UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ENABLING COMPARATIVE GENOMICS AT THE SCALE OF HUNDREDS OF SPECIES

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

BIOMOLECULAR ENGINEERING & BIOINFORMATICS

by

Joel Armstrong

September 2019

The Dissertation of Joel Armstrong
is approved:

Professor David Haussler, Chair

Professor Benedict Paten

Paul Flicek, D.Sc.

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Enabling comparative genomics at the scale of hundreds of species

by

Joel Armstrong

Comparing related (homologous) subsequences between genomes from different species gives insight into their function. This information is captured in "genome alignments", which are essential for almost all comparative genomics analyses. However, most existing methods to create a genome alignment suffer from reference-bias (where only one genome is fully aligned to all others), or ignore duplication events. Though the Cactus genome aligner avoided these restrictions, it could not align more than a few genomes without becoming cost-prohibitive as well as losing accuracy. I developed and refined a "progressive alignment" extension to Cactus to allow it to produce a full alignment in time linear in the number of input genomes while maintaining similar, or often improved, quality. This new method allows Cactus to align hundreds of large vertebrate genomes—enabling comparative genomics at an unprecedented scale. During its development I used Cactus as an essential component of several successful comparative genomics projects. Working closely with the 200 Mammals and Bird 10K projects, I have used Cactus to create an alignment of over 600 bird and mammal genomes, which is by far the largest genome alignment ever created. Finally, I have utilized this alignment to provide a highest-possible-resolution annotation of mammalian and avian evolutionary constraint, using the uniquely large number of taxa to enable the examination of weak effects of purifying selection.

For

Grace

Acknowledgments

First, I would like to thank my committee members: David Haussler, Benedict Paten, and Paul Flicek, for their time and guidance. I owe particular debt to two longtime members of the Genomics Institute, Mark Diekhans and Max Haeussler, who set me on this path, encouraging me to join a lab with outsize impact and grand ambitions. All of my labmates, especially John Vivian, Jordan Eizenga, Adam Novak, and Charlie Markello, were not only wonderful to work with, but became great friends. My students, especially Lon Blauvelt, Chang Kim, and Akul Goyal, demonstrated their research aptitude and have already gone on to do impressive things. I thank my funding sources, especially the GENCODE and 200 Mammals projects, for keeping food on my table. Finally, I feel tremendously grateful and uncommonly privileged to have had a career which constantly brought me into close collaboration with people from quite literally all over the planet. To Ed Green, David Ray, and Ed Braun from the archosaur project; Thomas Keane and Matthieu Muffato at the EBI/Sanger; Jeremy Johnson, Diane Genereux, Elinor Karlsson, and Kerstin Lindblad-Toh from 200 Mammals; Qi Fang, Josefin Stiller, Duo Xie, Shaohong Feng, Yuan Deng, James Cahill, Erich Jarvis, and Guojie Zhang from Bird 10K; Ksenia Krasheninnikova and Gaik Tamazian from the Dobzhansky Center; and more — thank you! The work described in this thesis would be quite different without the contribution and advice of my collaborators.

Chapter 1

Introduction

1.1 Preamble

What follows is the text of my review paper "Whole Genome Alignment and Comparative Annotation", co-written with Ian Fiddes and published in Annual Reviews of Genetics. I have omitted the section on comparative annotation as it is not relevant to the main topic of this thesis.

1.2 Genome alignment

1.2.1 Introduction

Alignment is possibly the most fundamental problem in genomics. The alignment problem is to establish a mapping between the letters of a set of sequences that approximates some relation that the user is interested in. In comparative genomics, we are generally interested in the *homology* relation—that is, does the lineage of two bases coalesce at a single base in a

single organism at some (recognizably recent) point in time? In typical real-world comparative genomics, there is no clear proof of homology, as we have absolutely no access to the true history of every base in a set of sequences. However, we can use our knowledge of molecular evolution to construct very good approximations to the homology relation. The potential for using sequence similarity to approximate homology was recognized and applied very early on, starting with the pioneering work of Needleman and Wunsch on optimal pairwise global alignment [88]. The pairwise global alignment work was quickly specialized to perform *local alignment*, which calculates the optimal alignment of subsequences rather than sequences, by Smith and Waterman [113].

The traditional dynamic-programming algorithms require $O(nm)$ time and space, where n and m are the lengths of the two sequences; obviously, as n and m grow to genome-scale the problem becomes too expensive to solve in practice. Another consideration is how genome rearrangements complicate the alignment problem. Smith-Waterman and Needleman-Wunsch both produce alignments which have *fixed order-and-orientation*, that is, insertions, deletions, and substitutions are the only allowed edit operations. When looking within short or well-conserved sequences, like genes, this requirement is usually fulfilled. But at large evolutionary distances and looking within a sufficiently large window, genomes almost always contain more complex rearrangements with respect to each other—inversions, transpositions, and duplications all cause breaks in order and orientation that cannot be captured under constant order and orientation (see Figure 1.2).

As long DNA sequences became available, it was soon recognized that Needleman-Wunsch or Smith-Waterman alignments were far too slow to be useful for megabase-scale

sequences, much less chromosome-scale sequences. The impractical running time of global alignment drove the development of several tools [12, 82, 14] that produce an approximately optimal global alignment through the use of high-confidence *anchors* in a single order and orientation, which are then used to partition the alignment into smaller problems which can be more efficiently solved. These anchors provided a very efficient and reliable way to break up the alignment problem, but relied on a constant order and orientation, which excludes any possibility of noticing rearrangements.

1.2.1.1 What is genome alignment?

One obvious possible solution to the problems of rearrangement and duplication is to use a fast, approximate local alignment algorithm and simply use the collection of all local alignments that it finds as the whole-genome alignment. However, naively applying a local alignment approach has its own problems. Local alignments, when applied at genome-wide scale, have both too low sensitivity and too low specificity to be useful at substantial evolutionary distances. That is, local alignments will miss homologous sequence that, by chance, happened to be further diverged than the sensitivity of the aligner could detect. They will also capture spurious alignments that can obscure more useful data. Even when they correctly identify homologous regions, the end-user is more often interested in *orthology* rather than homology: ancient duplications may share similar sequence, but often do not share similar function. For our purposes in this paper, we call any alignment that allows rearrangements (i.e. does not have fixed order-and-orientation) and attempts to determine orthology rather than just homology (even if restricted to single-copy) a *whole-genome alignment* (or for short, a *genome alignment*). Most

whole-genome alignment methods are based on local alignments, but do some filtering and post-processing to construct a useful end product [9]. Genome alignment tools offer more than simply a collection of local alignments—they must make decisions about where homology begins and mere similarity ends, and additionally must make decisions about what is orthologous and not merely homologous. The size of the problem in whole-genome alignment of large genomes (e.g. mammalians) causes alignments to take too long to be practical, forcing efficiency considerations to be taken into account. At the same time, they must handle genome rearrangements—global aligners cannot properly align genomes that are diverged by even a few millions of years, because the collinearity restriction of global alignment causes so many homologies to be missed.

1.2.1.2 Determining orthology and the single-copy heuristic

Choosing the single "best" target alignment for each region (based on alignment score or percent identity), which we will call the *single-copy* strategy, is a common, if overly simplistic, way [110, 11] to deal with the problems that duplications cause. It is simplistic because the best-fit strategy will not always find a correct ortholog, and indeed even a reciprocally-best-fit is not enough to guarantee finding an ortholog [57]. Perhaps more importantly, choosing a single best sequence ignores lineage-specific duplications. When lineage-specific duplication occurs, a gene outside that lineage will have *multiple* orthologs in the lineage, and should be aligned to multiple copies [68]. Single-copy alignments implicitly assume that orthology is a one-to-one relationship. However, in nature, orthology can often be a one-to-many relationship [68]. When that assumption of one-to-one orthology is violated, single-copy alignments can be very misleading.

