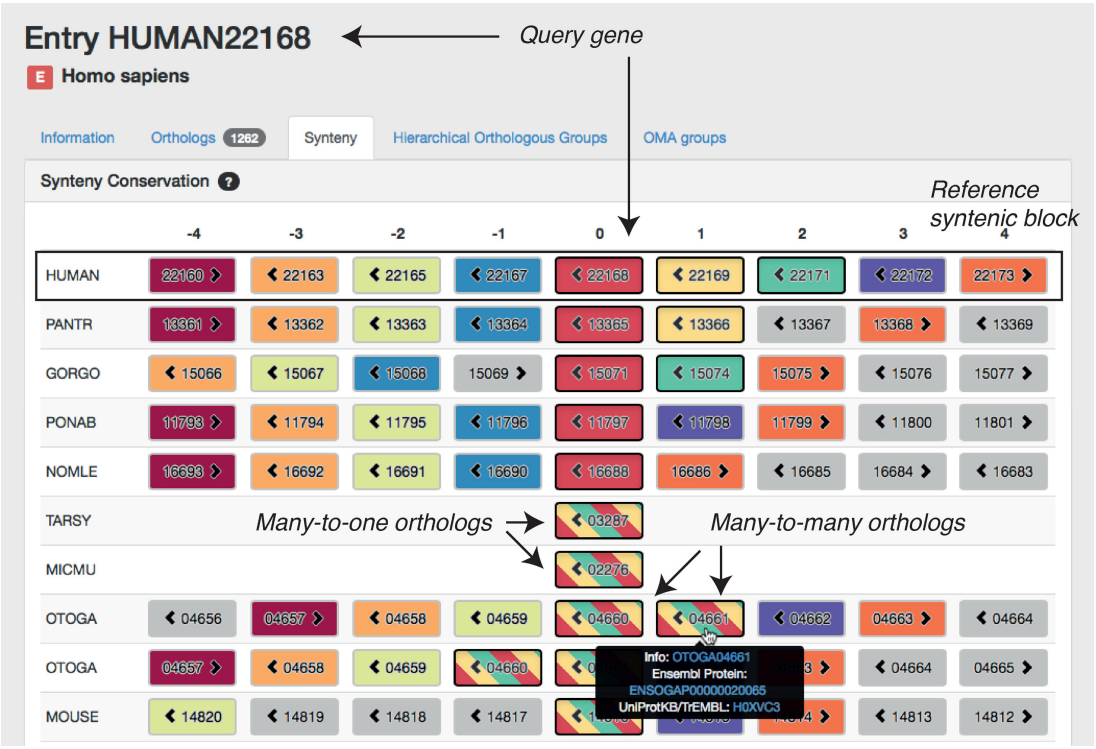


Figure 5. Screenshot of the new OMA synteny viewer with the *ADH1A* gene in human (Gene ID 22168) as query. Each gene is illustrated as a box containing a numerical OMA Gene ID and an arrow to indicate the gene's orientation. The colour of genes outside the query species indicates orthologous relationship with human genes, with bands of colour capturing many-to-one and many-to-many relationships. Genes that are non-orthologous to all nine human genes contained in this window are displayed in grey. The fragmented assemblies of tarsier (TARSY) and mouse lemur (MICMU) contain no genes next to 03287 and 02276, respectively.

respectively—they are likely to be fragments. Furthermore, the absence of flanking genes in the tarsier (TARSY) and mouse lemur (MICMU) is due to the low quality of the genome assembly in these regions.

BETTER SUPPORT FOR HOGS

As discussed above in the overview of the OMA pipeline, HOGs are a key output of the OMA algorithm; they group all the sequences that have descended from a single common ancestral gene within clades of interest. This provides an intuitive framework to generalise the concept of orthology to more than two species. For instance, if we consider the human ADH1A gene discussed in the previous section, it belongs to an HOG containing ADH1B and ADH1C as well, whilst at the more specific level of simians, the three genes belong to three distinct HOGs. This difference in resolution makes intuitive sense because as we consider a broader or narrower range of species, the shared attributes amongst them can be expected to be coarser or finer.

OMA HOGs are inferred from orthologous pairs using a fast and effective algorithm described previously (22). However, until recently, the OMA Browser had been dynamically inferring these HOGs on user demand. Large families could take a few minutes to process. Furthermore, because of the non-deterministic nature of the inference algorithm, there could be small inconsistencies for requests at different taxonomic levels (e.g. one sequence included in an HOG defined at the level of vertebrates but not included at the level of all bilateria). Starting with the latest release, HOGs are precomputed thereby providing rapid user access and consistent inferences. HOGs can now be downloaded in OrthoXML format (43) for further analyses.

One potential use of the HOGs data is to map gene losses, duplications and gains onto species trees. Indeed, since HOGs are defined in terms of ancestral genomes at all internal nodes in the species tree, keeping track of the number of HOGs and their content whilst traversing the tree can yield these quantities. Contrary to approaches solely based on gene counts in extant genomes (e.g. 44), HOGs take into account relationships between the actual sequences and thus can be expected to yield more precise estimates. Furthermore, this approach allows the user to identify the specific genes that underwent duplication or losses on particular branches of the phylogeny.

To illustrate this application, we provide an estimate of gains and losses in the primate tree obtained by parsing OMA HOGs (Figure 6). Large numbers of losses on terminal branches can be indicative of fragmentary genomes (45), such as the tarsier with its low 1.82x coverage. Even so, previous studies have reported elevated duplication and loss rates in the primate lineage (46).

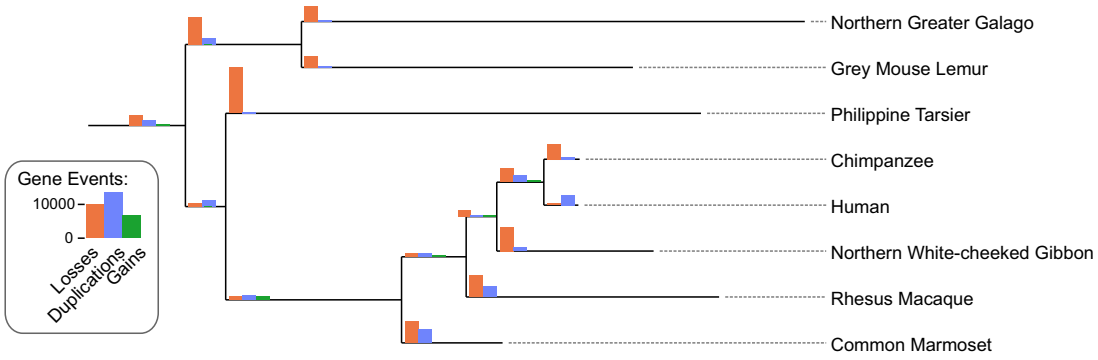


Figure 6. Gene losses, duplications and gains from hierarchical orthologous groups. Gene duplications, losses and gains on the primate lineage inferred from OMA hierarchical orthologous groups.

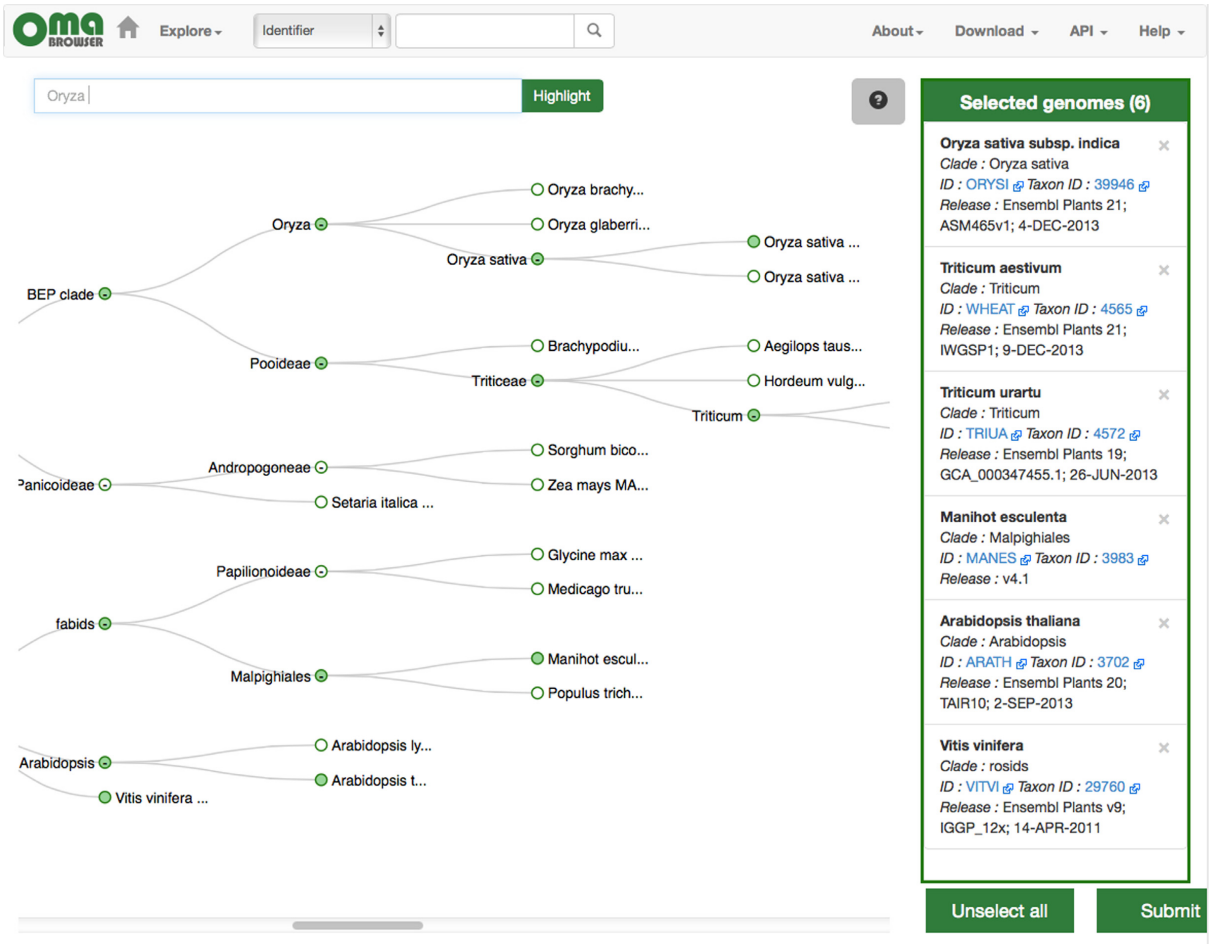


Figure 7. Selection tool for pre-computed genome export. This new function enables users to export genomes of interest and their associated all-against-all comparisons for analysis in the OMA standalone software.

EXPORT OF PROTEIN SETS AND THEIR ASSOCIATED ALL-AGAINST-ALL COMPUTATIONS

As genome and transcriptome sequencing are becoming affordable and ubiquitous, there is an increasing need for orthology prediction on custom data. As a solution to this, we have developed OMA standalone, a downloadable open source implementation of the OMA pipeline for Linux and Mac (the details of the software are the focus of a forthcoming publication). To enable users to efficiently combine custom and public genomes, we have added the possibility of exporting OMA genomes, including all-against-all computations amongst them, as input files for OMA standalone. The function is accessible via the 'Download' menu in the navigation bar of the new OMA Browser interface. Users can select up to 50 genomes for export (Figure 7), which together with OMA standalone are packaged for download as a single compressed *tar* file.

OUTLOOK

For just over a decade, the OMA database has provided orthology inference amongst complete genomes. It has remained true to its mission of providing reliable, high-quality orthology inferences across a broad taxonomic range. With 17 major releases, each including ∼100 additional and updated genomes, the project has been maintained with sustained endurance. At the same time it has also gained numerous functionalities, of which the most recent are highlighted in this update.

So what awaits OMA in the coming decade? One major challenge facing many phylogenomic resources is to keep abreast of the rapid increase in sequencing data (4). In OMA, the all-against-all protein comparison phase—the most time-consuming phase with >7 million CPU hours logged to date—grows quadratically with the number of sequences under consideration. But computational bottlenecks are nothing new in OMA; they have been a *leitmotif* all along and our experience has been that they can generally be overcome through software optimization (e.g. 47) or new heuristics (e.g. 48). We also see potential in sharing computations across different resources and have initiated a joint effort with OrthoDB (10) in that direction.

Another challenge lies with fragmentary, poorly annotated genomes and their potentially disruptive effect on orthology inference and interpretation. Yet at the same time, orthology can also help identify split genes (49). Furthermore, as discussed above, orthology combined with synteny information or integrated across multiple species in hierarchical groups can also uncover quality problems with the data.

One thing however seems certain: as the pace of genome sequencing continues to accelerate, elucidating evolutionary relationships across different genes will remain the key to exploiting the richness of this data. OMA is thus likely to stay relevant.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENT

We thank three anonymous referees for their comments on the manuscript.

FUNDING

Service and Infrastructure, Swiss Institute of Bioinformatics [to G.H.G., C.D.]; UK Biotechnology and Biological Sciences Research Council [BB/L018241/1 to C.D.]; UCL Impact Award, University College London [to C.D., I.P.]; Bayer CropScience NV [to N.G., I.P.]; Biomedical Vacation Studentship, Wellcome Trust Foundation [to A.S.]; EMBL [to K.G.]. Funding for open access charge: BBSRC via the University College London Library.
Conflict of interest statement. None declared.

REFERENCES

1. Fitch,W.M. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool.*, **19**, 99–113.
2. Sonnhammer,E.L.L. and Koonin,E.V. (2002) Orthology, paralogy and proposed classification for paralog subtypes. *Trends Genet.*, **18**, 619–620.
3. Gabaldón,T. and Koonin,E.V. (2013) Functional and evolutionary implications of gene orthology. *Nat. Rev. Genet.*, **14**, 360–366.
4. Sonnhammer,E.L.L., Gabaldón,T., Sousa da Silva,A.W., Martin,M., Robinson-Rechavi,M., Boeckmann,B., Thomas,P.D., Dessimoz,C. and the Quest for Orthologs consortium. (2014) Big data and other challenges in the quest for orthologs. *Bioinformatics*, **30**, 2993–2998.
5. Altenhoff,A.M. and Dessimoz,C. (2012) Inferring orthology and paralogy. In: Anisimova,M (ed). *Evolutionary Genomics. Methods in Molecular Biology*. Humana Press, Clifton, NJ, **855**, pp. 259–279.
6. Powell,S., Forslund,K., Szklarczyk,D., Trachana,K., Roth,A., Huerta-Cepas,J., Gabaldón,T., Rattei,T., Creevey,C., Kuhn,M. *et al.* (2014) eggNOG v4.0: nested orthology inference across 3686 organisms. *Nucleic Acids Res.*, **42**, D231–D239.
7. Flicek,P., Ahmed,I., Amode,M.R., Barrell,D., Beal,K., Brent,S., Carvalho-Silva,D., Clapham,P., Coates,G., Fairley,S. *et al.* (2013) Ensembl 2013. *Nucleic Acids Res.*, **41**, D48–D55.
8. Östlund,G., Schmitt,T., Forslund,K., Köstler,T., Messina,D.N., Roopra,S., Frings,O. and Sonnhammer,E.L.L. (2010) InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. *Nucleic Acids Res.*, **38**, D196–D203.
9. Uchiyama,I., Mihara,M., Nishide,H. and Chiba,H. (2012) MBGD update 2013: the microbial genome database for exploring the diversity of microbial world. *Nucleic Acids Res.*, **41**, D631–D635.
10. Waterhouse,R.M., Tegenfeldt,F., Li,J., Zdobnov,E.M. and Kriventseva,E.V. (2013) OrthoDB: a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Res.*, **41**, D358–D365.
11. Chen,F., Mackey,A.J., Stoeckert,C.J. and Roos,D.S. (2006) OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res.*, **34**, D363–D368.
12. Mi,H., Muruganujan,A. and Thomas,P.D. (2013) PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Res.*, **41**, D377–D386.
13. Huerta-Cepas,J., Capella-Gutiérrez,S., Pryszcz,L.P., Marcet-Houben,M. and Gabaldón,T. (2014) PhylomeDB v4: zooming into the plurality of evolutionary histories of a genome. *Nucleic Acids Res.*, **42**, D897–D902.
14. Proost,S., Van Bel,M., Sterck,L., Billiau,K., Van Parys,T., Van de Peer,Y. and Vandepoele,K. (2009) PLAZA: a comparative genomics resource to study gene and genome evolution in plants. *Plant Cell*, **21**, 3718–3731.
15. Altenhoff,A.M., Schneider,A., Gonnet,G.H. and Dessimoz,C. (2011) OMA 2011: orthology inference among 1000 complete genomes. *Nucleic Acids Res.*, **39**, D289–D294.
16. Altenhoff,A.M. and Dessimoz,C. (2009) Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput. Biol.*, **5**, e1000262.