1.2.2 Multiple alignment

Often it is necessary to consider the alignment between a set of more than two sequences, which we call *multiple alignment*. A multiple alignment is defined as an equivalence relation $\sim$ on a set of sequences $\mathcal{S} = \{s_1, s_2, \ldots\}$, such that for two bases $b_1 \in s_1 \in \mathcal{S}$ and $b_2 \in s_2 \in \mathcal{S}$, $b_1 \sim b_2$ if they are considered to be aligned to each other. The alignment is partitioned into *columns* by the equivalence classes of $\sim$: i.e. every base is related to all bases in its column, and no two bases in different columns are related. Unfortunately, even simple formulations of the multiple alignment problem are significantly more difficult than their pairwise alignment equivalent and known to be NP-hard [56]. Heuristics must be employed to efficiently solve the multiple alignment problem. *Progressive alignment* is the most popular strategy for approximate multiple alignment [36]. Progressive alignment uses as an additional input a *guide tree* relating the input sequences. The most closely related sequences are aligned first, then the resulting alignment is itself aligned to other sequences or alignments, following the structure of the guide tree. Often consensus sequences are used as a method of aligning alignments.

1.2.3 Reference-free alignment

Since the multiple alignment problem is so difficult, a common heuristic is to use a single *reference* genome to base the alignment on. All other sequences in the multiple alignment are simply aligned to this genome in a pairwise fashion, then the several pairwise alignments are combined to form a *reference-biased* multiple alignment. This approach performs very well when viewed from the reference genome, but information relating genomes distant from the reference is lost. See Figure 1.1 for an illustration of this effect. In the mid- to late-2000s the

first methods for reference-free multiple genome alignment allowing multiple copies began to appear (notably the Enredo-Pecan-Ortheus (EPO) pipeline [97] and the A-Bruijn aligner [104]). The EPO pipeline especially began to see wide use as part of the Ensembl genome browser [2]. While impressive, these pipelines left significant room for improvement, especially with regard to finding small-scale order-and-orientation-breaking rearrangements [97].

1.2.3.1 Genome histories

Alignments are conventionally described as a set of columns, each containing a set of bases that are all related to each other by some alignment relation $\sim$. Usually this relation represents orthology rather than homology. However, in that case, this model falls apart when considering reference-free alignments with multiple copies per genome. The orthology relation is not transitively closed [68], so it is impossible in the general case to create a set of columns containing bases that are all orthologous to each other. The only way to represent a reference-free, multi-copy, orthologous multiple genome alignment is by associating the alignment with phylogenetic trees, which are inferred (even if implicitly) during the alignment process. These trees must be *reconciled* [124] with the species tree so that the duplication events and speciation events are distinguished, to enable the determination of orthology relationships. We term these types of alignments *genome histories* to reflect that they require a different representation than typical alignments (which can be represented by a collection of only blocks or columns).

A *genome history* $\{\mathcal{S}, \sim, T_c, t_s, L\}$ consists of a set of genomes $\mathcal{S}$, a multiple alignment $\sim$ relating the bases of those genomes, a reconciled tree called a *column tree* $t \in T_c$ for each column in that alignment, a species tree t_s, and, optionally, a set of *links* L between columns,

indicating the ordering of the ancestral chromosomes. The columns of the genome history reflect the *homology* rather than *orthology* relation. Since homology is transitive, the homology-based alignment can be represented by columns. The set of trees (hereafter referred to as *column trees*) indicate the evolutionary history of the bases in each column. Where there are duplications, gains, or losses, the column tree $t \in T_c$ will differ from the species tree t_s. Though the genome history representation we present here is not the only possible representation, any other representation (such as a collection of all pairwise orthology relationships) can be transformed into this one.

A genome history can be used to define both *orthology* and *paralogy* relations. The orthology relation, which we will symbolize by $\sim_o$, uses the column trees of the genome history to determine which of the homologous bases in a column are also orthologous to each other. The orthologous bases are those homologous bases whose lineage coalesces in a speciation event in the reconciled column tree [68]. The paralogy relation $\sim_p$ simply relates homologous bases which are not orthologs.

A genome history can be *projected* onto any genome to create a more conventional referenced multiple alignment. These projected, reference-based alignments are collections of columns, each containing exactly one reference base, where every base in the column is orthologous to the reference base, but *not* necessarily orthologous to every other base in the column. These projected alignments are useful because they can be represented in conventional formats like MAF, and used as input to existing analysis tools.

1.2.4 Local alignment tools

Because genome alignment tools usually rely heavily on local alignments of some form, local alignment tools play a large role in genome alignments. Since finding all-against-all optimal local alignments has prohibitive time and memory requirements, approximate local aligners in the vein of BLAST [4] are used almost exclusively. These aligners typically look for short sections of exact matches called *seeds* (which may sometimes include positions which are allowed to vary, to increase sensitivity [80]), and then *extend* the alignment away from those seeds. Local aligners used for genome alignment are often different than read aligners like BWA [75]. Though they use the same basic ideas, local alignment *between* genomes generally involves much more evolutionary distance than read aligners, which are generally optimized for aligning reads to a reference genome which is near-identical to the sample. BLAT [62] is a popular, fast local alignment tool which is useful at short evolutionary distances, though it can handle longer evolutionary distances with its "translated BLAT" translated-protein vs. translated-protein mode. BLASTZ [110] and its successor LASTZ [48] are local aligners tuned to be more sensitive than normal BLAST, using PatternHunter-esque spaced seeds [80], while also allowing transitions for increased sensitivity. LAST [64] is a similar aligner, which can potentially use much smaller seeds than other aligners, without spending time going through uninteresting highly-repetitive alignments, because it extends partial matches until a low enough multiplicity is reached using an efficient substring index.

1.2.5 Genome alignment methods

Most genome aligners, at a high level, work in two stages: *filtering*, where a large number of local alignments are generated and filtered down to remove spurious false-positive alignments and identifying homologous, rearrangement-free regions (*locally collinear blocks* in the terminology used by Mauve [22]), and *refinement*, where the homologous regions undergo alignment with a collinear aligner. (Some aligners keep a subset of the original local alignments as "anchors" to be included in the final alignment, while others throw away all the original local alignments and align the rearrangement-free regions from scratch.) The filtering step can take many different forms, but many involve constructing a graph representation of the alignment and using various heuristics to simplify the graph (for a review, see [61]).

A table summarizing popular or historically significant genome alignment tools is given in Table 1.1 (for pairwise alignment) and Table 1.2 (for multiple alignment). In the following sections, we briefly survey some of the most significant tools.

1.2.5.1 Pairwise genome alignment tools

MUMmer MUMmer [82] is an extremely fast pairwise alignment tool, able to align the human and chimp genomes within less than 4 hours. It achieves this speed by using a suffix-tree data structure to find all maximal unique matches between the two input genomes. Optionally, the "nucmer" script included in the package can perform gapped extension between these matches to generate a more complete alignment. MUMmer is an efficient package for aligning very similar genomes, though as a tradeoff for its impressive speed, its sensitivity, especially with the default settings, is somewhat lower than slower aligners like LASTZ.

Shuffle-LAGAN Shuffle-LAGAN [15] is a pairwise genome alignment tool which aims to draw a compromise between the drawbacks of global and local alignment, using a method which the authors call "glocal" alignment. The method works by first performing an all-against-all local alignment of the two genomes using CHAOS [16], then finding a maximal-scoring *1-monotonic map*, which groups a subset of local alignments into "chains", each of which contains local alignments with only a single order and orientation. This map is restricted so that the chains must be non-decreasing with respect to a single reference genome, while they can be in an arbitrary order in the other genome to represent rearrangements. This allows homology to be detected despite rearrangements, though it will not be able to detect duplications in the non-reference genome. The alignment is then further refined by discarding the local alignments within the chains, and instead realigning the region bounded by each chain with the approximate global aligner LAGAN [14].

Chaining and netting *Chaining* [63] is a powerful technique for making sense of pairwise local alignments. Chains are simply maximal-scoring combinations of local alignments that maintain a single order and orientation. Chaining provides a good way of filtering out spurious alignments, which are likely to form short, low-scoring chains. However, the set of chains can often include distant paralogs or spurious sequence, which makes it difficult to understand the rearrangements that have taken place between the two input genomes. *Netting* [63] is a related technique that makes rearrangements relative to a reference genome much easier to find. In essence, netting finds the best-scoring set of chains that covers the bases of the reference genome only once. This makes it very easy to find high-confidence rearrangements like transpositions,

inversions, and deletions, but removes any duplications in the target genome, instead choosing a single copy to align to.

1.2.5.2 Multiple genome alignment tools

Mauve / progressiveMauve Mauve [22] is a reference-free multiple genome aligner that works by first finding all blocks that contain maximal unique matches from every species to use as anchors. To remove spurious matches, small matches that cause rearrangements are removed, until the alignment can be partitioned into "locally collinear blocks" which are all above a certain size. These blocks are then further refined to attempt to create alignment problems small enough that they can be handled using a conventional collinear multiple aligner, in this case CLUSTAL W [116]. The collection of these multiple alignments forms the final genome alignment.

The original version of Mauve performed poorly in large regions which were present in some but not all genomes, because only blocks containing sequence from every genome were used as anchors. progressiveMauve [23] was developed to relax this restriction. It builds a phylogenetic tree from the input sequences, then uses that tree as a guide to progressively apply an algorithm similar to the original Mauve at each internal node.

Mugsy Mugsy [5] is a reference-free multiple genome aligner which uses a graph-based algorithm to segment a large collection of local alignments into smaller, rearrangement-free sub-problems called "locally collinear blocks", which can be fed into a conventional non-genome multiple aligner. Mugsy first generates all-against-all pairwise alignments using MUMmer [82],

then constructs a graph representation of the local alignment relationships. This graph is used to segment the large alignment problem into smaller "locally collinear" subproblems, which are then aligned using a specialized version of TCoffee [92].

MultiZ / TBA MultiZ [11] is a reference-biased multiple genome alignment tool originally developed as part of the TBA [11] program. Because TBA is restricted to producing multiple alignments that have only a single order-and-orientation (though there exists an unpublished version that removes that restriction), MultiZ sees much wider use than TBA itself. It is the tool currently used to generate the multiple alignments on the UCSC Genome Browser [46].

MultiZ, in effect, is a method of aligning alignments. To produce MultiZ alignments in practice, usually pairwise alignments from a given reference to all other species are generated using a local alignment tool, sometimes post-processed using chains and nets, and then the "autoMZ" command is used to progressively align together these pairwise alignments using a guide tree.

ABA The A-Bruijn alignment method (ABA) [104] uses A-Bruijn graphs (introduced in [100]) to filter a collection of local alignments, removing inconsistencies and small rearrangements using simplification operations on the graph. The method is in principle reference-free if the input alignments are generated in an unbiased way. Though the method was mostly applied to protein alignment (where individual domains are often duplicated and shuffled during evolution), it was also shown to be capable of aligning small chloroplast genomes more completely than TBA.

VISTA-LAGAN VISTA-LAGAN, also known as SuperMap [28], is a reference-free multiple alignment tool built on the Shuffle-LAGAN [15] pairwise alignment algorithm. Unlike Shuffle-LAGAN, VISTA-LAGAN can detect duplications in any genome, not just a reference genome. VISTA-LAGAN progressively aligns each pair of genomes, creating an "ancestral" ordering of the alignment blocks at each step (which is not intended to be an accurate ancestral reconstruction) to continue the alignment to further outgroup genomes.

EPO The Enredo-Pecan-Ortheus (EPO) pipeline [97, 98] is a reference-free multiple alignment pipeline that, unlike TBA, can handle rearrangements. It is in wide use, being one of the main multiple genome alignments available on the Ensembl genome browser [3]. The process begins with a relatively sparse set of anchor points that are known homologies within a set of genomes. The Enredo algorithm builds a sequence graph from these anchors, and through various operations, attempts to remove homologies that are likely to be spurious or uninteresting. The Pecan algorithm then fills in the gaps between the sparse anchors selected by the Enredo algorithm. The Ortheus algorithm [98] is then optionally run to generate ancestral sequences for all blocks, creating a genome history. While EPO is in principle reference-free, the method that is currently used to generate its anchors is reference-biased [97].

Cactus Cactus uses an overall strategy similar in principle to the anchoring approach described above. The notion of a *cactus graph* [95] is used to create a filtered, high-confidence set of anchors. The unaligned space between anchors is then aligned using a sensitive pair-HMM to create a final multiple alignment. The first step of the Cactus process is to take small, uncertain local alignments captured by LASTZ [48] (which is similar to BLAST [4]), and combine them

naively to create a multiple alignment. Given the typical evolutionary distances involved, LASTZ is tuned to be very sensitive, but not very precise. The low precision means that the local alignments may be spurious (a small seed happened to match, and happened to be extended, in a region which is not truly homologous). The local alignments may also conflict—that is, several alignments may disagree on how to align a particular region. These inconsistencies and spurious alignments will manifest as tiny rearrangements—breaks in order and orientation—in the alignment. Using the Cactus Alignment Filter (CAF) algorithm defined in [96], these small rearrangements, which are unlikely to be biological, in the multiple alignment are discovered and removed, producing an alignment that only contains rearrangements longer than a certain length. After this process, the cactus graph contains anchors that are very likely to represent true regions of homology, but will have unaligned regions of homology between the anchors, which local alignment was not sensitive enough to pick up, or which were deleted in the CAF process. The Base Alignment Refinement (BAR) process [96] fills in these unaligned but homologous regions.

progressiveCactus The version of Cactus published in 2011 [96] was highly effective at aligning a small number of genomes in the tens to hundreds of megabases [29], but because it scaled quadratically with the total size of all genomes in the alignment problem, it could not efficiently create the alignments we needed, which require us to align hundreds of vertebrate-sized genomes. Recently a progressive-alignment extension (called progressiveCactus) to the original Cactus algorithm has been developed, which can efficiently scale to hundreds of genomes. The progressiveCactus process works as follows (see Figure 1.3). First, the problem is decomposed into several subproblems using an input guide tree. There is one subproblem per internal node

in the guide tree. Each subproblem involves aligning several genomes using the traditional Cactus process: the *ingroup* (children of the internal node) and *outgroup* (non-descendants of the internal node) genomes for the subproblem. This subproblem alignment is then used to infer a "reference" assembly that contains all blocks involving an ingroup. The blocks are arranged into sequences according to an algorithm that attempts to maximize the consistency between the order and orientation of all the sequences in the alignment [90]. The base-level sequence for these blocks is then generated by finding the ML base for each column using the guide tree. This assembly is a reconstruction of the ancestral genome at that node, which functions as a consensus sequence for the ingroups below it. The reference assembly is then fed as input into subproblems further up the guide tree.

1.2.6 Alignment formats

The fact that genome alignments include the potential for rearrangements and duplications makes representation in collinear alignment formats like aligned-FASTA impossible, because they represent alignments as only a series of insert, delete, and substitution operations. The most popular format for genome alignments currently is the Multiple Alignment Format (MAF). MAF is capable of representing referenced multiple alignments with rearrangements; however, because MAF is a column/block oriented format, it is impossible to represent complex orthology relationships in a reference-free way (see Section 1.2.3.1) without extending the format.

The Hierarchical Alignment Format (HAL) [49] was designed to be an efficiently accessible format representing a genome history, including any ancestral reconstructions avail-

able. HAL allows projection from this genome history onto any reference genome (including ancestors), creating a multiple genome alignment showing what is orthologous (related by $\sim_o$) to every base in that genome. This projection can be output in a traditional format like MAF, or simply used on-demand to visualize the alignment [89] or as part of downstream analysis pipelines.

1.3 Discussion

1.3.1 The future of whole-genome alignment

The average evolutionary distance between sequenced species is getting much shorter as more genomes are sequenced. In the next several years, thousands of genomes will be released due to the efforts of projects like Genome 10K [66] and Insect 5K [108]. By necessity, many comparative genomics projects will focus on alignment between hundreds to thousands of closely related genomes, instead of tens of distantly related genomes. Evaluation on simulated data has shown that while whole-genome aligners vary drastically in accuracy over long evolutionary distances (ranging from an F-score of 0.12–0.80 in a mammal-wide simulated alignment [29]), they all perform extremely well over closer evolutionary distances (ranging from an F-score of 0.97–0.99 in a primate-wide alignment [29]). In some ways, then, genome alignment will become easier because finding homologies is simpler with less evolutionary distance. However, in other ways, it will become more difficult. Creating an alignment of thousands of large genomes is a unique challenge, one which no aligner is currently prepared for. In addition, since the rate of new assemblies being generated will increase, with new assemblies projected to come every few

days or weeks rather than months, maintaining alignments at community comparative genomics resources like Ensembl Compara [2] or the UCSC Genome Browser [46] will necessitate adding new genomes to existing resources piecemeal, rather than regenerating alignments from scratch.