17. Afrasiabi,C., Samad,B., Dineen,D., Meacham,C. and Sjölander,K. (2013) The PhyloFacts FAT-CAT web server: ortholog identification and function prediction using fast approximate tree classification. *Nucleic Acids Res.*, **41**, W242–W248.
18. Boeckmann,B., Robinson-Rechavi,M., Xenarios,I. and Dessimoz,C. (2011) Conceptual framework and pilot study to benchmark phylogenomic databases based on reference gene trees. *Brief. Bioinform.*, **12**, 423–435.
19. Linard,B., Thompson,J.D., Poch,O. and Lecompte,O. (2011) OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, **12**, 11.
20. Roth,A.C.J., Gonnet,G.H. and Dessimoz,C. (2008) Algorithm of OMA for large-scale orthology inference. *BMC Bioinformatics*, **9**, 518.
21. Dessimoz,C., Boeckmann,B., Roth,A.C.J. and Gonnet,G.H. (2006) Detecting non-orthology in the COGs database and other approaches grouping orthologs using genome-specific best hits. *Nucleic Acids Res.*, **34**, 3309–3316.
22. Altenhoff,A.M., Gil,M., Gonnet,G.H. and Dessimoz,C. (2013) Inferring hierarchical orthologous groups from orthologous gene pairs. *PLoS One*, **8**, e53786.
23. Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Michael Cherry,J., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.
24. Dimmer,E.C., Huntley,R.P., Alam-Faruque,Y., Sawford,T., O'Donovan,C., Martin,M.J., Bely,B., Browne,P., Chan,W.M., Eberhardt,R. *et al.* (2012) The UniProt-GO Annotation database in 2011. *Nucleic Acids Res.*, **40**, D565–D570.
25. Skunca,N., Bošnjak,M., Kriško,A., Panov,P., Džeroski,S., Smuc,T. and Supek,F. (2013) Phyletic profiling with cliques of orthologs is enhanced by signatures of paralogy relationships. *PLoS Comput. Biol.*, **9**, e1002852.
26. Kersey,P.J., Lawson,D., Birney,E., Derwent,P.S., Haimel,M., Herrero,J., Keenan,S., Kerhornou,A., Koscielny,G., Kähäri,A. *et al.* (2010) Ensembl Genomes: extending Ensembl across the taxonomic space. *Nucleic Acids Res.*, **38**, D563–D569.
27. Monaco,M.K., Stein,J., Naithani,S., Wei,S., Dharmawardhana,P., Kumari,S., Amarasinghe,V., Youens-Clark,K., Thomason,J., Preece,J. *et al.* (2014) Gramene 2013: comparative plant genomics resources. *Nucleic Acids Res.*, **42**, D1193–D1199.
28. Rouard,M., Guignon,V., Aluome,C., Laporte,M.-A., Droc,G., Walde,C., Zmasek,C.M., Périn,C. and Conte,M.G. (2011) GreenPhylDB v2.0: comparative and functional genomics in plants. *Nucleic Acids Res.*, **39**, D1095–D1102.
29. Van Bel,M., Proost,S., Wischnitzki,E., Movahedi,S., Scheerlinck,C., Van de Peer,Y. and Vandepoele,K. (2012) Dissecting plant genomes with the PLAZA comparative genomics platform. *Plant Physiol.*, **158**, 590–600.
30. Rensing,S.A., Lang,D., Zimmer,A.D., Terry,A., Salamov,A., Shapiro,H., Nishiyama,T., Perroud,P.-F., Lindquist,E.A., Kamisugi,Y. *et al.* (2008) The Physcomitrella genome reveals evolutionary insights into the conquest of land by plants. *Science*, **319**, 64–69.
31. International Wheat Genome Sequencing Consortium, Marcussen,T., Sandve,S.R., Heier,L., Spannagl,M., Pfeifer,M., Jakobsen,K.S., Wulff,B.B.H., Steuernagel,B., Mayer,K.F.X. *et al.* (2014) Ancient hybridizations among the ancestral genomes of bread wheat. *Science*, **345**, 1250092.
32. International Wheat Genome Sequencing Consortium (IWGSC) (2014) A chromosome-based draft sequence of the hexaploid bread wheat (Triticum aestivum) genome. *Science*, **345**, 1251788.
33. Luo,M.C., Deal,K.R., Akhunov,E.D., Akhunova,A.R., Anderson,O.D., Anderson,J.A., Blake,N., Clegg,M.T., Coleman-Derr,D., Conley,E.J. *et al.* (2009) Genome comparisons reveal a dominant mechanism of chromosome number reduction in grasses and accelerated genome evolution in Triticeae. *Proc. Natl Acad. Sci. U.S.A.*, **106**, 15780–15785.
34. Akhunov,E.D., Sehgal,S., Liang,H., Wang,S., Akhunova,A.R., Kaur,G., Li,W., Forrest,K.L., See,D., Simková,H. *et al.* (2013) Comparative analysis of syntenic genes in grass genomes reveals accelerated rates of gene structure and coding sequence evolution in polyploid wheat. *Plant Physiol.*, **161**, 252–265.
35. Choulet,F., Alberti,A., Theil,S., Glover,N., Barbe,V., Daron,J., Pingault,L., Sourdille,P., Couloux,A., Paux,E. *et al.* (2014) Structural and functional partitioning of bread wheat chromosome 3B. *Science*, **345**, 1249721.
36. McHale,L., Tan,X., Koehl,P. and Michelmore,R.W. (2006) Plant NBS-LRR proteins: adaptable guards. *Genome Biol.*, **7**, 212.
37. Lagercrantz,U. and Axelsson,T. (2000) Rapid evolution of the family of CONSTANS LIKE genes in plants. *Mol. Biol. Evol.*, **17**, 1499–1507.
38. Geddy,R. and Brown,G.G. (2007) Genes encoding pentatricopeptide repeat (PPR) proteins are not conserved in location in plant genomes and may be subject to diversifying selection. *BMC Genomics*, **8**, 130.
39. Byrne,K.P. and Wolfe,K.H. (2005) The Yeast Gene Order Browser: combining curated homology and syntenic context reveals gene fate in polyploid species. *Genome Res.*, **15**, 1456–1461.
40. Louis,A., Muffato,M. and Roest Crollius,H. (2013) Genomicus: five genome browsers for comparative genomics in eukaryota. *Nucleic Acids Res.*, **41**, D700–D705.
41. Steinbach,D., Alaux,M., Amselem,J., Choisne,N., Durand,S., Flores,R., Keliet,A.-O., Kimmel,E., Lapalu,N., Luyten,I. *et al.* (2013) GnpIS: an information system to integrate genetic and genomic data from plants and fungi. *Database* , **2013**, bat058.
42. Carrigan,M.A., Uryasev,O., Davis,R.P., Zhai,L., Hurley,T.D. and Benner,S.A. (2012) The natural history of class I primate alcohol dehydrogenases includes gene duplication, gene loss, and gene conversion. *PLoS One*, **7**, e41175.
43. Schmitt,T., Messina,D.N., Schreiber,F. and Sonnhammer,E.L.L. (2011) Letter to the editor: SeqXML and OrthoXML: standards for sequence and orthology information. *Brief. Bioinform.*, **12**, 485–488.
44. De Bie,T., Cristianini,N., Demuth,J.P. and Hahn,M.W. (2006) CAFE: a computational tool for the study of gene family evolution. *Bioinformatics*, **22**, 1269–1271.
45. Milinkovitch,M.C., Helaers,R., Depiereux,E., Tzika,A.C. and Gabaldón,T. (2010) 2X genomes–depth does matter. *Genome Biol.*, **11**, R16.
46. Bailey,J.A. and Eichler,E.E. (2006) Primate segmental duplications: crucibles of evolution, diversity and disease. *Nat. Rev. Genet.*, **7**, 552–564.
47. Szalkowski,A., Ledergerber,C., Krähenbühl,P. and Dessimoz,C. (2008) SWPS3 - fast multi-threaded vectorized Smith-Waterman for IBM Cell/B.E. and x86/SSE2. *BMC Res. Notes*, **1**, 107.
48. Wittwer,L.D., Piližota,I., Altenhoff,A.M. and Dessimoz,C. (2014) Speeding up all-against-all protein comparisons while maintaining sensitivity by considering subsequence-level homology. *PeerJ*, **2**, e607.
49. Dessimoz,C., Zoller,S., Manousaki,T., Qiu,H., Meyer,A. and Kuraku,S. (2011) Comparative genomics approach to detecting split-coding regions in a low-coverage genome: lessons from the chimaera Callorhinchus milii (Holocephali, Chondrichthyes). *Brief. Bioinform.*, **12**, 474–484.

D

CLUSTERING GENES OF COMMON EVOLUTIONARY HISTORY

The work presented in chapters 2 and 3 was published in the journal Molecular Biology and Evolution.

Clustering Genes of Common Evolutionary History

Kevin Gori,[1] Tomasz Suchan,[2] Nadir Alvarez,[2] Nick Goldman,*[,1] and Christophe Dessimoz*[,1,2,3,4,5,6]

[1]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Campus, Hinxton, United Kingdom

[2]Department of Ecology and Evolution, Biophore Building, UNIL-Sorge, University of Lausanne, Lausanne, Switzerland

[3]Department of Genetics, Evolution & Environment, University College London, London, United Kingdom

[4]Department of Computer Science, University College London, London, United Kingdom

[5]Centre for Integrative Genomics, University of Lausanne, Lausanne, Switzerland

[6]Swiss Institute of Bioinformatics, Biophore, Lausanne, Switzerland

***Corresponding author:** E-mail: goldman@ebi.ac.uk; c.dessimoz@ucl.ac.uk.

Associate editor: Arndt von Haeseler

Abstract

Phylogenetic inference can potentially result in a more accurate tree using data from multiple loci. However, if the loci are incongruent—due to events such as incomplete lineage sorting or horizontal gene transfer—it can be misleading to infer a single tree. To address this, many previous contributions have taken a mechanistic approach, by modeling specific processes. Alternatively, one can cluster loci without assuming how these incongruencies might arise. Such "process-agnostic" approaches typically infer a tree for each locus and cluster these. There are, however, many possible combinations of tree distance and clustering methods; their comparative performance in the context of tree incongruence is largely unknown. Furthermore, because standard model selection criteria such as AIC cannot be applied to problems with a variable number of topologies, the issue of inferring the optimal number of clusters is poorly understood. Here, we perform a large-scale simulation study of phylogenetic distances and clustering methods to infer loci of common evolutionary history. We observe that the best-performing combinations are distances accounting for branch lengths followed by spectral clustering or Ward's method. We also introduce two statistical tests to infer the optimal number of clusters and show that they strongly outperform the silhouette criterion, a general-purpose heuristic. We illustrate the usefulness of the approach by 1) identifying errors in a previous phylogenetic analysis of yeast species and 2) identifying topological incongruence among newly sequenced loci of the globeflower fly genus *Chiastocheta*. We release treeCl, a new program to cluster genes of common evolutionary history (http://git.io/treeCl).

Key words: **phylogeny, incongruence, clustering, process-agnostic, nonorthology, incomplete lineage sorting.**

Article

Introduction

Molecular phylogenetic methods infer the evolutionary history of homologous sequences. The techniques of molecular phylogenetics were developed in the analysis of individual protein sequences (Neyman 1971; Kashyap and Subas 1974), but due to the modern abundance of sequencing data it is increasingly common to infer trees by jointly analyzing sequences from multiple loci (Delsuc et al. 2005). By considering more data, multilocus analyses are expected to deliver better-resolved and less biased inferences by averaging out uncertainty over a greater amount of data (Pamilo and Nei 1988).

There are a number of methods for multilocus phylogenetic analysis (Bininda-Emonds et al. 2002; de Queiroz and Gatesy 2007; Liu et al. 2009). Many of these proceed by inferring the single evolutionary tree that best fits the entire data set. Such "averaging" over multiple loci presumes that these loci share a common evolutionary history. However, when a data set comprises multiple loci, the trees derived from individual loci have the potential to be incongruent (Jeffroy et al. 2006). A key question here is whether incongruence results from sampling error, or if it indicates a real underlying difference in the evolution of distinct genomic loci. If we build a single summary tree from multiple loci, we are implicitly assuming the former: that each locus is a noisy estimate of the same underlying tree.

Alternatively, we might expect different regions of a genome to have different histories (Leigh, Lapointe, et al. 2011), due to a variety of processes such as horizontal gene transfer (HGT), hybridization, incomplete lineage sorting (ILS), and recombination. If we believe such processes have occurred, then we should expect that the trees derived from different loci could be incongruent with one another. Consequently, "summary" trees inferred from the entire data set may be only partially representative or, in the worst case, not representative of the evolution of any locus. Because this is a systematic error, rather than noise, we cannot expect it to be reduced by adding more data (Philippe et al. 2011). If we believe there is real heterogeneity in the evolutionary process that produced the genomes, and incongruence is an indication of this, then we should look for ways of partitioning multilocus data into groups that are related by the same history (Bull et al. 1993; Huelsenbeck et al. 1994; Cunningham 1997; Waddell et al. 2000).

Many methods dealing with incongruence make explicit assumptions about its biological basis. Such "mechanistic" approaches have been developed to model HGT (Hallett and Lagergren 2001; Dessimoz et al. 2008; Abby et al. 2010), ILS (Rannala and Yang 2003; Heled and Drummond 2010), recombination (Kosakovsky Pond et al. 2006), gene duplication (GD) (Chen et al. 2000; Boussau et al. 2013), and combinations of processes such as combined ILS/GD models (Bansal et al. 2010; Doyon et al. 2010; Szöllősi and Daubin 2012). However, mechanistic approaches can be computationally prohibitive, and may not be robust to other unmodeled sources of incongruence.

We focus our attention on an alternative class of methods that we will describe as "process-agnostic." These aim to detect the existence and extent of any significant incongruence within a data set, without relying on any assumptions about its biological basis. Existing process-agnostic approaches take the form of statistical tests of incongruence (Planet 2006; Leigh, Lapointe, et al. 2011) and clustering approaches relying on partitioning data sets into groups that are cohesive and self-similar (Nye 2008; Leigh, Schliep, et al. 2011).

Nye's Tree of Trees (2008) summarizes the phylogenetic similarities among genes as another tree, termed a meta-tree, where a tip corresponds to a tree derived from multilocus data, and an internal node represents the consensus of its child trees. The meta-tree is inferred from intertree Robinson-Foulds (1981) distances using an algorithm analogous to neighbor joining (Saitou and Nei 1987).

Similarly, Conclustador (Leigh, Schliep, et al. 2011) uses intertree distances as a basis for clustering. Trees are compared using a novel Euclidean distance among bipartitions weighted by bootstrap support, and for clustering Leigh et al. use a version of the *k*-means algorithm and a spectral clustering method (Kaufman and Rousseeuw 1987; Zelnik-Manor and Perona 2004). A conceptually similar method is PhyBin (Newton and Newton 2013), which can either identify genes with topologically identical trees or perform hierarchical clustering on the Robinson-Foulds distance matrix between every tree.

Statistical binning (Mirarab et al. 2014) uses a graph-based algorithm to divide a set of genes into a number of approximately equal-sized bins of phylogenetically compatible genes (Warnow 1994). This has been used as a preprocessing step, with the bins subsequently used as input for coalescent species tree estimation; binning is shown to reduce run times, and to increase accuracy in the presence of ILS (Mirarab et al. 2014).

BUCKy (Ané et al. 2007; Larget et al. 2010) uses a Bayesian probabilistic framework to estimate a gene-to-tree map that assigns each gene to one of the $(2n-3)!!$ possible unrooted trees on n taxa (Felsenstein 2004). A Dirichlet process prior (Ferguson 1973; Antoniak 1974) is used to determine the total number of distinct trees represented by the gene-to-tree map.

These methods have in common that they each adopt a specific clustering procedure. There are, however, many potential distance measures and clustering algorithms, and we know almost nothing about their relative performance in identifying genes that share common evolutionary histories under plausible biological scenarios. For instance, the Robinson-Foulds distance used in Tree of Trees ignores any difference in branch lengths among trees, yet these might provide useful information in the context of ILS; the Dirichlet process prior in BUCKy tends to result in uneven cluster sizes (Ané et al. 2007), yet this might be suboptimal in the context of recombination. Furthermore, the problem of determining the optimal number of clusters remains poorly understood, with methods providing no, or only generic, solutions.

Table 1. Distance Metrics Investigated.

Distance Measure	Features Incorporated
Robinson-Foulds	Topology
Euclidean	Branch lengths
Geodesic	Topology and branch lengths

Table 2. Clustering Methods Investigated.

Clustering Method	Type	Implementation
Single linkage	Hierarchical	Fastcluster (Müllner 2013)
Complete linkage	Hierarchical	Fastcluster
Average linkage (UPGMA)	Hierarchical	Fastcluster
Ward's method	Hierarchical	Fastcluster
Spectral clustering (using *k*-means for the final clustering step)	Coordinate transform	Spectral clustering: Custom implementation in treeCl (after Zelnik-Manor and Perona 2004) *k*-means: Scikit-learn (Pedregosa et al. 2011)
MDS + *k*-means	Coordinate transform	Custom implementation in treeCl (after Torgerson 1952)
k-medoids	Partitioning around medoids	C Clustering Library (de Hoon et al. 2004)

Here, we present a survey of clustering methods to partition multilocus data sets into groups with consistent underlying phylogenies. Our aims are to investigate whether this is a viable approach to use to partition multilocus data in an evolutionarily meaningful way, and to measure the relative effectiveness of each method. Specifically, we test combinations of three distance measures between trees (table 1) and seven well-established clustering algorithms (table 2) on simulated and empirical sequence data.

We also introduce two likelihood ratio tests for inferring the optimal number of clusters. We test them extensively through simulations and show that they accurately recover the true number of clusters and outperform the silhouette criterion, a general-purpose heuristic.

We apply the best combination of tree distance, clustering method, and stopping criterion to two empirical data sets: alignments of 344 loci in 18 yeast taxa (Hess and Goldman 2011), and of 176 loci in 306 taxa derived from 7 species of *Chiastocheta* genus globeflower flies.