As large sequencing projects produce hundreds to thousands more assemblies in the coming years, reference-bias in multiple genome alignments may become more of a problem. Though reference-biased alignments will serve the genome that they are referenced on well (usually a popular genome like human or mouse), it will certainly be cost-prohibitive to generate a full alignment referenced on all, or even most, new assemblies. This can be a disadvantage to researchers working on non-model organisms, who may not have the resources to run a full alignment referenced on their genome. A reference-free alignment would be more easily shared as a resource useful to many different communities researching many different species. However, reference-free alignment is a substantially more difficult problem than reference-biased alignment. In principle, a reference-free alignment should be equally good for every included genome. However, it may be the case that for any given genome, the quality of a reference-free alignment may be worse than if a new reference-biased alignment were generated referenced on that genome. To live up to their potential, reference-free aligners should aim to equal the quality of reference-biased aligners on reference genomes.

The unique challenges facing genome alignment are twofold: compared to global alignment, the challenge is to capture rearrangements; compared to local alignment, the challenge is to detect orthology rather than mere homology. The Alignathon [29] showed that modern genome aligners generally capture homologies well in the presence of rearrangements. However, orthology detection in modern genome aligners is still very simplistic, and not very accurate.

Many aligners still operate under a single-copy restriction, which, while a useful simplification of the alignment problem, obscures crucial aspects of genome evolution. Others, like Cactus [96] or EPO [97], can support multiple orthology in theory, but in practice use simple heuristics which can often lead to aligning paralogs or missing alignment to orthologs.

Determining orthology accurately, efficiently, and at genome-wide scale is possibly the most difficult unsolved problem in genome alignment. The problem can be framed as building phylogenetic trees for every column in the genome history, after which orthology relationships among the column's bases can be easily established using a reconciliation algorithm [124]. Simply applying maximum-likelihood methods such as RAxML [115] will not be sufficient, for multiple reasons. First, these methods require near-prohibitive amounts of compute power to apply genome-wide in large alignments: building trees from even a relatively small set of 2000 $\tilde{1}$000bp alignment regions among 48 avian species can take over 100 CPU-days to compute [87]. Second, and more importantly, with large numbers of genomes, the size of regions with the same duplication content can become smaller and smaller, leading to less and less phylogenetic information available for any given region, which could increase the chance of errors. It may be helpful to incorporate syntenic information to try to improve the accuracy of finding orthology relationships genome-wide, though it seems that there are still unanswered questions about how best to solve that problem [18]. One advantage of using syntenic information to establish orthology is that it may enable tracking of orthologous *loci* (sometimes called "toporthologs" or "positional orthologs" [24]) in addition to tracking orthologous *sequence*. Keeping track of orthologous loci may be useful to better track gene conversion events, which will cause discrepancies between the toporthology and orthology of a given region.

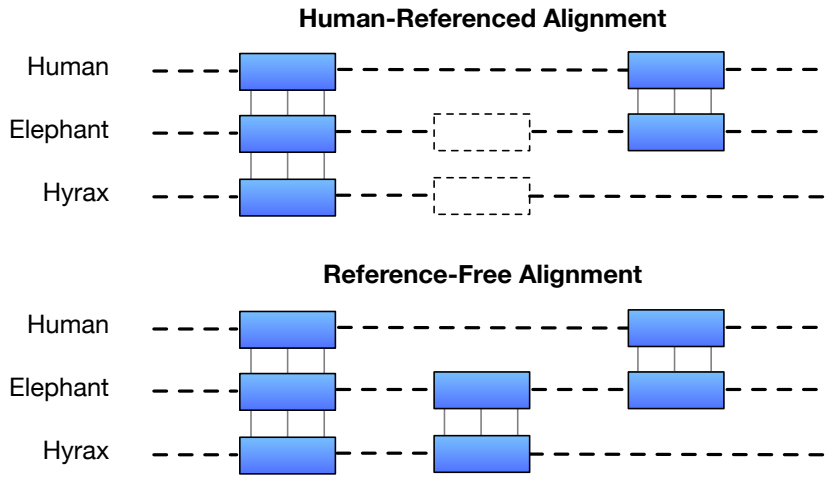


Figure 1.1: A diagram showing the difference between a reference-free and a reference-biased multiple alignment. In a human-biased multiple alignment, any large regions that are deleted in human, or inserted somewhere else in the tree, cannot be aligned.

This review has focused on inter-species comparison, but the future must include more convergence in thinking, models and reasoning about both inter- and intra-species variation. A key, relevant development in modeling population variation is the genome graph[99, 83], which represents variation by encoding individual genomes as paths through a graph structure representing the combined genome alignment. This process allows for variation to be comprehensively captured, and reduces the necessity of depending on a linear reference that does not accurately represent haplotypes present in the population. Extending the genome alignment process to handle *graph-to-graph* alignment, rather than merely sequence-to-sequence alignment, will bring together the fields of comparative and population genomics by enabling the integration of the analysis of inter- and intra-species variation. Such graphs also naturally fit with methods that model uncertainty about ancestral genomes, and therefore some of the software developed for genome graphs might be useful in modeling ancestral genome reconstructions.

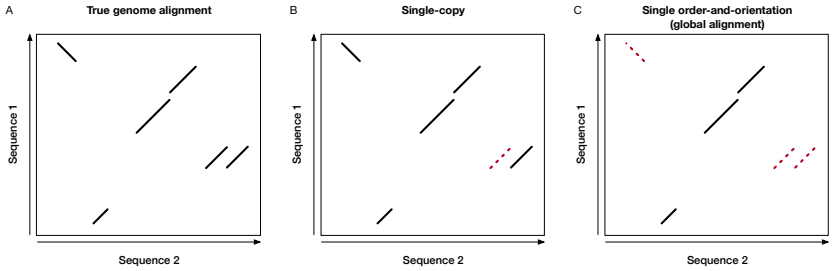


Figure 1.2: An example of how different heuristics affect a genome alignment. All panels are dotplots: a line with positive slope indicates an alignment from the positive strand of sequence 1 to the positive strand of sequence 2, and a negative slope indicates an alignment from the positive strand of sequence 1 to the negative strand of sequence 2. A: The true alignment between the two sequences. B: The same alignment if a single-copy aligner perfectly recovered the true alignment, except for the ignored duplication. C: The same alignment according to a global or approximately-global aligner: no edit operations except insertions, deletions, and substitutions are allowed, so substantial alignment is missing.

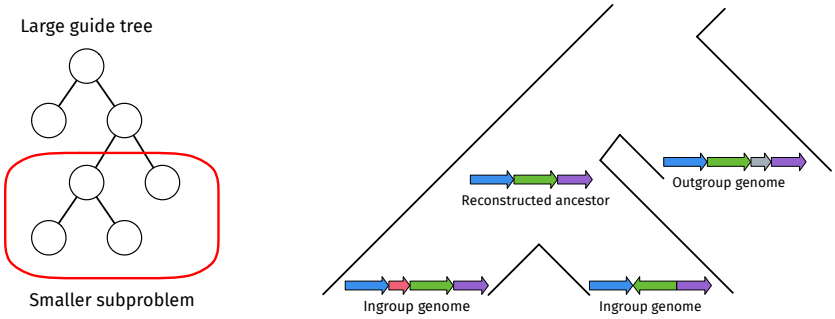


Figure 1.3: An example of how progressive genome alignment works, focused on aligners like VISTA-LAGAN (SuperMap) [28] and progressiveCactus [96] that reconstruct ancestral genomes as input for further alignment steps. A: A large guide tree (usually the species tree), which may include many species, is divided up into smaller local alignment problems of a few genomes each. B: A diagram of what occurs within each subproblem. Each subproblem is focused on reconstructing a single ancestral genome, which is then used as input for subproblems further up the tree. "Ingroup" genomes (children of the ancestor in question) and, optionally, "outgroup" genomes (non-descendants of the ancestor) are aligned together. A plausible ancestral reconstruction is generated for use in later subproblems.

Program	Year	Description
MUMmer	1999 [82]	Fast aligner relying on maximal unique matches from a query sequence to a reference sequence. Recent versions remove the colinearity restriction of the first version and improve the speed.
Chains and nets	2003 [63]	Combines fragmented local alignments into larger, high-scoring "chains", which are arranged into hierarchical "nets" representing rearrangements.
Shuffle-LAGAN	2003 [15]	A "glocal" (global + local) aligner that is less restrictive than global alignment, but still enforces monotonicity of the blocks relative to one sequence.

Table 1.1: Pairwise genome alignment tools.

Program	Year	Reference-bias	Single-copy	Description
TBA	2004 [11]		✓	Collinear multiple aligner (using MultiZ internally) that produces a collection of partially ordered "threaded blocksets."
Mugsy	2011 [5]			Uses a graph-based method to segment the alignment problem into "locally collinear blocks": small subregions with no local rearrangements, which are fed into a collinear multiple aligner.
MultiZ (autoMZ)	2004 [11]	✓	✓	Multiple alignment based on pairwise alignment from every genome to a single reference.
ABA	2004 [104]			Aligner based on the concept of A-Bruijn graphs.
EPO	2008 [97, 98]	*		Graph-based aligner which allows duplications and optionally produces ancestral reconstructions.
VISTA-Lagan (SuperMap)	2009 [28]			Progressive aligner based on Shuffle-LAGAN [15].
Mauve	2004 [22]		✓	Finds maximal unique matches present in every input species, then attempts to remove small matches that cause rearrangements which disrupt collinearity.
progressiveMauve	2010 [23]	✓		Progressive aligner that attempts to remove anchors causing small rearrangements by optimizing a breakpoint-weighted score.
Cactus	2011 [96]			Graph-based aligner that attempts to remove anchors representing small rearrangements.

Table 1.2: Popular and/or historically important multiple genome alignment tools. *: While the core method behind EPO is reference-free, as currently applied its anchor generation is reference-biased.

Chapter 2

Progressive alignment with Cactus: a genome aligner for the thousand-genome era

2.1 Preamble

The following is the main text of the progressive Cactus paper. I wrote the text of this paper, designed all and executed nearly all of the experiments, and contributed the figures. Qi Fang and Duo Xie contributed to the guide-tree analysis and MULTIZ comparison, respectively.

2.2 Introduction

New genome assemblies have been arriving at a rapidly increasing pace, thanks to rapid decreases in sequencing costs and improvements in third-generation sequencing technologies [31, 119, 54]. For example, the number of vertebrate genome assemblies currently in the NCBI database [65] has increased by over 50% in just the past year (to 1485 assemblies as of July

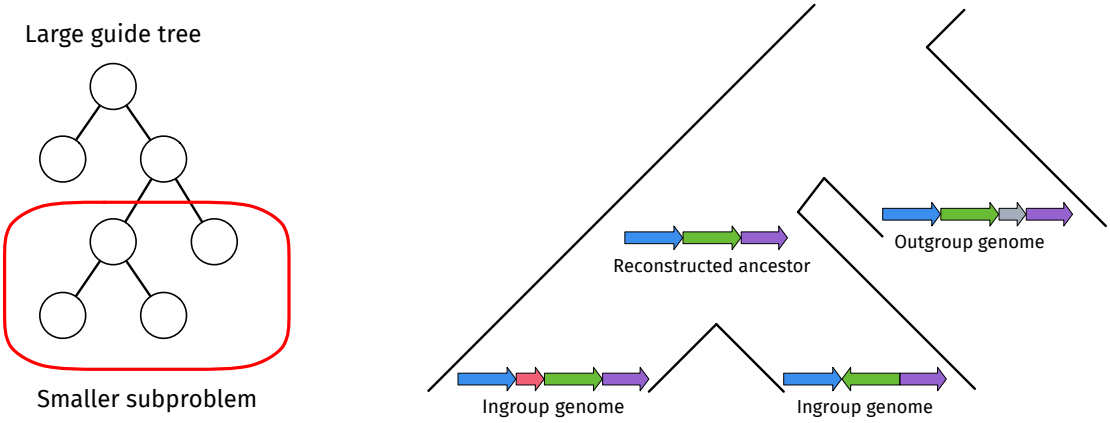
2019). The Vertebrate Genome Project, Genome 10K [66], the Earth BioGenome Project [73], the Bird 10K project [122], and the 200 Mammals project [43], among others, aim to release hundreds of high-quality assemblies of previously unsequenced genomes in the next year, and many thousands over the next decade.In addition to this influx of assemblies from different species, new human *de novo* assemblies [53] are being produced, which enable analysis of not just small polymorphisms, but also complex, large-scale structural differences between human individuals and haplotypes. This coming era and its unprecedented amount of data offers the opportunity to unlock many insights into genome evolution, but also presents challenges in adapting our analysis methods to meet the increased scale.Often we want to make use of these assemblies to conduct analyses like species-tree inference, comparative annotation [37, 67], or constraint detection [50, 41]. All of these require comparing an assembly against one or more other assemblies. This involves creating a mapping from each region of each genome to a corresponding region in each other genome, taking into account the possibility of complex rearrangements: this is the problem of creating a *genome alignment* [6].Genome aligners are one of the most fundamental tools used in comparative genomics, but since the problem is difficult, different aligners frequently give somewhat different results [29], and many intentionally limit the alignments they produce to simplify the problem. Two of the most common limitations are *reference-bias*, which constrains a multiple alignment to only regions present in a single reference genome, and restricting the alignment to be *single-copy*, which allows only a single alignment in any column in any given genome, causing the alignment to miss multiple-orthology relationships created by lineage-specific duplications. Cactus [96] is a genome alignment program which has neither of these restrictions; it is capable of generating a reference-free multiple alignment that

allows detecting multiple-orthology relationships.The version of Cactus available at the time performed very well in the Alignathon [29], an evaluation of genome aligners. However, the runtime of that initial iteration of Cactus scaled quadratically with the total number of bases in the alignment problem, making alignment of more than about ten vertebrate genomes completely impractical. To address these difficulties, we present fundamental changes to the Cactus process that incorporate a progressive alignment [36] strategy, which changes the runtime of the alignment to scale linearly with the number of genomes. We show that the result is an aligner that remains state-of-the-art in accuracy, and continues to lack reference bias, but which is tractable to use on hundreds to thousands of large, vertebrate-sized input genomes. This new version of Cactus has been developed over several years, and has already been successfully used as an integral component of high-profile comparative genomics projects [45, 26, 44, 77, 70]. We describe the many improvements to the original Cactus method that make large alignments tractable, while also increasing the accuracy of those alignments. We demonstrate it is capable of creating useful alignments across a wide range of evolutionary distances, from intra-species alignments useful in population genetics [42] to inter-species alignments spanning hundreds of millions of years of genome evolution. Because of its support for multiple-orthology relationships, it automatically supports diploid assemblies, which are becoming more common as new technologies enable phasing across long distances [69, 119].