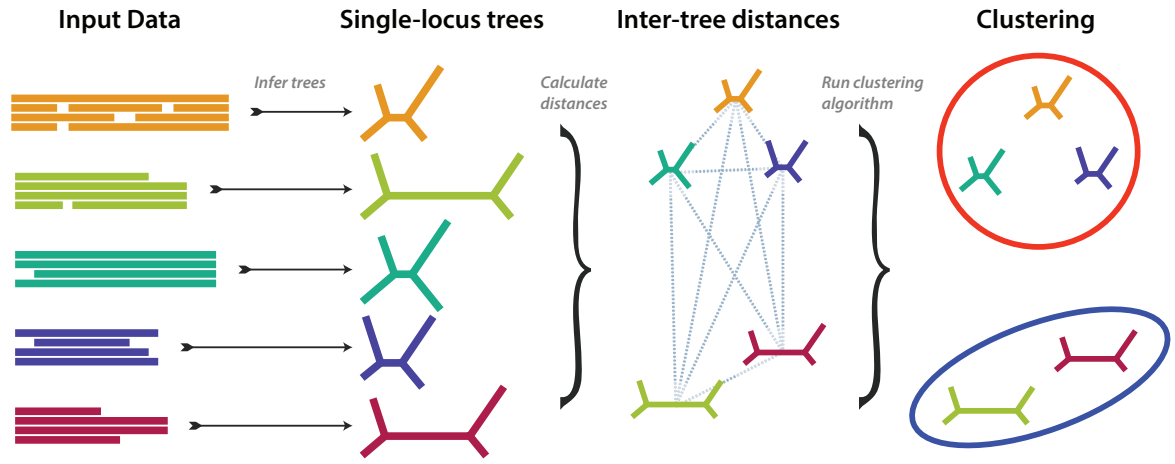


FIG. 1. Overview of the clustering process. From left to right: input alignments are read; trees are inferred from the alignments; intertree distances are computed and used as the basis for clustering. Further procedures are used to re-estimate one tree for each cluster and to choose the optimal number of clusters—see text for details.

Table 3. Attributes of the Four Simulated Data Set Scenarios with Incongruence Used to Test Combinations of Distance Metric and Clustering Method, and the Scenario Used to Test the Effect of Incomplete Occupancy.

Name	Taxa	Clusters	Loci	Distribution of Loci into Clusters
Small uniform	20	4	60	15, 15, 15, 15
Small skewed	20	4	60	5, 10, 15, 30
Large uniform	40	6	90	15, 15, 15, 15, 15, 15
Large skewed	40	6	90	5, 5, 10, 10, 20, 40
Incomplete occupancy	50	4	60	15, 15, 15, 15

The analyses were carried out using our new open source software package, treeCl, freely available at http://git.io/treeCl (last accessed March 1, 2016).

Results

The clustering approach investigated here takes a set of sequence alignments (one alignment per locus), and from them describes a partition of the data that divides the alignments into nonoverlapping subsets, each subset containing loci sharing a common phylogenetic history. Throughout this article we will describe such a division as a "partition," and the resulting subsets as "clusters." The approach is a three-step pipeline (fig. 1). First, we infer a separate phylogenetic tree for each input sequence alignment. Second, we gauge the level of evolutionary similarity among loci by measuring distances between pairs of trees. Third, we apply a clustering algorithm on the distances to generate a set of clusters. The number of clusters is either a fixed value decided a priori, or inferred from the data using tests introduced below.

In the following, we describe the results of a series of simulation experiments designed to explore the parameter space of the tree clustering approach and choose the most effective combinations of methods. We assess different stopping criteria for choosing the best-supported number of clusters from the data, again using simulation. Finally, we present the application of our method to data sets of yeast orthologs and of *Chiastocheta* genus globeflower flies.

Performance of the Combinations of Distance Metrics and Clustering Methods

Combinations of clustering methods (table 1) and distance metrics (table 2) were tested on simulated data over a range of conditions, described in Materials and Methods (table 3).

We investigated the performance of combinations of distance metrics and clustering methods for a fixed and known number of clusters. To assess the accuracy of each resulting partition, we computed the difference between the true partition (known from simulation) and the inferred partition using variation of information, an information-theoretic measure of the difference between two partitions of the same set (Meilă 2007). A variation of information value of zero is obtained when the two partitions are the same, and increasing positive values are obtained for partitions that are increasingly different.

Our results are summarized in figure 2. In terms of distance metrics, the performance using the Euclidean and geodesic distances is considerably better than Robinson-Foulds. Of these two, the geodesic distance performs marginally better than Euclidean. These conclusions hold for both skewed and uniform cluster size distributions, for the small and large data sets (supplementary figs. S1–S3, Supplementary Material on line), and for scenarios simulating both ILS (using nearest-neighbor interchange [NNI] rearrangements) and HGT (using subtree prune-and-regraft [SPR]).

In terms of clustering methods, the performance is worst using the simpler hierarchical methods—single linkage, complete linkage, and average linkage. Hierarchical clustering using Ward's criterion is more successful, but the best-performing methods are those involving embedding the distance matrix in a coordinate space: spectral and multidimensional scaling (MDS). However, MDS, as well as *k*-medoids, shows erratic behavior in some of the scenarios tested (supplementary figs.

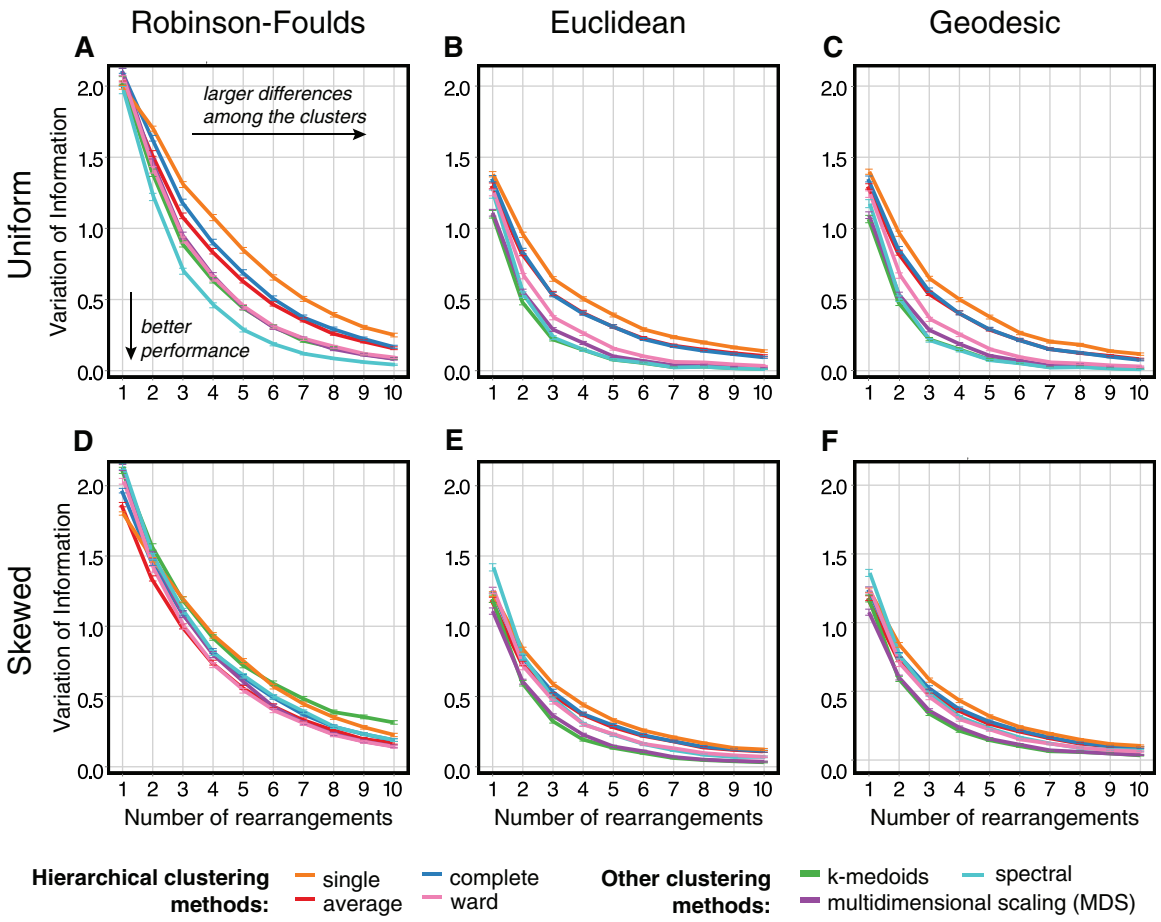


FIG. 2. The relative performances of combinations of distance metric (varying over columns of panels) and clustering methods (shown by the colors of the lines), as measured by the variation of information metric (*y*-axes; higher values show a larger departure from the correct solution). Lines show the mean value obtained from 1,000 replicates, and the error bars show the standard error of the mean. Rows correspond to the experiments with a partition of uniformly sized clusters (*A*–*C*) and those with a partition of clusters of skewed sizes (*D*–*F*). In each individual panel, the *x*-axis represents the number of NNI rearrangements separating the underlying clusters, so that increasing values along this axis correlate with the clustering problem becoming easier.

S1–S3, Supplementary Material online), and these were not considered for further analyses. Summarizing these observations, the combination of Euclidean or geodesic distances with spectral or Ward clustering seems to provide consistently the best overall performance across various conditions tested here. These combinations were used in our further analyses.

Performance of Methods for Determining the Number of Clusters

So far we have investigated performance with a known number of clusters, but this is typically unknown. To infer it, we devised two special-purpose likelihood ratio test procedures using empirical distributions of the test statistic: one a distribution derived from the input data via permutation, and the other derived via a parametric bootstrap resampling procedure (see Materials and Methods). We also compared these with a general-purpose "silhouette" criterion (Rousseeuw 1987). For a single point the silhouette value is the ratio of the mean of the distances to all other points in its cluster to the mean of the distances to all points in the nearest cluster. The silhouette score for the entire partition is the mean of these ratios over all points in the data set. The optimal number of clusters is inferred as the value for which the silhouette score is maximized.

For clarity, we first describe the results for a single set of sequences (one problem instance) before presenting our aggregate results. Given a problem instance, we repeat the clustering procedure with a varying number of clusters and compute the overall partition likelihood for each. Because specifying a greater number of clusters provides more freedom for the model to fit the data, the likelihood is expected to increase: this is generally what we observe. However, as in all likelihood ratio tests, the key consideration is by how much the likelihood must increase to warrant using the more complex model. To tackle this we generate empirical distributions of the likelihood increase from pseudoreplicate data derived from the data present in the instance, through the permutation and parametric bootstrap procedures described in

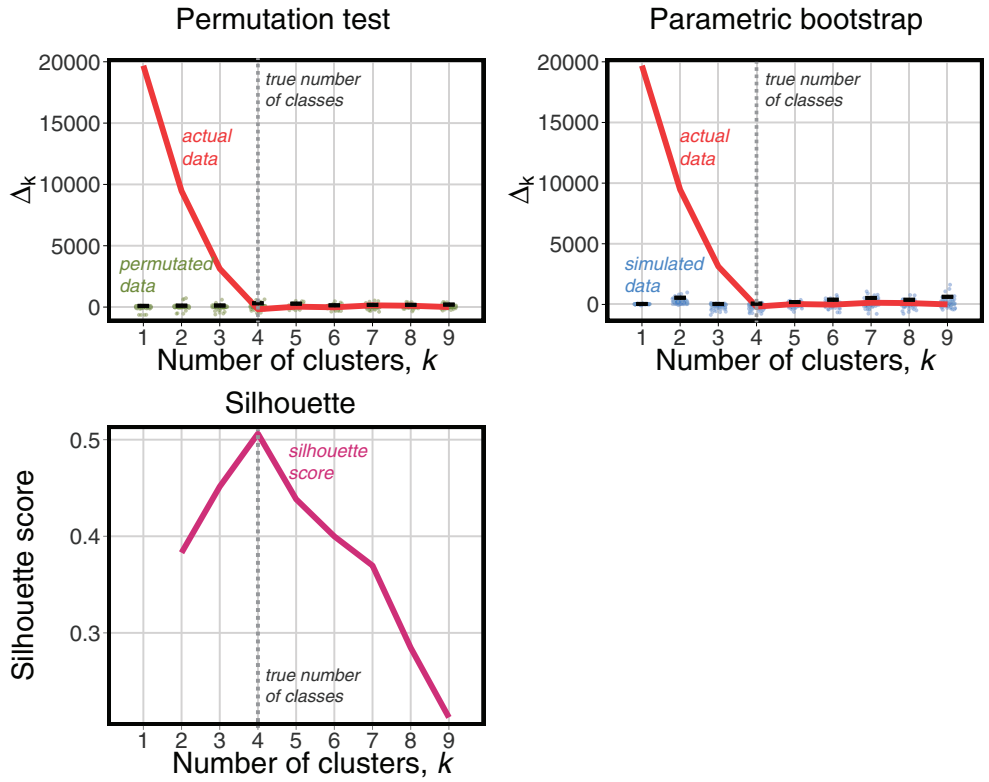


FIG. 3. Comparison of the criteria used to determine the number of clusters on a single problem instance—in this example, data simulated for 60 loci belonging to 4 clusters, each of size 15, with the clusters' trees separated by 1 SPR. As the proposed number of clusters increases, the likelihood increases, which is expected because of the greater number of free parameters in the model. (*A*) Permutation test: the improvement in likelihood for each additional cluster (red curve) is significantly greater than that observed for permuted data sets (green dots show the distribution of values over 100 permutations) until the comparison between four and five clusters is reached, correctly implying that the use of four clusters is optimal. (*B*) Parametric bootstrap test: again, the improvement for each additional cluster (red curve) is significantly greater than that for data sets simulated for one fewer cluster (blue dots) until the true number of clusters (four) has been reached. (*C*) Silhouette score: the general-purpose silhouette stopping criterion has its maximum at the true value of 4. We note that in this instance, comprising a single data set from one simulation design, the three methods agree on the true answer.

Materials and Methods. The likelihood increase from the original data is compared with the expected increase from the empirical distribution to determine significance (fig. 3 and supplementary fig. S4, Supplementary Material online).

Let us now consider the results over multiple problem instances. We simulated data sets using the procedure corresponding to the "small uniform" setup (see Simulating Data Sets with Incongruence), with two levels of difficulty: we generated 100 data sets from trees separated by 1 SPR move (referred to as "difficult"), and 100 separated by 5 SPR moves ("moderate"). Each data set was analyzed under the four combinations of Euclidean or geodesic distances with spectral or Ward's method clustering. This resulted in a total of 800 problem instances.

To investigate the overall performance of the three stopping criteria, we first consider the aggregate results for all 400 difficult and 400 moderate problem instances, that is, 100 each under all four combinations of distance metric and clustering procedure (fig. 4). For both the difficult and moderate cases, the distribution of the number of clusters chosen is centered on the true value, 4, for all three criteria. However, in the difficult case, the distributions of the permutation and bootstrap tests are much tighter than the silhouette score, indicating that these two stopping criteria make correct calls more often. The results are consistent in the moderate case, although the differences between criteria are smaller, with all of them making many more correct calls (supplementary fig. S5*E*–*H*, Supplementary Material online).

We also considered the performance of the stopping criteria separately for the different distance metrics and clustering methods. In terms of distance metrics, we see little difference between geodesic and Euclidean distances. In contrast, we observe that all three stopping criteria perform noticeably better in combination with spectral clustering than with Ward's method (supplementary fig. S5, Supplementary Material online). This is particularly the case for our two new criteria (permutation and bootstrap), which outperform the silhouette by a greater margin on the spectral clustering runs.

Dealing with Incomplete Occupancy across Loci

In the simulations considered so far, we have covered cases in which there has been no missing data. When analyzing real data, we cannot guarantee that all loci will be present for all

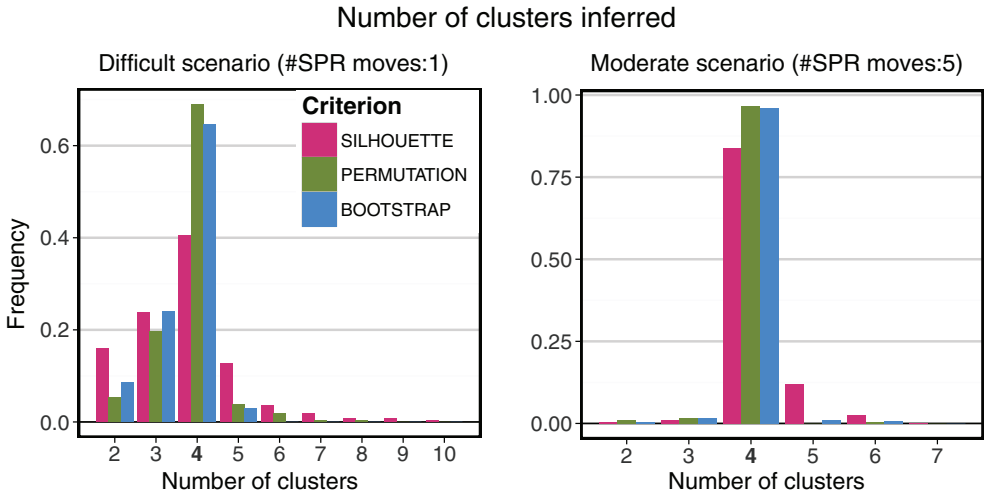


FIG. 4. Aggregate results for 400 difficult problem instances (left) and 400 moderate instances (right). The true number of clusters is 4. In both sets, our new stopping criteria (permutation and bootstrap) perform better than the general-purpose silhouette method.

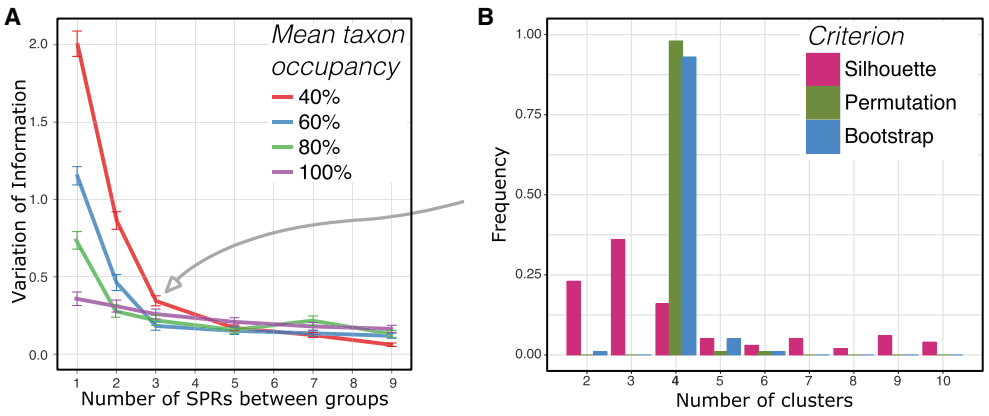


FIG. 5. (*A*) Distance of the spectral clustering of geodesic distances from the "true" clustering for varying levels of taxon occupancy. Just as with complete groups, partial groups converge to the correct assignment as the distance between clusters increases. When clusters differ from the underlying species tree by three SPRs or more, the effect of incomplete occupancy on performance is very slight. (*B*) Effect of incomplete taxon occupancy on cluster number selection criteria. Nonparametric permutation and parametric bootstrap recover the true number of clusters (four) in more than 90% of cases. The clusters were separated by three SPRs, and each locus had 40% mean taxon occupancy, which corresponds to the point on panel (*A*) indicated by the gray arrow.

taxa. The effect that missing data have on our method is that we are required to compare trees with different leaf sets, a circumstance for which distance metrics have not been defined. A simple measure to counteract this is to prune trees to the intersection of their taxon sets, and then measure the distance between these reduced trees.

To assess the impact of incomplete occupancy on our approach's ability to infer the correct clusters, we generated additional simulated data sets containing a varying proportion of randomly selected missing genes (see Materials and Methods) and analyzed the data using the best combination of distance measure and clustering method (geodesic distances and spectral clustering). With missing data, when the number of clusters is known in advance, the true partition of the data is recovered with high accuracy (measured by variation of information) as long as the clusters are separated by a few topological rearrangements—even when data are sparse (fig. 5*A*). When clusters are not well separated—differing by just 1 or 2 SPRs—sparseness has a detrimental effect on accuracy. Both of the permutation and bootstrap stopping criteria show high accuracy when inferring the number of clusters, strongly outperforming the silhouette (fig. 5*B* and supplementary fig. S6, Supplementary Material online).

Application to Empirical Data

We applied the best combination of distance measure (geodesic distance), clustering method (spectral clustering), and stopping criterion (permutation test) to two empirical data sets.

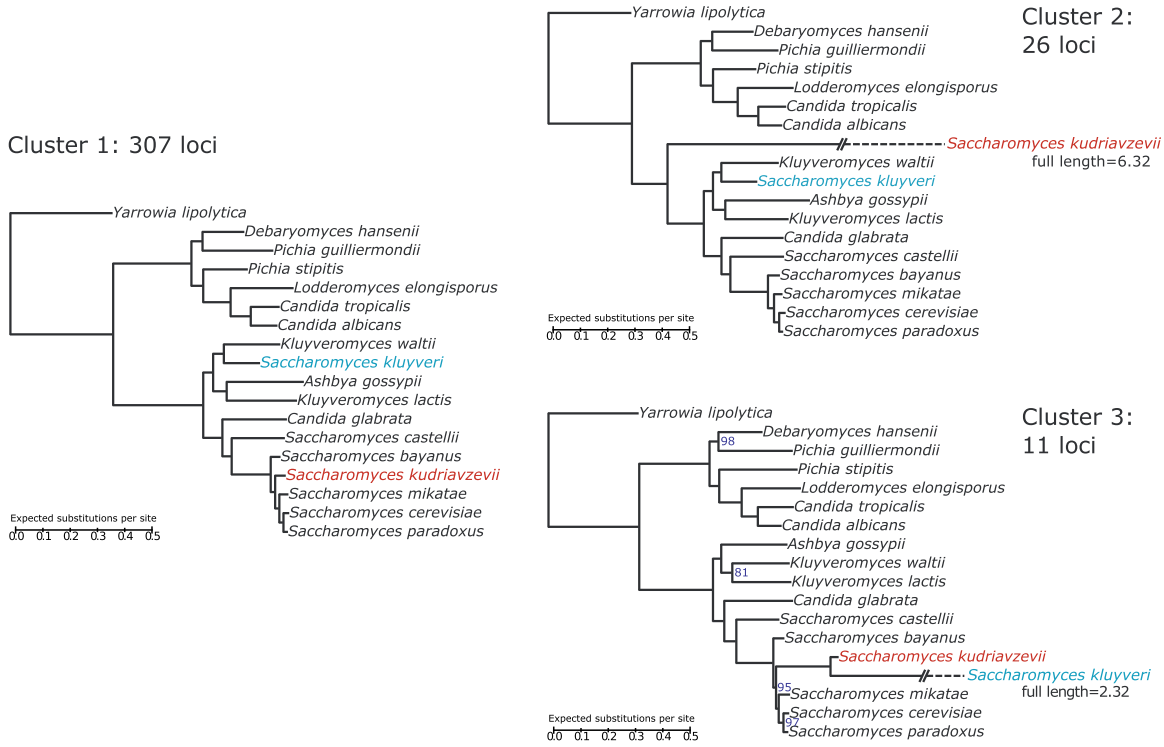


Fig. 6. Phylogenetic trees inferred from the three clusters found in the yeast analysis with treeCl. The tree on the left is that inferred from the largest cluster of 307 loci. This matches the established species tree for these 18 species of yeast. The taxa highlighted in red (*Saccharomyces kudriavzevii*) and blue (*Saccharomyces kluyveri*) are those that are found on long branches in the trees inferred from clusters 2 and 3 (shown respectively right, upper, and right, lower). In these trees, the branches leading to *S. kudriavzevii* (in cluster 2) and *S. kluyveri* (in cluster 3) have been truncated so as to fit reasonably on the plot. Their full lengths are as indicated. Otherwise, branch lengths can be determined by the scale bars shown (all equal scales). Branch support measures were calculated using approximate Bayes (aBayes). Where aBayes branch supports are less than the maximum possible value of 100%, their values are indicated by a number to the right of the branch.

Yeast Data Set

The first empirical data set consists of 344 curated orthologous sets of genes from 18 ascomycetous yeast species, which was previously used to infer a species phylogeny robust to intergene heterogeneities (Hess and Goldman 2011). Applying our method to this data set resulted in a partition of the 344 loci into 3 clusters (supplementary fig. S7, Supplementary Material online). The clusters are of unequal sizes: there is a large cluster, consisting of 307 loci, and two small clusters containing 26 loci and 11 loci. Although the numbering of clusters produced by treeCl has no special meaning, for clarity "cluster 1" will consistently refer to the cluster of 307 loci, and "cluster 2" and "cluster 3" to the clusters of 26 and 11 loci, respectively.

Despite the high degree of incongruence among trees estimated from individual loci, the overall species tree relating these yeasts has been well-studied, and has been established with little controversy (Dujon 2010). This species tree can be seen as the tree on the left in figure 6, which is also the cluster tree derived from cluster 1. The trees on the right of figure 6 are the cluster trees inferred for clusters 2 and 3.

The tree for cluster 2 yields nearly the same topology as that for cluster 1, with the sole modification that *Saccharomyces kudriavzevii* appears basal to, rather than within, the *Saccharomyces sensu stricto* clade. Branch lengths are also modified in the cluster 2 tree: minor changes aside, note that the branch leading to *S. kudriavzevii* is very much longer than in the cluster 1 tree.

A similar observation can be made of the inferred tree from cluster 3. In this case, it is *Saccharomyces kluyveri* that is incorrectly placed relative to the species tree, again with a very long branch. The cluster 3 tree also differs from the cluster 1 tree in the arrangement of the clade consisting of the species *Kluyveromyces waltii*, *Ashbya gossypii*, and *Kluyveromyces lactis*, the clade to which *S. kluyveri* belongs in the other two trees. The cluster 3 tree is also the only one for which the branch support values, as measured using approximate Bayes, are below 100%. The lowest branch support, 81%, is found within the rearranged *K. waltii*, *A. gossypii*, and *K. lactis* clade. With this exception, the remaining branches all show greater than 95% approximate Bayes branch support, even though there is incongruence among the loci underlying these trees. However, this may not necessarily be a strong case for these topologies being correct, as it has been suggested that concatenation tends to inflate branch support values (Larget et al. 2010; Weisrock et al. 2012).

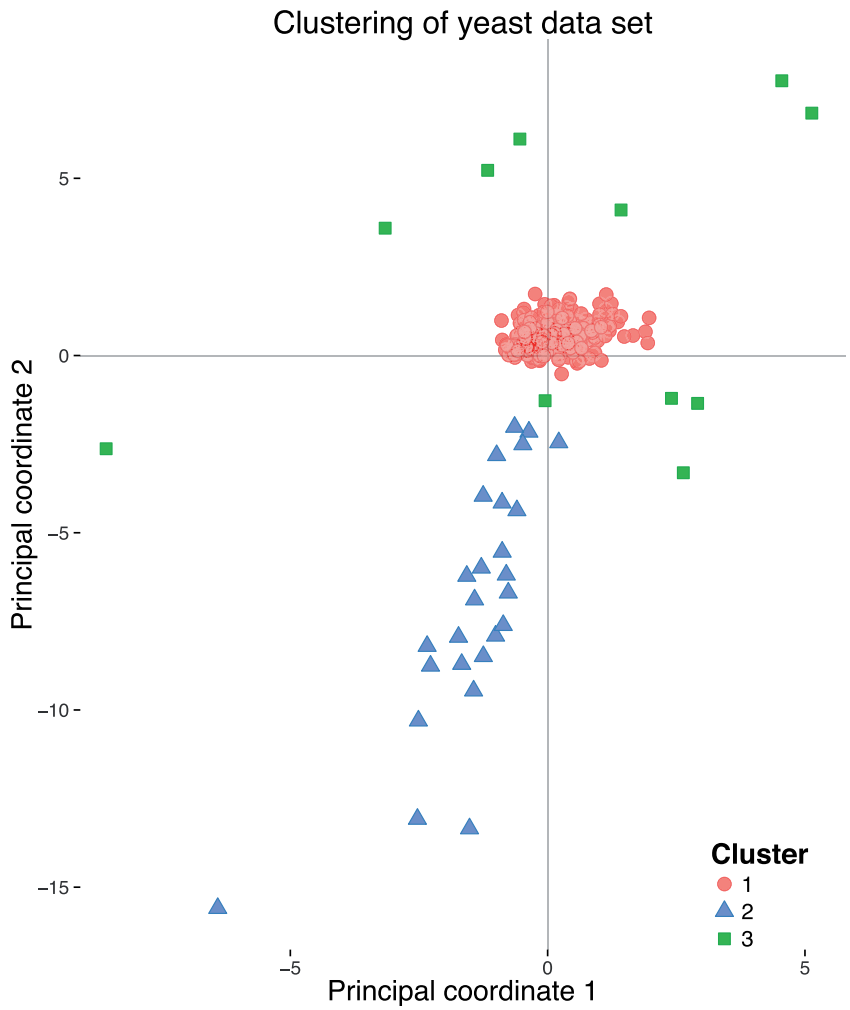


Fig. 7. Visualization of application of treeCl to the yeast data set. The scatterplot shows the embedding, by MDS, of the geodesic distances between the 344 trees. Three clusters were found by spectral clustering: red circles indicate the largest cluster, with 307 members; the 37 remaining loci are indicated by blue triangles (cluster 2) and green squares (cluster 3). Loci belonging to the first, largest cluster are tightly grouped and yield the correct species phylogeny, whereas trees belonging to the second and third clusters are disparate and all have odd and inconsistent phylogenies as a result of incorrectly called orthology (see text for full details).

As an attempt to visualize the distribution of the individual locus trees, we embedded them in two-dimensional space using MDS (fig. 7). In this representation, cluster 1 appears as a very tight cluster of points in the center of the figure, while clusters 2 and 3 are more diffuse. Although clusters 1 and 3 appear to overlap, keep in mind that while it may seem to be difficult to assign these clusters on the basis of this figure, the actual clustering is done in a higher dimensional space and using a different coordinate transform than the one visualized here. What can be noted from this figure is that all members of clusters 2 and 3 are positioned relatively large distances away from cluster 1, which suggests that these clusters consist of loci for which the underlying tree distances are large, when measured from those loci from cluster 1.

To try to understand the source of incongruence in the smaller clusters, we examined the sequences associated with the long branch in the gene tree associated with each of their 37 loci. They each included one particularly long terminal branch, but none of the single-locus topologies matched the ones inferred for cluster 2 or cluster 3 as a whole. The 37 trees are reproduced in supplementary figure S8, Supplementary Material online. Reciprocal best hit analyses of these sequences with *Saccharomyces cerevisiae* indicate that they were erroneously classified as orthologs (supplementary table S1, Supplementary Material online). We thus conclude that the major source of incongruence in this 344 gene yeast data set is derived from erroneous orthology calling, particularly involving the *S. kudriavzevii* and *S. kluyveri* genomes. In this example, treeCl has identified 307 loci that support the species tree; of 37 that do not, it has detected two clusters, one primarily consisting of cases where the *S. kudriavzevii* gene has been misannotated and one where

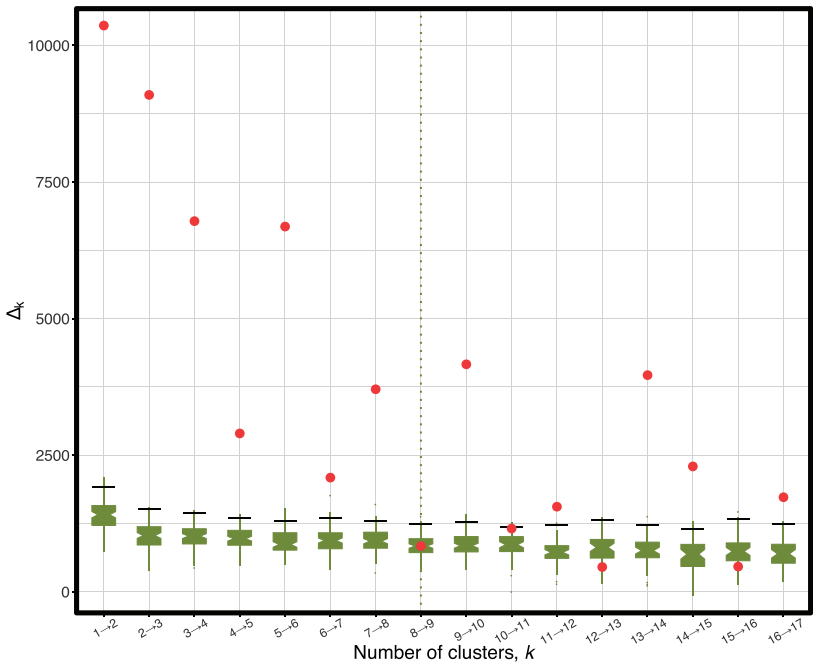


Fig. 8. Likelihood improvement gained when partitioning the *Chiastocheta* data into increasing numbers of clusters (red points). Resampled distributions (boxplots) were generated using the permutation procedure. The number of clusters selected by the stopping criterion is indicated by the vertical dashed line. For two to eight clusters, the improvement is statistically significant; increasing to nine clusters is not.

S. kluyveri misannotations are similarly implicated. Even for these two clusters, the inferred phylogeny agrees fully or very nearly with the species tree, aside from the position of the primary misannotated species.

Given the "outlier" nature of the loci identified in the small cluster, we also applied a specialized outlier detection package, kdetrees (Weyenberg et al. 2014). Remarkably, with geodesic distances, it identified the exact same 37 loci as outliers (supplementary fig. S9, Supplementary Material online). This provides additional evidence that these 37 loci should indeed be excluded in the inference of the species tree.

Chiastocheta Data Set

The globeflower flies, genus *Chiastocheta*, are pollinators and seed parasites of the plant species from the *Trollius* genus (Ranunculaceae) (Pellmyr 1992; Suchan et al. 2015). *Chiastocheta* have a recent origin, with most diversification events occurring less than ca. 1.6 Ma, and their phylogenetic relationships are uncertain (Després et al. 2002; Espíndola et al. 2012). Particularly, only two globeflower fly species were found to be phylogenetically supported using mitochondrial markers (Espíndola et al. 2012).

RAD-sequencing of 306 samples from 7 European *Chiastocheta* species (25 *Chiastocheta dentifera* individuals, 48 *Chiastocheta inermella*, 52 *Chiastocheta lophota*, 34 *Chiastocheta macropyga*, 70 *Chiastocheta rotundiventris*, 36 *Chiastocheta setifera*, and 41 *Chiastocheta trollii*) collected across their whole ranges yielded a data matrix of 5,574 orthologous sets of sequences (loci), containing in total 253,866 variable, and 81,379 parsimony informative sites. Because of inherent technical limitations of RAD-sequencing, the majority of these loci had sparse coverage over the individuals. To focus on the phylogenetically most informative loci, we disregarded loci present in fewer than 100 individuals. This resulted in a matrix of 176 loci (i.e., 10.2% of the overall number of loci identified). Each locus contained, on average, 44.2% of the taxon set.

Application of treeCl (with geodesic distance, spectral clustering, and permutation test stopping criterion) identified eight clusters. However, the plot of the likelihood improvement against the number of clusters (fig. 8) is not smooth: most of the improvement is obtained by increasing the number of clusters up to four and by increasing it from five to six; in contrast, adding a fifth or seventh cluster only moderately improves the fit. Thus, a cautious interpretation of this analysis is that there are at least four distinct clusters of loci. This conclusion is also supported by the parametric bootstrap criterion (supplementary fig. S10, Supplementary Material online).

The trees inferred for the four clusters (fig. 9) substantially differ, both in topology and branch lengths. In particular, many of the deep relationships are well-resolved but different across clusters, suggesting genuine differences in the history of the loci. However, with very few exceptions, each species forms a distinct monophyletic group. This is consistent with well-documented differences in genital morphology across most of these species (Després et al. 2002). With greater data available, phylogeny and morphology now agree.