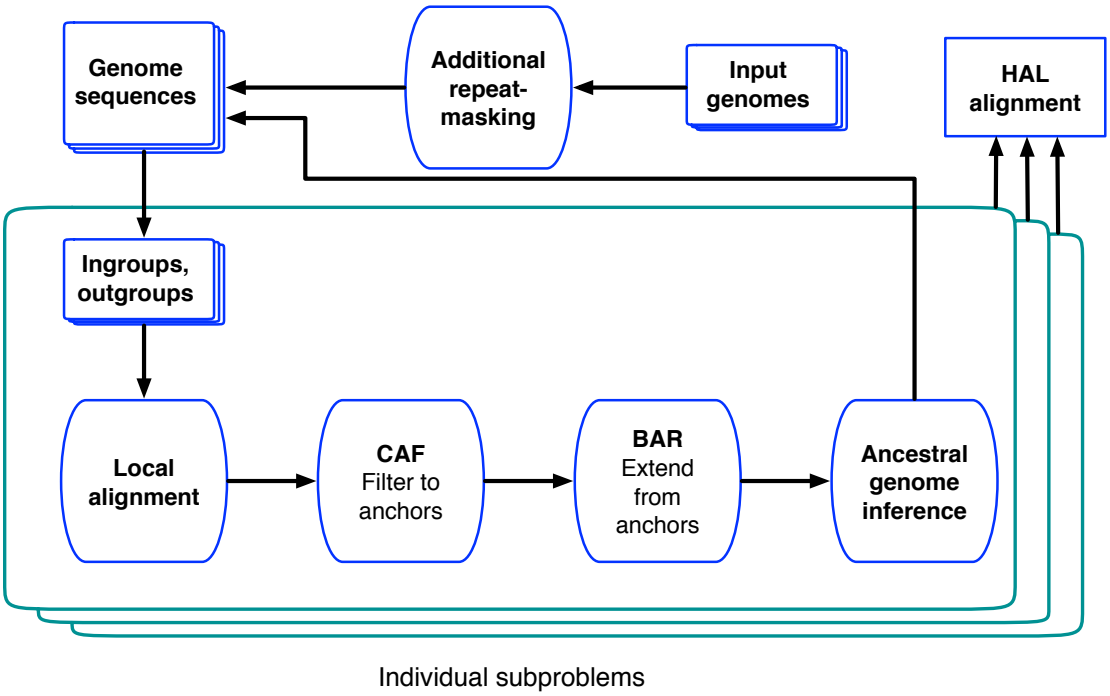
2.3 Results

2.3.1 Cactus

The new progressive Cactus pipeline is freely available and open source. The only inputs needed are a guide tree and a FASTA file for each genome assembly.The key innovation of the new Cactus aligner is to adapt the classic *progressive* strategy (used in collinear multiple alignment for decades [36]) to a whole-genome alignment setting. Progressive aligners use a *guide tree* to recursively break a multiple alignment problem into many smaller sub-alignments, each of which is solved independently; the resulting sub-alignments are themselves aligned together according to the tree structure to create the final alignment. Progressive alignment has been succesfully applied to whole-genome alignment before, for example by progressiveMauve [23] and TBA/MULTIZ [11]. Cactus now follows a similar strategy, with the key innovation being that Cactus implements a progressive-alignment strategy for whole-genome alignment using reconstructed ancestral assemblies as the method for combining sub-alignments. This strategy not only results in a much faster alignment runtime, but also produces ancestral reconstructions.As a practical matter, Cactus also now uses the Toil [117] workflow framework to organize and distribute its computational tasks. Because it runs on Toil and supports container execution via Docker and Singularity [72], Cactus can be run on many different environments: single machines (for small alignments), conventional HPC clusters, as well as the GCP, AWS, and Azure clouds.Figure 2.1A shows the overall organization of the new Cactus process. The guide tree, which need not be fully resolved (binary), is used to recursively split a large alignment problem (comparing every genome to every other genome) into many small subproblems, each of



(a)



(b)

Figure 2.1: A diagram of the progressive process within Cactus. A:A large alignment problem is decomposed into many smaller subproblems using an input guide tree. Each subproblem compares a set of ingroup genomes (the children of the internal node to be reconstructed) against each other as well as a sample of outgroup genomes (non-descendants of the internal node in question). B:This flowchart represents the phases which each subproblem alignment proceeds through.The end result is a new genome assembly representing Cactus's reconstruction of the ancestral genome, as well as an alignment between this ancestral genome and its children. After all subproblems have been completed, the parent-child alignments are combined to create the full reference-free alignment in the HAL [49] format.

which compares only a small number (usually 2–5) of genomes against one another. The purpose of each subproblem is to reconstruct an ancestral assembly at each internal node in the guide tree, as well as to generate alignments between that internal node's children and its ancestral reconstruction. The ancestral assemblies are then used as input genomes in subproblems further up the tree, while the parent-child alignments are later combined to produce the full alignment. Two sets of genomes are considered: the children of the internal node (which we call the *ingroup genomes*), and a set of non-descendants of that node (the *outgroup genomes*). The ingroup genomes form the core alignment relationship being established at this node. The outgroup genomes serve to answer the question of what sequence from the ingroups is also present in the ancestor (whether an indel among the ingroups is likely a deletion rather than an insertion), and in how many copies (whether a duplication predates or postdates the speciation event the node represents). The outgroups also provide information for guiding the ancestral assembly by providing additional order-and-orientation information, as well as additional information when generating ancestral base calls. These genome sets are used as the input to the main subproblem workflow, which we outline below and in Figure 2.1B, and describe in detail in Section 2.5.1.Each individual subproblem follows a procedure akin to the original Cactus process. The subproblem procedure begins with a set of pairwise local alignments generated via LASTZ [48]. These pairwise alignments are then filtered and combined into a cactus graph representing an initial multiple alignment using the CAF algorithm described in our earlier work [96], though we note important changes to the filtering in Section 2.5.1.4 and Section 2.5.1.5. The initial alignment is refined using the BAR algorithm again described in earlier work [96] to create a more complete alignment. The ancestral assembly is then created by ordering the blocks in this final alignment

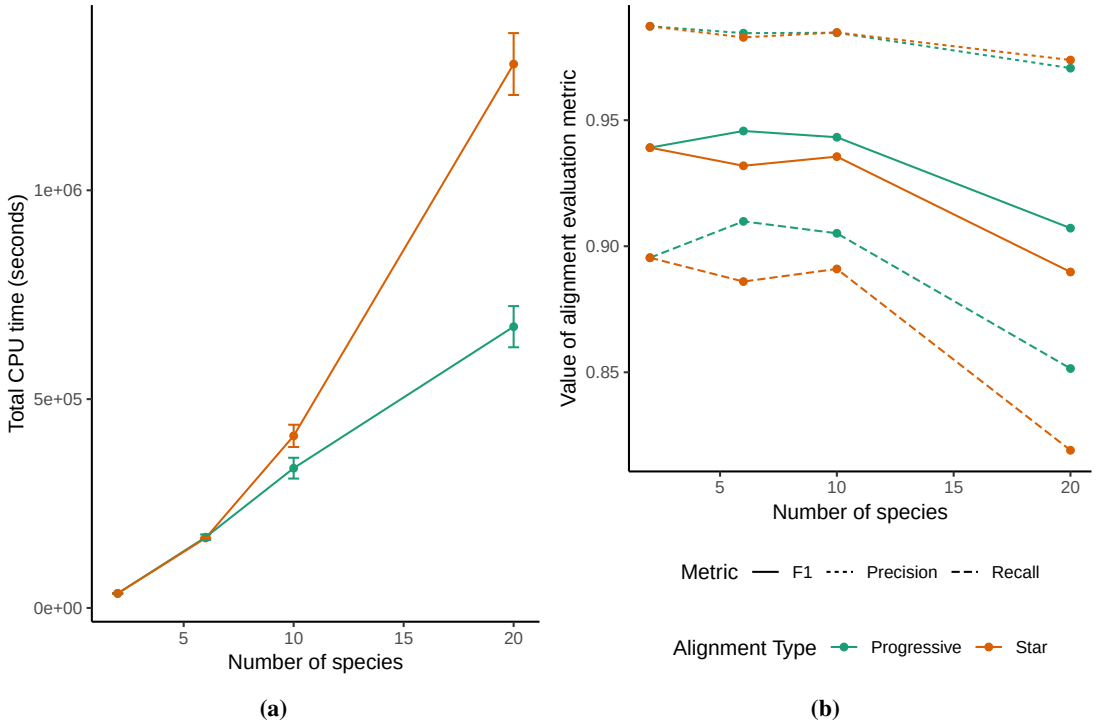


Figure 2.2: Results from alignments of varying numbers of simulated genomes using the progressive mode of Cactus ("Progressive"), versus the mode without progressive decomposition similar to originally described in [96] ("Star"). A) The total runtime of the two alignment methods across 3 runs. The runtime is nearly identical when aligning two genomes since the alignment problem is not further decomposed, but the linear scaling of the progressive mode means it is much faster with large numbers of genomes than the quadratic scaling required without progressive alignment. B) The precision, recall, and F1 score (harmonic mean of precision and recall) of aligned pairs for each alignment compared to pairs from the true alignment produced by the simulation.

and establishing a most-likely base call for each column in each block. The resulting ancestral sequence is then fed into later subproblems (unless the subproblem represents the root of the guide tree, which indicates the end of the alignment process).