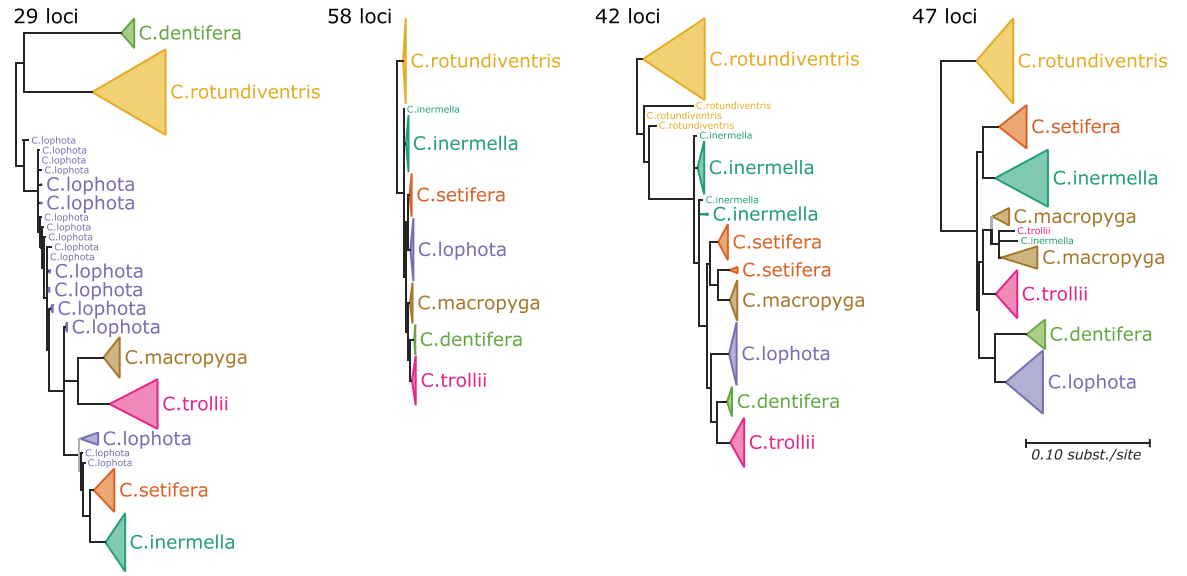


Fig. 9. Trees obtained when clustering RAD-seq data from globeflower flies of the genus *Chiastocheta*. The trees are drawn to scale, and are rooted at their midpoint, as the outgroup is unknown. Leaves are colored according to species membership. Branch support is indicated as follows: branches with support values below 0.9 are collapsed into multifurcations; those with support in the range 0.9–0.95 are colored gray; those with support >0.95 are colored black. Support values are calculated using approximate Bayes (Anisimova et al. 2011).

Furthermore, the even cluster size distribution (cluster sizes of 29, 58, 42, and 47 loci) suggests that the method is not simply finding groups that consist of one or two outliers. The greatest departure from monophyly is shown in the group consisting of 29 loci. In this group, the majority of the representatives of species *C. lophota* are found at the base of a clade that also contains *C. macropyga*, *C. trollii*, *C. setifera*, and *C. inermella*. For partitions into greater numbers of clusters than four, we observe at least one tree in which species monophyly is largely absent (supplementary fig. S11, Supplementary Material online), which may indicate that likelihood improvements gained when clustering into more than four groups are due to fitting to the noise in the data, extracting loci with weak or conflicting signal. In this case, attempting to visualize the individual locus trees in two-dimensional space does not yield informative results (supplementary fig. S12, Supplementary Material online).

Overall, the picture that emerges from the analysis confirms the existence of seven distinct species in the *Chiastocheta* genus, but implies that the branching order among them varies substantially across loci. Such variation is suggestive of ILS, particularly as six of the seven species (except *C. rotundiventris*) are thought to have radiated more or less synchronously (Espíndola et al. 2012). To rigorously test this hypothesis, future work could assess the fit of this data under a mechanistic model of ILS.

Discussion

In this study, we investigated clustering multilocus data sets into evolutionarily similar groups based on their inferred phylogenies. This work is motivated by the observation that phylogenetic incongruence among loci can arise through various evolutionary processes, in which case a single tree is insufficient to describe the disparate processes underlying the data. At the other extreme, reporting one tree for each locus suffers from the drawbacks of single-locus phylogenetics—lack of signal, sampling error, unrepresentativeness—and in addition it is difficult to interpret a large and unwieldy collection of trees. By clustering loci, we allow the possibility that a meaningful representation be given by some intermediate number of trees, each capturing a common evolutionary history for some of the loci. We do this in a process-agnostic way, in that we do not seek to view our observations through the lens of any particular mechanism. This may lose inferential power in the case where organisms have evolved mainly through a process that we fail to model explicitly, but has the advantage that we will not bias the analysis by imposing mathematical models inappropriate for the processes that have occurred.

To investigate the performance of this approach, we assessed combinations of different distance metrics and clustering methods using simulation. Overall, Euclidean and geodesic distances, which take branch lengths into account, performed better than Robinson-Foulds distances. Spectral clustering and Ward's method gave the best clusters, most reliably over the range of simulations analyzed. We note that two methods—MDS and *k*-medoids—are successful in many cases, but produce some anomalous results in which performance becomes worse as the problems become easier (supplementary figs. 1*D*, 2*A*, and 3*E*, Supplementary Material online).

We introduced new statistical tests to determine the best-supported number of clusters, and compared them with a general-purpose cluster assessment statistic. In simulation the new measures outperformed the general-purpose criterion. If we look at the results from the difficult case, it seems that all

criteria have a tendency to be conservative and underestimate the number of clusters (fig. 4 and supplementary fig. S5, Supplementary Material online). We consider this a valuable feature; it is more parsimonious to erroneously infer too few rather than too many clusters. When moving from simulated data to a real data set of 344 orthologous groups from yeast (Hess and Goldman 2011), subtle errors in orthology inference could be detected and corrected. This highlights the high potential of the approach for quality control in multilocus phylogenetic analyses. Furthermore, the unexpected nature of the errors observed in that data set is a good illustration of the flexibility of process-agnostic methods for detecting incongruence. The examination of a large data set of *Chiastocheta* flies demonstrates that our method is applicable to data sets of the scale that is routinely produced by high-throughput sequencing approaches such as RAD-seq, and not only to more artificial simulations.

The range of methods and conditions investigated in this study is considerable, but inevitably not exhaustive. There are other distance metrics and clustering methods not tested here. These were omitted mainly for reasons of being too numerous for their inclusion to be practical. Some were not considered because they overlapped closely with metrics and methods that were considered: for instance, kernel Principal Components Analysis (PCA) is a coordinate transformation procedure that could have been used in a similar way to spectral embedding and MDS; however, it is largely analogous to spectral embedding (Ng et al. 2001) and initial investigation showed it to give very similar results. Other clustering methods such as Markov Clustering (Enright et al. 2002), DBScan (Ester et al. 1996), and Affinity Propagation (Frey and Dueck 2007) were not investigated as they provide no means to specify the number of clusters they return, which is a property we specifically wanted so we could test our stopping criteria. Similar concerns led to us to exclude such distance measures as Quartet Distance (Estabrook et al. 1985) or Matching (Lin et al. 2012) as they provide discrete topology-only measures similar to Robinson-Foulds. Tree edit measures such as the SPR distance are highly computationally difficult to calculate (Bordewich and Semple 2005) and so were not investigated. This method may become tractable with the advent of fast approximation algorithms (Chung et al. 2013; Whidden et al. 2013).

We tested our method under a range of simulation criteria. However, the combinatorics of the range of parameters that can be varied are such that it was not possible to test them all. This also limits the degree to which we can test whether tuning certain clustering procedures might improve their performance (for instance, the number of dimensions to embed the intertree distances in when using MDS). Likewise, many biological phenomena leading to incongruence were not investigated (including variation in rate of evolution across genes and between taxa, differential duplication and loss between species within gene families, etc.). Nevertheless, we think that the variety of problems studied, and in particular the range of levels of difficulty, are enough to provide convincing evidence that process-agnostic clustering methods can work effectively and give useful results.

The clustering methods investigated in this work are also applicable to data sets with incomplete "occupancy" among species, such as the one obtained for *Chiastocheta* flies by RAD sequencing, a technique that is typically prone to having a large proportion of missing data. Indeed, our simulations suggest that as long as the clusters are separated by a few topological moves, occupancy as low as 40% incurs negligible performance degradation. Likewise, the new stopping criteria introduced in this study cope well with sparse data matrices, in contrast to the general-purpose silhouette method.

It is unclear how sensitive the method is to the quality of the inferred single-locus trees. Inferring these is the first step in our analyses, and all further steps proceed as if the trees are correct; the distance matrix is calculated based on these initial trees, which are not re-estimated. To improve our approach we could introduce a cycle in our algorithm in which the single-locus trees are re-estimated based on parameters estimated while inferring the cluster trees, and the distance matrix and cluster assignments updated. However, this is likely to be computationally expensive. Another possibility is to incorporate measures of phylogenetic uncertainty—such as the bootstrap—into the distance estimation and the clustering step.

Practically, however, the amount of computation required to apply distance metrics and clustering methods to whole-genome-scale data poses a challenge. For instance, calculating geodesic distances takes time of order $O(n^4)$ (Owen and Provan 2011), where n is the number of leaves in the tree, while Euclidean and Robinson-Foulds distances can be computed in linear time (Pattengale et al. 2007). There is also the burden of pruning trees to their overlapping taxa. These factors could prove prohibitive in the case of very large trees. Whatever the details of the distance calculations, they must be performed $\binom{m}{2}$ times, where m is the number of loci in the data set. Clustering the resulting $m \times m$ distance matrix using any spectral technique—requiring eigen decomposition—takes time of order $O(m^3)$. This burden can be reduced by applying an approximation such as the Nyström method (Fowlkes et al. 2004), which produces approximations to the eigenvalues and eigenvectors from a reduced input set, reducing the number of pairwise tree distance comparisons required. We have demonstrated that the relatively efficient Euclidean distance and Ward's method for hierarchical clustering produce good results, and may thus be preferred in large data sets. In the work carried out in this article, by far the largest amount of time is spent in tree inference; this remains the bottleneck.

We applied our method to two empirical data sets, one from yeasts and one from *Chiastocheta* flies. Both data sets show a high degree of phylogenetic incongruence, although this is likely to be for different reasons: misannotated orthology for the yeast data set, and ILS for *Chiastocheta*. Due to its process-agnostic nature, we were able to apply our method in the same way to both data sets, and learn something about the incongruent signals in the data. This allows us to identify the likely processes at play, and prioritize different types of

follow-up analysis—stringent orthology identification in the first case, and analysis under a mechanistic ILS model in the second. In this way, our process-agnostic is complementary, rather than in opposition, to mechanistic models of incongruence.

Looking ahead, it seems clear that the assumption in multilocus phylogenetics that all loci are derived from the same tree is too strong, and should be relaxed. Partitioning model parameters is commonplace (Hess and Goldman 2011; Lanfear et al. 2012); tree-topology partitioning is a logical next step.

Materials and Methods

In the following sections, we first describe the components of this clustering process in more detail, including the various distance and clustering algorithms investigated in this study. Next, we describe a partition likelihood quality score that we use to compare the performance of combinations of distances and clustering methods, and introduce new tests to infer the optimal number of clusters in a data set. Finally, we describe the simulated and empirical data used in our analyses. The analyses were carried out using our treeCl software, which is available as an open source python package (http://git.io/treeCl).

Input Data

The input data are a set of multiple sequence alignments, one per locus being examined. The sequences can be of nucleotides or proteins.

Tree Inference

In principle, any method of tree estimation can be used. We use maximum likelihood (ML) estimation of phylogenies, which is statistically robust (Felsenstein 2004) and enables us to use a likelihood criterion for cluster membership comparisons and cluster number decisions. For each locus, we infer the ML phylogenetic tree using the Phylogenetic Likelihood Library (PLL) (Flouri et al. 2015).

In the experiments described in this article, we use PLL's full ML estimation with tree search. We use either the General Time Reversible model (GTR) model (Tavaré 1986) for nucleotide data or the Whelan and Goldman model (WAG; Whelan and Goldman 2001) for proteins, coupled with a gamma distributed model of rate variation with four discrete categories (Yang 1994), and the RAxML search strategy (Stamatakis 2014).

Intertree Distances

Once the tree for each locus has been estimated, their similarities are assessed according to a particular distance metric. We have investigated three distance measures: Robinson-Foulds (Robinson and Foulds 1981), Euclidean (Kuhner and Felsenstein 1994), and geodesic (Billera et al. 2001) (table 1). With a set of m trees we compute all $m(m-1)/2$ pairwise distances. We implemented the tree distance algorithms in C++ and Python. The geodesic distance algorithm used is that of Owen and Provan (2011). Source code is available from https://pypi.python.org/pypi/tree_distance/0.0.6 (last accessed March 1, 2016).

Missing Data

For pairwise tree comparisons when taxon sets differ, the trees are pruned to the taxa they have in common. Distances are calculated on the resulting reduced trees. In the case that the intersection of taxon sets contains fewer than four taxa—the minimum number required that can produce a tree with at least one internal edge—the distance is taken to be zero.

Clustering

The resulting distance matrix is used as the input for a clustering algorithm. We have investigated seven such algorithms, detailed in table 2. Each algorithm presumes that the required number of clusters is known in advance; we investigate approaches for choosing the optimal number of clusters below. All methods work directly on the distance matrix, except the coordinate transform methods. These transform the distance matrix into the coordinates of a set of points, then use k-means to perform the final clustering step. k-means is not suitable for use directly on a distance matrix.

Partition Likelihood for Assessing Clustering

In order to assess partitions, which may be obtained from different clustering approaches, we describe the "partition likelihood." This can be used as a quality score to assess the best combination of distance and clustering method.

Each cluster comprises a subset of the loci, and is a collection of genes putatively sharing a common evolutionary history. Hoping to benefit from a more robust evolutionary inference by combining the data from homogeneous sources, we therefore concatenate the alignments of the member loci and infer the ML tree using the same model as for the individual loci. The log likelihood is calculated for each cluster tree conditioned on the concatenated cluster alignment. The partition log likelihood, L^P, is the sum of all optimal cluster log likelihoods, and is in effect the maximum log likelihood under a model where the genes within each cluster share a common evolutionary history and evolutionary dynamics, but there are no constraints that different clusters share any evolutionary parameters.

Choice of Number of Clusters

The number of clusters, k, can take any integer value in the interval $[1, m]$, where m is the number of loci in the data set. Let us consider the case of choosing between k and $k + 1$ clusters. This is equivalent to choosing between the hypotheses that the loci are sampled from k evolutionary trees, or $k + 1$ evolutionary trees. These form our null and alternative hypotheses, respectively. The alternative hypothesis is able to recapitulate the null model, and therefore the hypotheses are nested. To illustrate that the alternative hypothesis nests the null, consider that if two of the trees associated with clusters in the alternative model are identical it is equivalent to the case that those clusters are combined, decreasing the effective number of clusters by one and reproducing the null. We can thus calculate the partition log likelihood of each hypothesis, L^P_k and $L^P_k + 1$, and the increase in log likelihood, $\Delta_k = L^P_{k+1} - L^P_k$.

With nested hypotheses, $2\Delta_k$ is asymptotically chi-squared distributed, with the number of degrees of freedom corresponding to the difference in the number of parameters between the null and alternative hypotheses (Wilks 1938). However, counting parameters prove difficult in this case: the extra parameters in the alternative hypothesis include an inferred tree topology, and tree topology parameters are difficult to quantify (Goldman 1993). This means we cannot specify which chi-squared distribution we should use for our test. This also precludes the application of information criteria such as the Akaike Information Criterion (AIC) and Bayes Information Criterion (BIC) (Akaike 1974; Schwarz 1978).

Alternatively, we can estimate the distribution of Δ_k by repeatedly calculating Δ_k values from new data sets generated under the null hypothesis. Such a procedure does not require that the difference in degrees of freedom be known or even that the hypotheses be nested. We devised two such procedures: a nonparametric permutation test, in which the new data sets are produced by randomizing the original data, and a parametric bootstrap test, in which new data sets are generated via simulation. These permit us to compare whether $k + 1$ clusters are statistically supported over k clusters, and we apply such tests successively for $k = 1, 2, 3, \ldots$ and use the stopping criterion that k^* clusters are taken to be optimal where k^* is the smallest value of k for which $k + 1$ clusters are not statistically supported over k clusters.

Permutation Test

The permutation test generates a new data set from the input data set by permuting the columns of all the multiple sequence alignments—the alignments are concatenated, the columns are shuffled, and the concatenated alignment is broken back into individual alignments of the same lengths as the original ones. The effect of this is to uniformly distribute the columns over the data set, removing any between-locus incongruence that might form the basis for clustering. These resampled data are analyzed twice, by partitioning into k and $k+1$ clusters, and we calculate Δ_k. The whole permutation procedure is repeated 100 times to estimate the distribution of Δ_k.

Note that it would be conceptually preferable to permute the columns such that a distribution of loci among exactly k underlying trees is preserved (as per the null hypothesis). However, we have not found a good way to do so. Thus, we implicitly assume that the distribution of the improvement in likelihood from k to $k + 1$ is the same whether the true number of clusters is 1 or k. Our extensive simulations suggest that this approximation works well in practice.

Parametric Bootstrap

As a parametric alternative to the nonparametric permutation test, we use simulation to generate new data sets using parameters estimated during the analysis of the original data. After the analysis, each locus belongs to one of k clusters, and is therefore associated with one of k cluster trees. In the simulated data set, each locus is simulated along its associated cluster tree, using evolutionary model parameters estimated in the analysis. Alignment length and gap positions are duplicated from the initial data (Goldman et al. 1998). Consequently, the data are simulated under the null hypothesis that loci evolved along k underlying trees. The simulated data are clustered and separately analyzed with k clusters and with $k + 1$ clusters to calculate the increase in the partition log likelihood, Δ_k. This "parametric bootstrap" procedure is repeated for 100 data sets to estimate the distribution of Δ_k. The simulation code makes use of the Bio++ libraries (Guéguen et al. 2013).

Simulating Data Sets with Incongruence

The simulated data used in this study were generated to represent evolutionary histories with incongruent phylogenies. Consequently, generating the simulated data involved three stages: 1) deciding on the number of taxa, clusters, loci, and distribution of loci into clusters; 2) for each cluster, generating an evolutionary tree; and 3) for each locus, simulating sequences along its cluster's tree.

Number of Taxa, Clusters, Loci, and Distribution of Loci into Clusters

We produced data sets according to four scenarios with varying numbers of taxa, loci, clusters, and distribution of loci among these clusters, as described in table 3.

Generating Cluster Trees

All cluster trees are derived from an underlying "species tree." For each data set, we simulated a random species tree using a Yule pure speciation model (Yule 1925), implemented in Dendropy (Sukumaran and Holder 2010).

To generate incongruent cluster trees, we started from this species tree and applied sequences of random rearrangements of potential biological relevance. The type of rearrangements was either NNI or Subtree Prune and Regraft (SPR). NNI makes local rearrangements, such as those that might be found as a result of ILS. SPRs were used to make rearrangements involving branches at a greater separation within the tree, consistent with the kind of rearrangements observed in HGT (Galtier 2007).

We applied a predetermined number of rearrangements to the underlying tree for any given data set. This number was varied to control the difficulty of the data set, that is, the expected difficulty for a clustering method to reproduce the correct partition of the data. A data set with a small number of rearrangements is derived from cluster trees that are more similar to each other than one with a large number of rearrangements, and therefore represents a more difficult case. The number of rearrangements we used ranged from 1 to 10; beyond 10 NNIs or SPRs the underlying trees were so different that all clustering strategies performed so well that there was no distinction between them.

Combining the 4 scenarios from the previous section with the 2 rearrangement types and 10 difficulty levels yields 80 different parameterizations that describe the attributes of the data sets we generate.

Simulating Data Sets for Testing Combinations

For each parameterization we generated 1,000 replicate data sets according to the following process:

- Randomly generate an ultrametric species tree according to the Yule process.
- For each cluster, apply a sequence of random tree rearrangements to the species tree to generate the cluster tree. The species tree is reset at the end of the sequence of rearrangements, so that it is identical for each cluster prior to the rearrangements being applied. The rearrangements are either NNI or SPR. The branches at which these operations are applied are selected randomly according to the following procedure: the tree length, L, is the sum of all branch lengths. A line $(0, L)$ can be interpreted as all the branches in the tree laid end-to-end. A random value drawn from $U(0, L)$ gives us both a randomly selected branch—according to the branch segment it falls in—and a position on that branch.
- Draw a set of branch lengths for the cluster trees: inner branch lengths are set to values drawn from Gamma(shape $=$ 0.67, scale $=$ 0.16), terminal branch lengths to values drawn from Gamma(shape $=$ 0.54, scale $=$ 0.48). These distributions were fit to the branch lengths inferred for the yeast data set.
- Simulate alignments from each cluster tree according to the distribution of loci into clusters. Protein sequences were simulated using ALF (Dalquen et al. 2012), using the WAG model of substitution (Whelan and Goldman 2001) with four categories of gamma distributed rates ($\alpha = 1$); (Yang 1994). Sequence lengths were drawn from a gamma distribution with shape $=$ 1.772 and scale $=$ 279.9. These parameters were estimated from the distribution of alignment lengths of the yeast data set (see Yeasts).
- Sequences were removed from the alignments with probability (1 − occupancy).

Empirical Data

Yeasts

After validating the performance of our method under the controlled conditions of simulation, we investigated its performance on a data set of 344 orthologous groups from 18 yeast species (Hess and Goldman 2011). We analyzed protein sequences using the WAG model (Whelan and Goldman 2001). The loci were clustered based on geodesic distances and spectral clustering, with the number of clusters determined by parametric bootstrap.

Chiastocheta

The second data set consisted of the RAD sequences obtained from *Chiastocheta* flies (Diptera: Anthomyiidae) collected across their whole European range. Samples were genotyped using a modified ddRAD protocol (Peterson et al. 2012; Mastretta-Yanes et al. 2015). De novo locus assembly was performed using the pyRAD 2.0 package (Eaton 2014), with read clustering similarity threshold of 75%, both on within- and among-sample level. Other parameters were set as follows: all nucleotides with Phred quality lower than 20 were treated as unknown bases, and reads with more than 4 unknown bases were removed from the data set; possible paralogs were removed by filtering out the loci that had more than five heterozygous positions per locus within individuals, more than 10 heterozygotes per nucleotide position among samples, and the loci for which more than two alleles were present per individual. In total, 273 individuals were sequenced, with 33 technical replicates. For the purpose of this study, only high coverage loci (i.e., present in at least 100 samples) were retained. This resulted in a matrix of 176 loci across 306 samples. Phylogenetic analysis was performed using the GTR model + 4 categories of Gamma-distributed rates across sites. Clustering parameters were geodesic distances, spectral clustering, and the number of clusters was estimated using the nonparametric permutation test stopping criterion.

Data Available for Download

Simulation data and results from "Performance of the Combinations of Distance Metrics and Clustering Methods," "Performance of Methods for Determining the Number of Clusters," and "Dealing with Incomplete Occupancy across Loci," and the alignments and trees for the original loci and for the optimal clusters for the yeast (344 loci; 3 clusters) and *Chiastocheta* (176 loci; 4 clusters) data sets, are available for download from http://www.ebi.ac.uk/goldman-srv/treeCl (last accessed March 1, 2016).

Acknowledgments

K.G. and N.G. acknowledge funding from the European Molecular Biology Laboratory. The work was also supported by Swiss National Science Foundation grants PP00P3_150654 to C.D. and PP00P3_144870 to N.A.

References

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

Akaike H. 1974. A new look at the statistical model identification. *IEEE Trans Automat Control*. 19:716–723.

Ané C, Larget B, Baum DA, Smith SD, Rokas A. 2007. Bayesian estimation of concordance among gene trees. *Mol Biol Evol*. 24:412–426.

Anisimova M, Gil M, Dufayard JF, Dessimoz C, Gascuel O. 2011. Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes. *Syst Biol*. 60:685–699.

Antoniak CE. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann Stat*. 2:1152–1174.

Bansal MS, Burleigh JG, Eulenstein O. 2010. Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics* 11(Suppl 1):S42.

Billera LJ, Holmes SP, Vogtmann K. 2001. Geometry of the space of phylogenetic trees. *Adv Appl Math*. 27:733–767.

Bininda-Emonds ORP, Gittleman JL, Steel MA. 2002. The (super)tree of life: procedures, problems, and prospects. *Annu Rev Ecol Syst*. 33:265–289.

Bordewich M, Semple C. 2005. On the computational complexity of the rooted subtree prune and regraft distance. *Ann Comb*. 8:409–423.

Boussau B, Szöllosi GJ, Duret L, Gouy M, Tannier E, Daubin V. 2013. Genome-scale coestimation of species and gene trees. *Genome Res*. 23:323–330.

Bull JJ, Huelsenbeck JP, Cunningham CW, Swofford DL, Waddell PJ. 1993. Partitioning and combining data in phylogenetic analysis. *Syst Biol*. 42:384–397.

Chen K, Durand D, Farach-Colton M. 2000. NOTUNG: a program for dating gene duplications and optimizing gene family trees. *J Comput Biol*. 7:429–447.
Chung Y, Perna NT, Ané C. 2013. Computing the joint distribution of tree shape and tree distance for gene tree inference and recombination detection. *IEEE/ACM Trans Comput Biol Bioinform*. 10:1263–1274.
Cunningham CW. 1997. Can three incongruence tests predict when data should be combined? *Mol Biol Evol*. 14:733–740.
Dalquen DA, Anisimova M, Gonnet GH, Dessimoz C. 2012. ALF—a simulation framework for genome evolution. *Mol Biol Evol*. 29:1115–1123.
de Hoon MJL, Imoto S, Nolan J, Miyano S. 2004. Open source clustering software. *Bioinformatics* 20:1453–1454.
de Queiroz A, Gatesy J. 2007. The supermatrix approach to systematics. *Trends Ecol Evol*. 22:34–41.
Delsuc F, Brinkmann H, Philippe H. 2005. Phylogenomics and the reconstruction of the tree of life. *Nat Rev Genet*. 6:361–375.
Després L, Pettex E, Plaisance V, Pompanon F. 2002. Speciation in the globeflower fly *Chiastocheta* spp. (Diptera: Anthomyiidae) in relation to host plant species, biogeography, and morphology. *Mol Phylogenet Evol*. 22:258–268.
Dessimoz C, Margadant D, Gonnet GH. 2008. DLIGHT—lateral gene transfer detection using pairwise evolutionary distances in a statistical framework. In: Vingron M, Wong L, editors. Research in computational molecular biology, Lecture Notes in Computer science. Berlin Heidelberg (Germany): Springer. p. 315–330.
Doyon JP, Scornavacca C, Gorbunov KY, Szöllősi GJ, Ranwez V, Berry V. 2010. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In: Tannier E, editor. Comparative genomics. Lecture Notes in Computer Science. Berlin Heidelberg (Germany): Springer. p. 93–108.
Dujon B. 2010. Yeast evolutionary genomics. *Nat Rev Genet*. 11:512–524.
Eaton DAR. 2014. PyRAD: assembly of de novo RADseq loci for phylogenetic analyses. *Bioinformatics* 30:1844–1849.
Enright AJ, Van Dongen S, Ouzounis CA. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*. 30:1575–1584.
Espíndola A, Buerki S, Alvarez N. 2012. Ecological and historical drivers of diversification in the fly genus *Chiastocheta* Pokorny. *Mol Phylogenet Evol*. 63:466–474.
Estabrook GF, McMorris FR, Meacham CA. 1985. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Syst Biol*. 34:193–200.
Ester M, Kriegel HP, Sander J, Xu X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han J, Fayyad U, editors. Second International Conference on Knowledge Discovery and Data Mining. AAAI Press. p. 226–231.
Felsenstein J. 2004. Inferring phylogenies. Sunderland (MA): Sinauer Associates.
Ferguson TS. 1973. A Bayesian analysis of some nonparametric problems. *Ann Stat*. 1:209–230.
Flouri T, Izquierdo-Carrasco F, Darriba D, Aberer AJ, Nguyen LT, Minh BQ, Von Haeseler A, Stamatakis A. 2015. The phylogenetic likelihood library. *Syst Biol*. 64:356–362.
Fowlkes C, Belongie S, Chung F, Malik J. 2004. Spectral grouping using the Nyström method. *IEEE Trans Pattern Anal Mach Intell*. 26:214–225.
Frey BJ, Dueck D. 2007. Clustering by passing messages between data points. *Science* 315:972–976.
Galtier N. 2007. A model of horizontal gene transfer and the bacterial phylogeny problem. *Syst Biol*. 56:633–642.
Goldman N. 1993. Statistical tests of models of DNA substitution. *J Mol Evol*. 36:182–198.
Goldman N, Thorne JL, Jones DT. 1998. Assessing the impact of secondary structure and solvent accessibility on protein evolution. *Genetics* 149:445–458.
Guéguen L, Gaillard S, Boussau B, Gouy M, Groussin M, Rochette NC, Bigot T, Fournier D, Pouyet F, Cahais V, et al. 2013. Bio++: efficient extensible libraries and tools for computational molecular evolution. *Mol Biol Evol*. 30:1745–1750.
Hallett MT, Lagergren J. 2001. Efficient algorithms for lateral gene transfer problems. In: Proceedings of the Fifth Annual International Conference on Computational Biology. RECOMB '01. New York: ACM. p. 149–156.
Heled J, Drummond AJ. 2010. Bayesian inference of species trees from multilocus data. *Mol Biol Evol*. 27:570–580.
Hess J, Goldman N. 2011. Addressing inter-gene heterogeneity in maximum likelihood phylogenomic analysis: yeasts revisited. *PLoS One* 6:e22783.
Huelsenbeck JP, Swofford DL, Cunningham CW, Bull JJ, Waddell PJ. 1994. Is character weighting a panacea for the problem of data heterogeneity in phylogenetic analysis? *Syst Biol*. 43:288–291.
Jeffroy O, Brinkmann H, Delsuc F, Philippe H. 2006. Phylogenomics: the beginning of incongruence? *Trends Genet*. 22:225–231.
Kashyap RL, Subas S. 1974. Statistical estimation of parameters in a phylogenetic tree using a dynamic model of the substitutional process. *J Theor Biol*. 47:75–101.
Kaufman L, Rousseeuw P. 1987. Clustering by means of medoids. Delft (Netherlands): North-Holland.
Kosakovsky Pond SL, Posada D, Gravenor MB, Woelk CH, Frost SDW. 2006. Automated phylogenetic detection of recombination using a genetic algorithm. *Mol Biol Evol*. 23:1891–1901.
Kuhner MK, Felsenstein J. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol Biol Evol*. 11:459–468.
Lanfear R, Calcott B, Ho SYW, Guindon S. 2012. Partitionfinder: combined selection of partitioning schemes and substitution models for phylogenetic analyses. *Mol Biol Evol*. 29:1695–1701.
Larget BR, Kotha SK, Dewey CN, Ané C. 2010. BUCKy: gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics* 26:2910–2911.
Leigh JW, Lapointe FJ, Lopez P, Bapteste E. 2011. Evaluating phylogenetic congruence in the post-genomic era. *Genome Biol Evol*. 3:571–587.
Leigh JW, Schliep K, Lopez P, Bapteste E. 2011. Let them fall where they may: congruence analysis in massive phylogenetically messy data sets. *Mol Biol Evol*. 28:2773–2785.
Lin Y, Rajan V, Moret BME. 2012. A metric for phylogenetic trees based on matching. *IEEE/ACM Trans Comput Biol Bioinform*. 9:1014–1022.
Liu L, Yu L, Kubatko L, Pearl DK, Edwards SV. 2009. Coalescent methods for estimating phylogenetic trees. *Mol Phylogenet Evol*. 53:320–328.
Mastretta-Yanes A, Arrigo N, Alvarez N, Jorgensen TH, Piñero D, Emerson BC. 2015. Restriction site-associated DNA sequencing, genotyping error estimation and de novo assembly optimization for population genetic inference. *Mol Ecol Resour*. 15:28–41.
Meilă M. 2007. Comparing clusterings—an information based distance. *J Multivar Anal*. 98:873–895.
Mirarab S, Bayzid MS, Boussau B, Warnow T. 2014. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science* 346:1250463.
Müllner D. 2013. fastcluster: fast hierarchical, agglomerative clustering routines for R and Python. *J Stat Softw*. 53:1–18.
Newton RR, Newton ILG. 2013. PhyBin: binning trees by topology. *PeerJ* 1:e187
Neyman J. 1971. Molecular studies of evolution: a source of novel statistical problems. In: Gupta SS, Yackel J, editors. Statistical decision theory and related topics. New York: Academic Press. p. 1–27.
Ng AY, Jordan MI, Weiss Y. 2001. On spectral clustering: analysis and an algorithm. In: Dietterich T, Becker S, Ghahramani Z, editors. Advances in neural information processing system. Vol. 2. MIT Press. p. 849–856.
Nye TMW. 2008. Trees of trees: an approach to comparing multiple alternative phylogenies. *Syst Biol*. 57:785–794.
Owen M, Provan JS. 2011. A fast algorithm for computing geodesic distances in tree space. *IEEE/ACM Trans Comput Biol Bioinform*. 8:2–13.
Pamilo P, Nei M. 1988. Relationships between gene trees and species trees. *Mol Biol Evol*. 5:568–583.

Pattengale ND, Gottlieb EJ, Moret BME. 2007. Efficiently computing the Robinson-Foulds metric. *J Comput Biol*. 14:724–735.
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 12:2825–2830.
Pellmyr O. 1992. The phylogeny of a mutualism: evolution and coadaptation between *Trollius* and its seed-parasitic pollinators. *Biol J Linn Soc Lond*. 47:337–365.
Peterson BK, Weber JN, Kay EH, Fisher HS, Hoekstra HE. 2012. Double digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS One* 7:e37135.
Philippe H, Brinkmann H, Lavrov DV, Littlewood DTJ, Manuel M, Wörheide G, Baurain D. 2011. Resolving difficult phylogenetic questions: why more sequences are not enough. *PLoS Biol*. 9:e1000602.
Planet PJ. 2006. Tree disagreement: measuring and testing incongruence in phylogenies. *J Biomed Inform*. 39:86–102.
Rannala B, Yang Z. 2003. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics* 164:1645–1656.
Robinson DF, Foulds LR. 1981. Comparison of phylogenetic trees. *Math Biosci*. 53:131–147.
Rousseeuw PJ. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math*. 20:53–65.
Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*. 4:406–425.
Schwarz G. 1978. Estimating the dimension of a model. *Ann Stat*. 6:461–464.
Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30:1312–1313.
Suchan T, Beauverd M, Trim N, Alvarez N. 2015. Asymmetrical nature of the *Trollius–Chiastocheta* interaction: insights into the evolution of nursery pollination systems. *Ecol Evol*. 5:4766–4777.
Sukumaran J, Holder MT. 2010. DendroPy: a Python library for phylogenetic computing. *Bioinformatics* 26:1569–1571.
Szöllősi GJ, Daubin V. 2012. Modeling gene family evolution and reconciling phylogenetic discord. *Methods Mol Biol*. 856:29–51.
Tavaré S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. In: Miura RM, editor. Lectures on mathematics in the life sciences. Vol. 17. Providence (RI): American Mathematic Society. p. 57–86.
Torgerson WS. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* 17:401–419.
Waddell PJ, Kishino H, Ota R. 2000. Rapid evaluation of the phylogenetic congruence of sequence data using likelihood ratio tests. *Mol Biol Evol*. 17:1988–1992.
Warnow TJ. 1994. Tree compatibility and inferring evolutionary history. *J Algorithm Comput Technol*. 16:388–407.
Weisrock DW, Smith SD, Chan LM, Biebouw K, Kappeler PM, Yoder AD. 2012. Concatenation and concordance in the reconstruction of mouse lemur phylogeny: an empirical demonstration of the effect of allele sampling in phylogenetics. *Mol Biol Evol*. 29:1615–1630.
Weyenberg G, Huggins PM, Schardl CL, Howe DK, Yoshida R. 2014. kdetrees: non-parametric estimation of phylogenetic tree distributions. *Bioinformatics* 30:2280–2287.
Whelan S, Goldman N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol Biol Evol*. 18:691–699.
Whidden C, Beiko RG, Zeh N. 2013. Fixed-parameter algorithms for maximum agreement forests. *SIAM J Comput*. 42:1431–1466.
Wilks SS. 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Ann Math Stat*. 9:60–62.
Yang Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J Mol Evol*. 39:306–314.
Yule GU. 1925. A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FRS. *Philos Trans R Soc Lond B* 213:21–87.
Zelnik-Manor L., Perona P. 2004. Self-tuning spectral clustering. In: Saul LK, Weiss Y, Bottou L, editor. Advances in Neural Information Processing Systems 17 (NIPS 2004). Vancouver, Canada. p. 1601–1608.