2.3.2 Evaluation on simulated data

To evaluate the improvements in quality and runtime of the alignments produced using the new progressive alignment strategy, we simulated the evolution of 20 30-megabase genomes using Evolver [30] along a tree of catarrhines. We ran two alignment strategies — one using a fully-resolved binary guide tree (which takes full advantage of the new progressive mode) and one using a fully-unresolved star guide tree (which is similar to the originally published version of Cactus) — across variously sized subsets of these genomes (for details of the simulation and alignments, see Section A.0.1). The alignments using the progressive strategy were much faster, especially when aligning large numbers of genomes, as expected given its linear scaling runtime, as opposed to the quadratic scaling of the star-tree (Figure 2.2A). The simulated genomes have a known true alignment relating them, which is produced during the simulation process; using this it is possible to evaluate the quality of the alignments produced by the two strategies (Figure 2.2B). The progressive strategy is significantly more sensitive (89% recall) than the star strategy (82% recall) when aligning all 20 genomes: this reflects the fact that the increasing number of genomes will decrease the length of rearrangement-free regions, limiting the effectiveness of the rearrangement-based alignment filtering method that Cactus utilizes. Since the progressive strategy only aligns a constant number of genomes together at a time, it is able not only to make the runtime of aligning large numbers of genomes practical, but to align them with an accuracy unattainable by the previous versions of Cactus.

Alignment	Precision	Recall	F1-score	CPU time
Genome added to branch	97.27%	88.40%	92.63%	11 h
Genome added to node	97.21%	88.35%	92.57%	16 h
Full realignment of entire tree & new genome	97.19%	88.33%	92.55%	176 h

Table 2.1: Results of adding a new genome to an alignment of simulated genomes. Precision, recall, and F1-score statistics are all of aligned pairs that contain a base of the added genome. An alignment where the genome was included initially is shown for comparison.

2.3.3 Adding new genomes to an existing alignment

Given the rate of arrival of new assembly versions and newly sequenced genomes, adding new information to an alignment without recomputing it from scratch is valuable, especially for large alignments where recomputing the entire alignment is often cost-prohibitive.Cactus supports adding a new genome to an existing alignment by taking advantage of the tree structure of the progressive alignments it produces. There are three ways that a new genome can be added to an alignment, depending on its phylogenetic position relative to the existing genomes: 1) as outgroup to all the existing genomes in the alignment, 2) by being added as a new child of an existing ancestral genome in the alignment, or 3) by splitting a branch in the existing alignment, creating a new internal node and two new branches (Figure A.1). Cactus allows adding a new genome in any of these ways, though the details differ; see Section 2.5.2. Assemblies can be replaced with new versions by simply deleting them and adding the new assembly in as a leaf. Adding multiple genomes is possible, either iteratively or (if the new genomes are monophyletic) by aligning together the new genomes and adding in the ancestral clade root.We tested the effect of adding a new genome to an existing alignment using the same set of simulated catarrhine genomes as in Section 2.3.2. To replicate the use-case of an end-user wanting to add a genome

Guide tree	Jarvis	Prum	Consensus	Permuted
Jarvis	1			
Prum	0.9867	1		
Consensus	0.9882	0.9883	1	
Permuted	0.9843	0.9822	0.9836	1

Table 2.2: Comparison of alignment similarity between four alignments of the same 48 avian genomes with different guide trees. Similarity between each pair of alignments is represented by the F1 score (harmonic mean of precision and recall) of aligned-pair relationships in the two alignments.

to a previously-created alignment, we generated an alignment holding out one of the 20 genomes (the crab-eating macaque), and added that genome back into the alignment by both splitting an existing branch (resulting in the same topology as a full alignment would), and by adding the macaque as a new child of an existing ancestor (creating a trifurcation which did not exist in the original tree). For details of this process, see Section A.0.2. Both methods resulted in alignments that had accuracy nearly identical to the full alignment that included the macaque from the start: both addition methods as well as the full alignment achieved an F1 score of 0.926 (Table 2.1).

2.3.4 Effect of the guide tree

Since Cactus uses an input guide tree to decompose the alignment problem, the guide tree can potentially impact the resulting alignment. This could be problematic when the exact species tree relating the input set of genomes is unknown or controversial. However, Cactus aims to reduce any effect of the guide tree by including a great deal of outgroup information, including multiple outgroups when possible. To quantify the effect of the guide tree on a large alignment with an uncertain species tree, we created four alignments of a set of 48 avian species, which we subsetted down to a single chromosome (Chromosome 1). The avian species tree is somewhat

debated, with many different plausible hypotheses [55, 103], making birds an excellent test case with no single clearly correct guide tree. We aligned these birds using four different guide trees: two trees that represent two different hypotheses about the avian species tree [55, 103], one consensus tree between the former two trees, and one tree that was randomly permuted from the Jarvis et al. tree [55] (see Section A.0.3 for details on the alignments and Figure A.2 for a visualization of the four guide trees). The four alignments were highly similar, with an average of 98.5% of aligned pairs exactly identical between any two different alignments: detailed results are shown in Table 2.2.

2.3.5 Timing duplication events

Users of a genome alignment are almost always interested in *orthology*, rather than *homology*, between a set of sequences. For example, when comparing human and chimpanzee KZNF genes, providing an alignment from each gene to the over-400 [51] homologous KZNF genes in the other genome is nigh-useless; the user is likely interested in only the orthologous copy or copies (in the case of a lineage-specific duplication) in the other genome. For this reason, Cactus alignments are capable of representing complex orthology/paralogy relationships, with an ability to display the alignment(s) labeled as orthologous, but also the option for a user to request alignments to paralogs at a customizable coalescence-time threshold. This is achieved by implicitly producing a gene tree as the alignment is built, albeit with some restrictions imposed by the output HAL [49] format, namely that a duplication event is represented by multiple regions in the child(ren) aligned to a single region in the parent species. This forbids the representation of gene-tree-species-tree discordance as would occur in incomplete lineage-sorting or horizontal

transfer, as well as the exact ordering of multiple duplication events along a single branch. The restricted problem we solve at each subproblem step is that each block should represent all regions orthologous to a single region of the ancestral sequence, possibly multiple per species; we make no attempt to fully resolve the gene tree when multiple duplications take place along a single branch. However, this still requires resolving the timing of all duplication events: duplicated sequences whose coalescence precedes the speciation event represented in the subproblem should be split, while those following the speciation event should be kept together.Because it is impractical to generate maximum-likelihood trees for every block in the subproblem, Cactus relies on heuristically filtering alignments to remove paralogs before building its cactus graph. For this we developed two heuristics: a filter based on similarity to outgroup sequence, which was used in the many projects which used the beta versions of progressive Cactus, and (more recently) a method of pre-filtering alignments that only allows any given base to contribute one "best" alignment in most cases (described in Section 2.5.1.4). Of the two methods, the newer best-hit filtering removes many more likely-paralogous alignments, especially to closely-related genomes, while leaving approximately the same amount of sequence covered by a single homology. For example, in two comparison alignments of the same 12 genomes, one using the best-hit filtering and one using the outgroup filtering, the amount of human sequence mapping to two or more places in the chimpanzee genome was reduced from 6.1% to 2.6%, while the total amount of human covered by chimpanzee actually increased despite the removed homologies (see Figure 2.3A,B for an example visualization and Figure 2.3C for aggregate statistics; see Section A.0.4.1 for details on the alignments).To confirm that these improvements were likely caused by removal of paralogous rather than orthologous alignments, we compared phylogenetic

trees implicit in the columns of HAL alignments to independently re-estimated approximately-ML trees produced by FastTree [102] for the same regions Section A.0.4.3. Since HAL does not produce a fully binarized history of duplication events, we compared the species assigned to the most recent common ancestor (MRCA) of randomly selected pairs of sites from genomes containing a duplication within the column.If the species assigned to the MRCA in the HAL tree is a descendant of the species within the reconciled ML tree, that implies that there are paralogs represented as orthologs within the HAL tree (since a duplication event must have been resolved too early). Similarly, if the MRCA species within the HAL tree is an ancestor of that within the reconciled ML tree, a duplication event must have been resolved too late in the HAL, implying additional false loss / deletion events.The number of paralogous alignments (represented by the coalescence time between duplicated sequences being too "early" in the HAL tree relative to the ML tree) in the alignment of the 12 boreoeutherian genomes was clearly reduced (46% in the outgroup filtering vs 26% in the best-hit filtering) (Figure 2.3D).We separately ran the Comparative Annotation Toolkit (CAT) [37] on identical chimpanzee and gorilla assemblies in two alignments using the outgroup and best-hit filtering methods (Section A.0.4.2). Not only was CAT less likely to identify a human gene in multiple chimp loci using the best-hit filtering (e.g. 6.5% vs. 9.8% multiple-mappings across all genes in chimp, and 5.9% vs. 13.8% for the recently-duplicated KRAB zinc-finger gene family) (Figure 2.3E), but as a result orthologs for 104 more human genes were identified in the output gene set for chimp (182 in gorilla) (Table A.3). This is likely because tens of thousands fewer paralogous transcripts were filtered out in the initial filtering phase of CAT (Table A.2), reducing confusion about which transcript projection to put into the gene set.