BIBLIOGRAPHY

Abby, S. S., E. Tannier, M. Gouy, and V. Daubin. 2010. Detecting lateral gene transfers by statistical reconciliation of phylogenetic forests. *BMC Bioinformatics* 11:1–13.

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, September 2-8, 1971*, edited by B. N. Petrov and F. Csáki, pp. 267–281. Budapest: Akadémiai Kiadó.

Altenhoff, A. M., B. Boeckmann, S. Capella-Gutierrez, D. A. Dalquen, T. DeLuca, K. Forslund, J. Huerta-Cepas, B. Linard, C. Pereira, L. P. Pryszcz, F. Schreiber, A. S. da Silva, et al. 2016. Standardized benchmarking in the quest for orthologs. *Nat. Methods.*

Altenhoff, A. M., M. Gil, G. H. Gonnet, and C. Dessimoz. 2013. Inferring hierarchical orthologous groups from orthologous gene pairs. *PLoS One* 8:e53786.

Altenhoff, A. M., A. Schneider, G. H. Gonnet, and C. Dessimoz. 2011. OMA 2011: orthology inference among 1000 complete genomes. *Nucleic Acids Res.* 39:D289–D294.

Altenhoff, A. M., N. Škunca, N. Glover, C.-M. Train, A. Sueki, I. Piližota, K. Gori, B. Tomiczek, S. Müller, H. Redestig, G. H. Gonnet, and C. Dessimoz. 2015. The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements. *Nucleic Acids Res.* 43:D240–D249.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215:403–410.

Ané, C., B. Larget, D. A. Baum, S. D. Smith, and A. Rokas. 2007. Bayesian estimation of concordance among gene trees. *Mol. Biol. Evol.* 24:412–426.

Anisimova, M., and O. Gascuel. 2006. Approximate likelihood-ratio test for branches: A fast, accurate, and powerful alternative. *Syst. Biol.* 55:539–552.

Anisimova, M., M. Gil, J.-F. Dufayard, C. Dessimoz, and O. Gascuel. 2011. Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes. *Syst. Biol.* 60:685–699.

Antoniak, C. E. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Stat.* 2:1152–1174.

Arnold, B., R. B. Corbett-Detig, D. Hartl, and K. Bomblies. 2013. RADseq underestimates diversity and introduces genealogical biases due to nonrandom haplotype sampling. *Mol. Ecol.* 22:3179–3190.

Baird, N. A., P. D. Etter, T. S. Atwood, M. C. Currey, A. L. Shiver, Z. A. Lewis, E. U. Selker, W. A. Cresko, and E. A. Johnson. 2008. Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PLoS One* 3:e3376.

Bansal, M. S., J. G. Burleigh, and O. Eulenstein. 2010. Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics* 11:1–9.

Bao, L., H. Gu, K. A. Dunn, and J. P. Bielawski. 2008. Likelihood-based clustering (LiBaC) for codon models, a method for grouping sites according to similarities in the underlying process of evolution. *Mol. Biol. Evol.* 25:1995–2007.

Beiko, R. G., T. J. Harlow, and M. A. Ragan. 2005. Highways of gene sharing in prokaryotes. *Proc. Natl. Acad. Sci. U. S. A.* 102:14332–14337.

Benson, D. A., I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. 2005. GenBank. *Nucleic Acids Res.* 33:D34–D38.

Billera, L. J., S. P. Holmes, and K. Vogtmann. 2001. Geometry of the space of phylogenetic trees. *Adv. Appl. Math.* 27:733–767.

Bininda-Emonds, O. R. P., J. L. Gittleman, and M. A. Steel. 2002. The (super) tree of life: procedures, problems, and prospects. *Annu. Rev. Ecol. Syst.* 33:265–289.

Boussau, B., G. J. Szöllosi, L. Duret, M. Gouy, E. Tannier, and V. Daubin. 2013. Genome-scale coestimation of species and gene trees. *Genome Res.* 23:323–330.

Brown, J. M., and A. R. Lemmon. 2007. The importance of data partitioning and the utility of Bayes factors in Bayesian phylogenetics [in en]. *Syst. Biol.* 56:643–655.

Bull, J. J., J. P. Huelsenbeck, C. W. Cunningham, D. L. Swofford, and P. J. Waddell. 1993. Partitioning and combining data in phylogenetic analysis. *Syst. Biol.* 42:384–397.

Byrne, K. P., and K. H. Wolfe. 2005. The Yeast Gene Order Browser: combining curated homology and syntenic context reveals gene fate in polyploid species. *Genome Res.* 15:1456–1461.

Caliński, T., and J. Harabasz. 1974. A dendrite method for cluster analysis. *Commun. Stat. Simul. Comput.* 3:1–27.

Cavalli-Sforza, L., and A. W. F. Edwards. 1967. Phylogenetic analysis. Models and estimation procedures. *Am. J. Hum. Genet.* 19:233–257.

Celeux, G., and G. Govaert. 1992. A classification EM algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal.* 14:315–332.

Chattopadhyay, B., K. M. Garg, and U. Ramakrishnan. 2014. Effect of diversity and missing data on genetic assignment with RAD-Seq markers. *BMC Res. Notes* 7:1–3.

Chen, K., D. Durand, and M. Farach-Colton. 2000. NOTUNG: a program for dating gene duplications and optimizing gene family trees. *J. Comput. Biol.* 7:429–447.

Chor, B., M. D. Hendy, B. R. Holland, and D. Penny. 2000. Multiple maxima of likelihood in phylogenetic trees: an analytic approach. *Mol. Biol. Evol.* 17:1529–1541.

Chung, Y., N. T. Perna, and C. Ané. 2013. Computing the joint distribution of tree shape and tree distance for gene tree inference and recombination detection. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 10:1263–1274.

Cox, D. R. 1961. Tests of separate families of hypotheses. *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability* 1:105–123.

Cox, D. R. 1962. Further Results on Tests of Separate Families of Hypotheses. *J. R. Stat. Soc. Series B Stat. Methodol.* 24:406–424.

Crisp, A., C. Boschetti, M. Perry, A. Tunnacliffe, and G. Micklem. 2015. Expression of multiple horizontally acquired genes is a hallmark of both vertebrate and invertebrate genomes. *Genome Biol.* 16:50.

Crosby, R. W., and T. L. Williams. 2012. A fast algorithm for computing the quartet distance for large sets of evolutionary trees. In *Bioinformatics Research and Applications,* edited by L. Bleris, I. Măndoiu, R. Schwartz, and J. Wang, pp. 60–71. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Cunningham, C. W. 1997. Can three incongruence tests predict when data should be combined? *Mol. Biol. Evol.* 14:733–740.

Darriba, D., and D. Posada. 2015. "The impact of partitioning on phylogenomic accuracy."

Darwin, C. 1859. On the origin of species by means of natural selection, or. *The Preservation of Favoured Races in the Struggle for Life, London/Die Entstehung der Arten durch natürliche Zuchtwahl, Leipzig oJ.*

Dasgupta, B. 2006. *Applied mathematical methods.* Pearson Education India.

Davey, J. W., P. A. Hohenlohe, P. D. Etter, J. Q. Boone, J. M. Catchen, and M. L. Blaxter. 2011. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nat. Rev. Genet.* 12:499–510.

David, L. A., and E. J. Alm. 2011. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature* 469:93–96.

Day, W. H. E. 1985. Optimal algorithms for comparing trees with labeled leaves. *J. Classification* 2:7–28.

Delsuc, F., H. Brinkmann, and H. Philippe. 2005. Phylogenomics and the reconstruction of the tree of life. *Nat. Rev. Genet.* 6:361–375.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B Stat. Methodol.* 39:1–38.

Després, L., E. Pettex, V. Plaisance, and F. Pompanon. 2002. Speciation in the globeflower fly *Chiastocheta* spp. (*Diptera*: Anthomyiidae) in relation to host plant species, biogeography, and morphology. *Mol. Phylogenet. Evol.* 22:258–268.

Dessimoz, C., B. Boeckmann, A. C. J. Roth, and G. H. Gonnet. 2006. Detecting non-orthology in the COGs database and other approaches grouping orthologs using genome-specific best hits. *Nucleic Acids Res.* 34:3309–3316.

Dessimoz, C., D. Margadant, and G. H. Gonnet. 2008. DLIGHT—Lateral Gene Transfer detection using pairwise evolutionary distances in a statistical framework. In *Research in Computational Molecular Biology*, edited by M. Vingron and L. Wong, pp. 315–330. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Do, C. B., and S. Batzoglou. 2008. What is the expectation maximization algorithm? *Nat. Biotechnol.* 26:897–899.

Domazet-Lošo, T., J. Brajković, and D. Tautz. 2007. A phylostratigraphy approach to uncover the genomic history of major adaptations in metazoan lineages. *Trends Genet.* 23:533–539.

Doyon, J.-P., C. Scornavacca, K. Yu. Gorbunov, G. J. Szöllősi, V. Ranwez, and V. Berry. 2010. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In *Comparative Genomics*, pp. 93–108. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Dujon, B. 2010. Yeast evolutionary genomics. *Nat. Rev. Genet.* 11:512–524.

Eaton, D. A. R. 2014. PyRAD: assembly of de novo RADseq loci for phylogenetic analyses. *Bioinformatics* 30:1844–1849.

Edgar, R. C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* 32:1792–1797.

Edwards, A. W. 1992. Likelihood, expanded ed. *Baltimore: Johns Hopkins.*

Efron, B. 1981. Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika* 68:589–599.

Efron, B. 1985. Bootstrap confidence intervals for a class of parametric problems. *Biometrika* 72:45–58.

Espíndola, A., S. Buerki, and N. Alvarez. 2012. Ecological and historical drivers of diversification in the fly genus *Chiastocheta Pokorny*. *Mol. Phylogenet. Evol.* 63:466–474.

Estabrook, G. F., F. R. McMorris, and C. A. Meacham. 1985. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Syst. Biol.* 34:193–200.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach [in English]. *J. Mol. Evol.* 17:368–376.

Felsenstein, J. 1973. Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Syst. Zool.* 22:240–249.

Felsenstein, J. 1985. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* 39:783–791.

Felsenstein, J. 1988. Phylogenies from molecular sequences: inference and reliability. *Annu. Rev. Genet.* 22:521–565.

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

Ferguson, T. S. 1973. A Bayesian analysis of some nonparametric problems. *Ann. Stat.* 1:209–230.

Fitch, W. M. 1970. Distinguishing homologous from analogous proteins [in en]. *Syst. Zool.* 19:99–113.

Fitch, W. M., and E. Margoliash. 1967. Construction of phylogenetic trees. *Science* 155:279–284.

Flicek, P., I. Ahmed, M. R. Amode, D. Barrell, K. Beal, S. Brent, D. Carvalho-Silva, P. Clapham, G. Coates, S. Fairley, S. Fitzgerald, L. Gil, et al. 2013. Ensembl 2013. *Nucleic Acids Res.* 41:D48–D55.

Flicek, P., M. R. Amode, D. Barrell, K. Beal, K. Billis, S. Brent, D. Carvalho-Silva, P. Clapham, G. Coates, S. Fitzgerald, L. Gil, C. G. Girón, et al. 2014. Ensembl 2014. *Nucleic Acids Res.* 42:D749–D755.

Flouri, T., F. Izquierdo-Carrasco, D. Darriba, A. J. Aberer, L.-T. Nguyen, B. Q. Minh, A. Von Haeseler, and A. Stamatakis. 2015. The phylogenetic likelihood library. *Syst. Biol.* 64:356–362.

Force, A., M. Lynch, F. B. Pickett, A. Amores, Y. L. Yan, and J. Postlethwait. 1999. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics* 151:1531–1545.

Fujita, A., D. Y. Takahashi, and A. G. Patriota. 2014. A non-parametric method to estimate the number of clusters. *Comput. Stat. Data Anal.* 73:27–39.

Gascuel, O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.* 14:685–695.

Gautier, M., K. Gharbi, T. Cezard, J. Foucaud, C. Kerdelhué, P. Pudlo, J.-M. Cornuet, and A. Estoup. 2013. The effect of RAD allele dropout on the estimation of genetic variation within and between populations. *Mol. Ecol.* 22:3165–3178.

Georgi, B., I. G. Costa, and A. Schliep. 2010. PyMix—the python mixture package—a tool for clustering of heterogeneous biological data. *BMC Bioinformatics* 11:9.

Goldman, N., J. L. Thorne, and D. T. Jones. 1998. Assessing the impact of secondary structure and solvent accessibility on protein evolution. *Genetics* 149:445–458.

Goldman, N., and Z. Yang. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol. Biol. Evol.* 11:725–736.

Goldman, N. 1993. Statistical tests of models of DNA substitution. *J. Mol. Evol.* 36:182–198.

Gonnet, G. H., M. A. Cohen, and S. A. Benner. 1992. Exhaustive matching of the entire protein sequence database. *Science* 256:1443–1445.

Guéguen, L., S. Gaillard, B. Boussau, M. Gouy, M. Groussin, N. C. Rochette, T. Bigot, D. Fournier, F. Pouyet, V. Cahais, A. Bernard, C. Scornavacca, et al. 2013. Bio++: efficient extensible libraries and tools for computational molecular evolution. *Mol. Biol. Evol.* 30:1745–1750.

Guindon, S., J.-F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.* 59:307–321.

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

Hallett, M. T., and J. Lagergren. 2001. Efficient Algorithms for Lateral Gene Transfer Problems. In *Proceedings of the Fifth Annual International Conference on Computational Biology*, pp. 149–156. RECOMB '01. Montreal, Quebec, Canada: ACM.

Hasegawa, M., H. Kishino, and T. Yano. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*

Heijden, R. T. J. M. van der, B. Snel, V. van Noort, and M. A. Huynen. 2007. Orthology prediction at scalable resolution by phylogenetic tree analysis. *BMC Bioinformatics* 8:83.

Heled, J., and A. J. Drummond. 2010. Bayesian inference of species trees from multilocus data. *Mol. Biol. Evol.* 27:570–580.

Hess, J. 2011. "Evolution of transcription factor repertoires in the Saccharomycotina." PhD diss., University of Cambridge.

Hess, J., and N. Goldman. 2011. Addressing inter-gene heterogeneity in maximum likelihood phylogenomic analysis: yeasts revisited. *PLoS One* 6:e22783.

Hillis, D. M., C. Moritz, B. K. Mable, and R. G. Olmstead. 1996. *Molecular systematics.* Vol. 23. Sinauer Associates Sunderland, MA.

Hobolth, A., J. Y. Dutheil, J. Hawks, M. H. Schierup, and T. Mailund. 2011. Incomplete lineage sorting patterns among human, chimpanzee, and orangutan suggest recent orangutan speciation and widespread selection. *Genome Res.* 21:349–356.

Hogeweg, P., and B. Hesper. 1984. The alignment of sets of sequences and the construction of phyletic trees: an integrated method [in English]. *J. Mol. Evol.* 20:175–186.

Hordijk, W., and O. Gascuel. 2005. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. *Bioinformatics* 21:4338–4347.

Huelsenbeck, J. P., D. L. Swofford, C. W. Cunningham, J. J. Bull, and P. J. Waddell. 1994. Is character weighting a panacea for the problem of data heterogeneity in phylogenetic analysis? *Syst. Biol.* 43:288–291.

Iantorno, S., K. Gori, N. Goldman, M. Gil, and C. Dessimoz. 2014. Who watches the watchmen? An appraisal of benchmarks for multiple sequence alignment. In *Multiple Sequence Alignment Methods,* edited by D. J. Russell, pp. 59–73. Methods in Molecular Biology. Totowa, NJ: Humana Press.

Jacques, N., A. Zenouche, N. Gunde-Cimerman, and S. Casaregola. 2015. Increased diversity in the genus *Debaryomyces* from Arctic glacier samples. *Antonie Van Leeuwenhoek* 107:487–501.

Jarvis, E. D., S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. W. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard, A. Suh, C. C. Weber, et al. 2014. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science* 346:1320–1331.

Jeffroy, O., H. Brinkmann, F. Delsuc, and H. Philippe. 2006. Phylogenomics: the beginning of incongruence? *Trends Genet.* 22:225–231.

Jukes, T. H., and C. R. Cantor. 1969. Evolution of protein molecules. *Mammalian protein metabolism* 3:132.

Kashyap, R. L., and S. Subas. 1974. Statistical estimation of parameters in a phylogenetic tree using a dynamic model of the substitutional process. *J. Theor. Biol.* 47:75–101.

Katoh, K., K.-I. Kuma, H. Toh, and T. Miyata. 2005. MAFFT version 5: improvement in accuracy of multiple sequence alignment [in English]. *Nucleic Acids Res.* 33:511–518.

Kaufman, L., and P. J. Rousseeuw. 1987. Clustering by means of medoids. In *Reports of the Faculty of Mathematics and Informatics.* Delft, Netherlands: Delft University of Technology.

Kelley, L. A., S. P. Gardner, and M. J. Sutcliffe. 1996. An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally related subfamilies. *Protein Eng.* 9:1063–1065.

Kendall, M., and C. Colijn. 2015. A tree metric using structure and length to capture distinct phylogenetic signals. arXiv: 1507.05211.

Kimura, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*

Köhler, F., and G. Deein. 2010. Hybridisation as potential source of incongruence in the morphological and mitochondrial diversity of a Thai freshwater gastropod (Pachychilidae, *Brotia* H. Adams, 1866). *Zoosyst. Evol.* 86:301–314.

Kosakovsky Pond, S. L., D. Posada, M. B. Gravenor, C. H. Woelk, and S. D. W. Frost. 2006. Automated phylogenetic detection of recombination using a genetic algorithm. *Mol. Biol. Evol.* 23:1891–1901.

Kruskal, J. B. 1964. Nonmetric multidimensional scaling: a numerical method. *Psychometrika* 29:115–129.

Kubatko, L. S. 2009. Identifying hybridization events in the presence of coalescence via model selection. *Syst. Biol.* 58:478–488.

Kuhner, M. K., and J. Felsenstein. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.* 11:459–468.

Kuzniar, A., R. C. H. J. van Ham, S. Pongor, and J. A. M. Leunissen. 2008. The quest for orthologs: finding the corresponding gene across genomes [in en]. *Trends Genet.* 24:539–551.

Lakner, C., P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist. 2008. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Syst. Biol.* 57:86–103.

Larget, B. R., S. K. Kotha, C. N. Dewey, and C. Ané. 2010. BUCKy: gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics* 26:2910–2911.

Lartillot, N., N. Rodrigue, D. Stubbs, and J. Richer. 2013. PhyloBayes MPI: phylogenetic reconstruction with infinite mixtures of profiles in a parallel environment. *Syst. Biol.* 62:611–615.

Le, S. Q., and O. Gascuel. 2008. An improved general amino acid replacement matrix. *Mol. Biol. Evol.* 25:1307–1320.

Leigh, J. W., F.-J. Lapointe, P. Lopez, and E. Bapteste. 2011. Evaluating phylogenetic congruence in the post-genomic era. *Genome Biol. Evol.* 3:571–587.

Leigh, J. W., K. Schliep, P. Lopez, and E. Bapteste. 2011. Let them fall where they may: congruence analysis in massive phylogenetically messy data sets. *Mol. Biol. Evol.* 28:2773–2785.

Li, L., C. J. Stoeckert Jr, and D. S. Roos. 2003. OrthoMCL: identification of ortholog groups for eukaryotic genomes [in English]. *Genome Res.* 13:2178–2189.

Li, M., and L. Zhang. 1999. Twist-rotation transformations of binary trees and arithmetic expressions. *J. Algorithm. Comput. Technol.* 32:155–166.

Lin, Y., V. Rajan, and B. M. E. Moret. 2012. A metric for phylogenetic trees based on matching. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9:1014–1022.

Liu, L., L. Yu, L. Kubatko, D. K. Pearl, and S. V. Edwards. 2009. Coalescent methods for estimating phylogenetic trees. *Mol. Phylogenet. Evol.* 53:320–328.

Luxburg, U. von. 2007. A tutorial on spectral clustering. *Stat. Comput.* 17:395–416.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics.* The Regents of the University of California.

Marcet-Houben, M., and T. Gabaldón. 2015. Beyond the whole-genome duplication: phylogenetic evidence for an ancient interspecies hybridization in the baker's yeast lineage. *PLoS Biol.* 13:e1002220.

Mastretta-Yanes, A., N. Arrigo, N. Alvarez, T. H. Jorgensen, D. Piñero, and B. C. Emerson. 2015. Restriction site-associated DNA sequencing, genotyping error estimation and *de novo* assembly optimization for population genetic inference. *Mol. Ecol. Resour.* 15:28–41.

Meilă, M. 2007. Comparing clusterings—an information based distance. *J. Multivar. Anal.* 98:873–895.

Minh, B. Q., M. A. T. Nguyen, and A. von Haeseler. 2013. Ultrafast approximation for phylogenetic bootstrap. *Mol. Biol. Evol.* 30:1188–1195.

Mirarab, S., M. S. Bayzid, B. Boussau, and T. Warnow. 2014. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science* 346:1250463.

Murata, M., J. S. Richardson, and J. L. Sussman. 1985. Simultaneous comparison of three protein sequences [in en]. *Proc. Natl. Acad. Sci. U. S. A.* 82:3073–3077.

NCBI Resource Coordinators. 2015. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research* 43:D6–D17.

Needleman, S. B., and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins [in en]. *J. Mol. Biol.* 48:443–453.

Nei, M., and S. Kumar. 2000. *Molecular evolution and phylogenetics.* Oxford university press.

Newton, R. R., and I. L. G. Newton. 2013. PhyBin: binning trees by topology. *PeerJ* 1:e187.

Neyman, J. 1971. Molecular studies of evolution: a source of novel statistical problems. In *Statistical Decision Theory and Related Topics,* edited by S. S. Gupta and J. Yackel, pp. 1–27. New York: Academic Press.

Ng, A. Y., M. I. Jordan, and Y. Weiss. 2002. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS),* edited by T. Dietterich, S. Becker, and Z. Ghahramani, pp. 849–856. Vancouver, Canada.

Nguyen, T. H., J.-P. Doyon, S. Pointet, A.-M. A. Chifolleau, V. Ranwez, and V. Berry. 2012. Accounting for gene tree uncertainties improves gene trees and reconciliation inference. In *Algorithms in Bioinformatics,* pp. 123–134. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Nocedal, J., and S. Wright. 2006. *Numerical optimization.* Edited by T. Mikosch, S. Robinson, and S. Resnick. Springer Science & Business Media.

Nosenko, T., F. Schreiber, M. Adamska, M. Adamski, M. Eitel, J. Hammel, M. Maldonado, W. E. G. Müller, M. Nickel, B. Schierwater, J. Vacelet, M. Wiens, et al. 2013. Deep metazoan phylogeny: when different genes tell different stories. *Mol. Phylogenet. Evol.* 67:223–233.

Nye, T. M. W. 2008. Trees of trees: an approach to comparing multiple alternative phylogenies. *Syst. Biol.* 57:785–794.

Ostlund, G., T. Schmitt, K. Forslund, T. Köstler, D. N. Messina, S. Roopra, O. Frings, and E. L. L. Sonnhammer. 2010. InParanoid 7: new algorithms and tools for eukaryotic orthology analysis [in English]. *Nucleic Acids Res.* 38:D196–203.

Owen, M., and J. S. Provan. 2011. A fast algorithm for computing geodesic distances in tree space. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8:2–13.

Paganini, J., A. Campan-Fournier, M. Da Rocha, P. Gouret, P. Pontarotti, E. Wajnberg, P. Abad, and E. G. J. Danchin. 2012. Contribution of lateral gene transfers to the genome composition and parasitic ability of root-knot nematodes. *PLoS One* 7:e50875.

Pamilo, P., and M. Nei. 1988. Relationships between gene trees and species trees. *Mol. Biol. Evol.* 5:568–583.

Park, H.-S., and C.-H. Jun. 2009. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* 36:3336–3341.

Pattengale, N. D., M. Alipour, O. R. P. Bininda-Emonds, B. M. E. Moret, and A. Stamatakis. 2010. How many bootstrap replicates are necessary? *J. Comput. Biol.* 17:337–354.

Pellmyr, O. 1992. The phylogeny of a mutualism: evolution and coadaptation between *Trollius* and its seed-parasitic pollinators. *Biol. J. Linn. Soc. Lond.* 47:337–365.

Peng, X., J. Alföldi, K. Gori, A. J. Eisfeld, S. R. Tyler, J. Tisoncik-Go, D. Brawand, G. L. Law, N. Skunca, M. Hatta, D. J. Gasper, S. M. Kelly, et al. 2014. The draft genome sequence of the ferret (*Mustela putorius furo*) facilitates study of human respiratory disease. *Nat. Biotechnol.* 32:1250–1255.

Penny, D., E. E. Watson, and M. A. Steel. 1993. Trees from languages and genes are very similar. *Syst. Biol.* 42:382–384.

Peterson, B. K., J. N. Weber, E. H. Kay, H. S. Fisher, and H. E. Hoekstra. 2012. Double digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS One* 7:e37135.

Philippe, H., H. Brinkmann, D. V. Lavrov, D. T. J. Littlewood, M. Manuel, G. Wörheide, and D. Baurain. 2011. Resolving difficult phylogenetic questions: why more sequences are not enough. *PLoS Biol.* 9:e1000602.

Planet, P. J. 2006. Tree disagreement: measuring and testing incongruence in phylogenies. *J. Biomed. Inform.* 39:86–102.

Puigbò, P., S. Garcia-Vallvé, and J. O. McInerney. 2007. TOPD/FMTS: a new software to compare phylogenetic trees. *Bioinformatics* 23:1556–1558.

Puigbò, P., Y. I. Wolf, and E. V. Koonin. 2012. Genome-wide comparative analysis of phylogenetic trees: the prokaryotic forest of life. *Methods Mol. Biol.* 856:53–79.

Queiroz, A. de, and J. Gatesy. 2007. The supermatrix approach to systematics. *Trends Ecol. Evol.* 22:34–41.

Rannala, B., and Z. Yang. 2003. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics* 164:1645–1656.

Ravenhall, M., N. Škunca, F. Lassalle, and C. Dessimoz. 2015. Inferring horizontal gene transfer. *PLoS Comput. Biol.* 11:e1004095.

Reis, M. dos, P. C. J. Donoghue, and Z. Yang. 2016. Bayesian molecular clock dating of species divergences in the genomics era [in en]. *Nat. Rev. Genet.* 17:71–80.

Rice, D. W., A. J. Alverson, A. O. Richardson, G. J. Young, M. V. Sanchez-Puerta, J. Munzinger, K. Barry, J. L. Boore, Y. Zhang, C. W. dePamphilis, E. B. Knox, and J. D. Palmer. 2013. Horizontal transfer of entire genomes via mitochondrial fusion in the angiosperm *Amborella*. *Science* 342:1468–1473.

Robinson, D. F., and L. R. Foulds. 1979. Comparison of weighted labelled trees. In *Combinatorial Mathematics VI*, edited by A. F. Horadam and W. D. Wallis, pp. 119–126. Lecture Notes in Mathematics. Springer Berlin Heidelberg.

Robinson, D. F., and L. R. Foulds. 1981. Comparison of phylogenetic trees. *Math. Biosci.* 53:131–147.

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

Ronquist, F., M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Höhna, B. Larget, L. Liu, M. A. Suchard, and J. P. Huelsenbeck. 2012. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Syst. Biol.* 61:539–542.

Rosenfeld, J. A., A. Payne, and R. DeSalle. 2012. Random roots and lineage sorting. *Mol. Phylogenet. Evol.* 64:12–20.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20:53–65.

Saitou, N., and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4:406–425.

Salichos, L., and A. Rokas. 2013. Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature* 497:327–331.

Sang, T., and Y. Zhong. 2000. Testing hybridization hypotheses based on incongruent gene trees. *Syst. Biol.* 49:422–434.

Scally, A., J. Y. Dutheil, L. W. Hillier, G. E. Jordan, I. Goodhead, J. Herrero, A. Hobolth, T. Lappalainen, T. Mailund, T. Marques-Bonet, S. McCarthy, S. H. Montgomery, et al. 2012. Insights into hominid evolution from the gorilla genome sequence. *Nature* 483:169–175.

Scannell, D. R., K. P. Byrne, J. L. Gordon, S. Wong, and K. H. Wolfe. 2006. Multiple rounds of speciation associated with reciprocal gene loss in polyploid yeasts. *Nature* 440:341–345.

Schwarz, G. 1978. Estimating the dimension of a model. *Ann. Stat.* 6:461–464.

Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13:2498–2504.

Smith, T. F., and M. S. Waterman. 1981. Identification of common molecular subsequences [in en]. *J. Mol. Biol.* 147:195–197.

Sonnhammer, E. L. L., T. Gabaldón, A. W. Sousa da Silva, M. Martin, M. Robinson-Rechavi, B. Boeckmann, P. D. Thomas, C. Dessimoz, and Quest for Orthologs consortium. 2014. Big data and other challenges in the quest for orthologs [in en]. *Bioinformatics* 30:2993–2998.

Stamatakis, A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30:1312–1313.

Stamatakis, A., P. Hoover, and J. Rougemont. 2008. A rapid bootstrap algorithm for the RAxML Web servers. *Syst. Biol.* 57:758–771.

Stegemann, S., M. Keuthe, S. Greiner, and R. Bock. 2012. Horizontal transfer of chloroplast genomes between plant species. *Proc. Natl. Acad. Sci. U. S. A.* 109:2434–2438.

Suchan, T., M. Beauverd, N. Trim, and N. Alvarez. 2015. Asymmetrical nature of the *Trollius–Chiastocheta* interaction: insights into the evolution of nursery pollination systems. *Ecol. Evol.* 5:4766–4777.

Sugar, C. A., and G. M. James. 2011. Finding the number of clusters in a dataset. *J. Am. Stat. Assoc.* 98:750–763.

Sukumaran, J., and M. T. Holder. 2010. DendroPy: a Python library for phylogenetic computing. *Bioinformatics* 26:1569–1571.

Szöllősi, G. J., and V. Daubin. 2012. Modeling gene family evolution and reconciling phylogenetic discord. In *Evolutionary Genomics,* edited by M. Anisimova, pp. 29–51. Methods in Molecular Biology. Humana Press.

Tavaré, S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences:*57–86.

Thompson, J. D., D. G. Higgins, and T. J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice [in English]. *Nucleic Acids Res.* 22:4673–4680.

Tibshirani, R., G. Walther, and T. Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Series B* 63:411–423.

Torgerson, W. S. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* 17:401–419.

Vilella, A. J., J. Severin, A. Ureta-Vidal, L. Heng, R. Durbin, and E. Birney. 2009. EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.* 19:327–335.

Waddell, P. J., H. Kishino, and R. Ota. 2000. Rapid evaluation of the phylogenetic congruence of sequence data using likelihood ratio tests. *Mol. Biol. Evol.* 17:1988–1992.

Wapinski, I., A. Pfeffer, N. Friedman, and A. Regev. 2007. Automatic genome-wide reconstruction of phylogenetic gene trees. *Bioinformatics* 23:i549–i558.

Ward, J. H., Jr. 1963. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* 58:236–244.

Warnow, T. J. 1994. Tree compatibility and inferring evolutionary history. *J. Algorithm. Comput. Technol.* 16:388–407.

Weisrock, D. W., S. D. Smith, L. M. Chan, K. Biebouw, P. M. Kappeler, and A. D. Yoder. 2012. Concatenation and concordance in the reconstruction of mouse lemur phylogeny: an empirical demonstration of the effect of allele sampling in phylogenetics. *Mol. Biol. Evol.* 29:1615–1630.

Weyenberg, G., P. M. Huggins, C. L. Schardl, D. K. Howe, and R. Yoshida. 2014. kdetrees: non-parametric estimation of phylogenetic tree distributions. *Bioinformatics* 30:2280–2287.

Whelan, S. 2007. New approaches to phylogenetic tree search and their application to large numbers of protein alignments. *Syst. Biol.* 56:727–740.

Whelan, S., and N. Goldman. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol.* 18:691–699.

Whidden, C., R. G. Beiko, and N. Zeh. 2013. Fixed-parameter algorithms for maximum agreement forests. *SIAM J. Comput.* 42:1431–1466.

Whidden, C., and F. A. Matsen. 2015. Quantifying MCMC exploration of phylogenetic tree space. *Syst. Biol.* 64:472–491.

Whidden, C., N. Zeh, and R. G. Beiko. 2014. Supertrees based on the subtree prune-and-regraft distance. *Syst. Biol.* 63:566–581.

Wilks, S. S. 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Ann. Math. Stat.* 9:60–62.

Wolfe, K. H., and D. C. Shields. 1997. Molecular evidence for an ancient duplication of the entire yeast genome. *Nature* 387:708–713.

Wolfe, K. H. 2015. Origin of the yeast whole-genome duplication. *PLoS Biol.* 13:e1002221.

Wu, B., A. Buljic, and W. Hao. 2015. Extensive horizontal transfer and homologous recombination generate highly chimeric mitochondrial genomes in yeast. *Mol. Biol. Evol.* 32:2559–2570.

Wu, C. I., and W. H. Li. 1985. Evidence for higher rates of nucleotide substitution in rodents than in man [in en]. *Proc. Natl. Acad. Sci. U. S. A.* 82:1741–1745.

Wu, Y.-C., M. D. Rasmussen, M. S. Bansal, and M. Kellis. 2013. TreeFix: statistically informed gene tree error correction using species trees. *Syst. Biol.* 62:110–120.

Xi, Z., Y. Wang, R. K. Bradley, M. Sugumaran, C. J. Marx, J. S. Rest, and C. C. Davis. 2013. Massive mitochondrial gene transfer in a parasitic flowering plant clade. *PLoS Genet.* 9:e1003265.

Yang, Z. 2000. Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A. *J. Mol. Evol.* 51:423–432.

Yang, Z. 2014. *Molecular evolution: a statistical approach.* Oxford University Press.

Yang, Z., N. Goldman, and A. Friday. 1994. Comparison of models for nucleotide substitution used in maximum-likelihood phylogenetic estimation. *Mol. Biol. Evol.* 11:316–324.

Yang, Z., N. Goldman, and A. Friday. 1995. Maximum likelihood trees from DNA sequences: a peculiar statistical estimation problem. *Syst. Biol.* 44:384–399.

Yang, Z., and B. Rannala. 2012. Molecular phylogenetics: principles and practice [in English]. *Nat. Rev. Genet.* 13:303–314.

Yule, G. U. 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* 213:21–87.

Zelnik-Manor, L., and P. Perona. 2004. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, edited by L. K. Saul, Y. Weiss, and L. Bottou, pp. 1601–1608. Vancouver, Canada.

Zitnik, M., and B. Zupan. 2012. NIMFA: A Python Library for Nonnegative Matrix Factorization. *Journal of Machine Learning Research* 13:849–853.