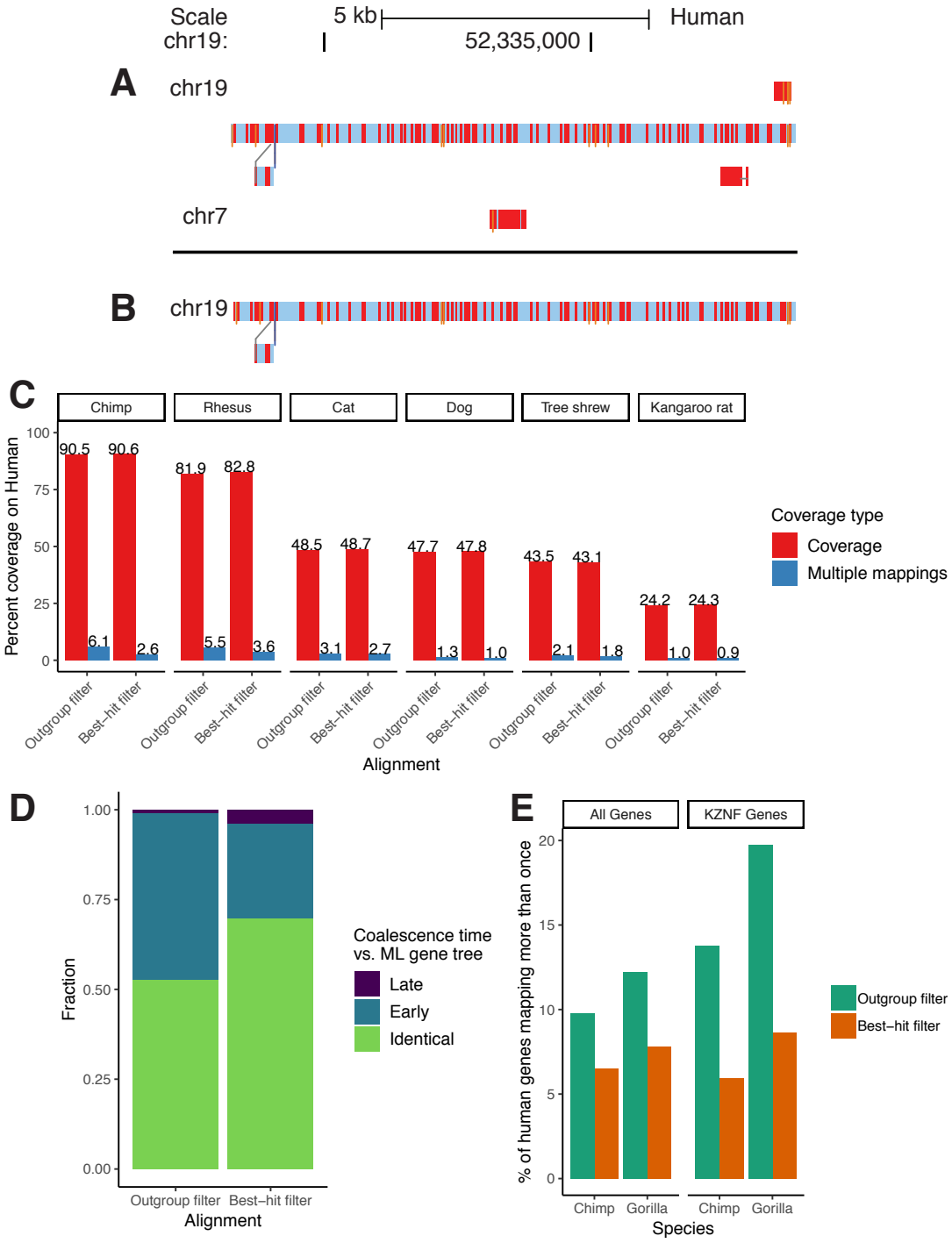


Figure 2.3: Results from the improved paralog-filtering method. A/B: A sample snake track [89] within a recently duplicated region before (A) and after (B) the filtering change. Nucleotide subsitutions are shown as red bars, and insertions are shown as thin orange bars. C: Coverage results from two alignments of identical assemblies using the outgroup and best-hit filtering methods. Multiple-mappings: sites which map to two or more sites on the target genome. D: Results from comparing phylogenetic trees implicit in the HAL alignment to ML re-estimated trees of the same regions. "Early" coalescences imply that too many duplication events have been created in the reconciled trees, while "Late" implies that too many loss events have been created. E: Percent of human genes that map more than once to the chimp/gorilla genomes in two CAT [37] annotations using alignments created with the outgroup and best-hit filtering methods. KZNF: KRAB zinc-finger genes.

2.3.6 600-way amniote alignment

To demonstrate this new version of Cactus we present results from an alignment of 605 amniote genomes, relating in a reference-free manner a total of over 1 trillion bases of DNA across hundreds of millions of years of genome evolution. The amniote-wide alignment combines two smaller alignments: one created for the 200 Mammals project [43], representing 242 placental mammals, and one for the Bird 10K project [122], which relates 363 avians. The overall topology is shown in Figure 2.4A. To our knowledge this represents the largest whole-genome alignment yet created. Table 2.3 contains aggregate statistics on this alignment, which was computed using the Amazon Web Services (AWS) cloud infrastructure (for details on the construction, see Section A.0.6).Coverage within the 600-way alignment unsurprisingly closely tracks phylogenetic distance and genome size, with e.g. a median coverage on human of 2.3 Gb from Euarchonta species, vs. 1.2 Gb from Laurasiatheria species and 1.0 Gb from Glires species (Figure 2.4B,C). The ancestral reconstructions within the 600-way alignment are highly complete, especially for conserved sequence: 86% of human coding bases are represented in our reconstruction of the ancestor of all placental mammals, while 95% of chicken coding bases are represented in our reconstruction of the common ancestor of avians (Figure 2.4D,E). The ancestral assemblies consistently contain a relatively higher proportion of avian than mammal sequence even across similar phylogenetic distance, reflecting a much more conservative mode of genome evolution in avians as well as the lower repeat content and denser gene content of avian genomes [123].The reference-free nature of Cactus alignments enables examining genome evolution along all branches equally well, rather than being restricted to sequence present in

one reference genome. In addition, the ancestral reconstructions implicitly provide a history of substitution, indel, and rearrangement events. Though this history is by its nature only a hypothetical reconstruction of the true history of genome evolution along the tree, it is by and large accurate enough to be useful. To demonstrate the utility of the indel history, we examined rates of small ($\leq$ 20 bp) insertion and deletion events in the 600-way alignment. As expected given previous studies [19, 45], the rate of small indels in any given branch was correlated with the rate of nucleotide substitution (an R^2 of 0.49 for insertions and 0.59 for deletions), though remained much lower (1.3% of the substitution rate for insertions, and 1.7% of the substitution rate for deletions) (Figure 2.5A).The ancestral assemblies also represent even difficult-to-align regions such as transposable elements. We ran RepeatMasker [112] on several human ancestors, focusing on the recently-emerged L1PA6 family of L1 retrotransposons. When ascending the primate tree, approaching the origin of modern L1PA6 elements above the human-rhesus ancestor, L1PA6 elements appear increasingly similar to their consensus sequence (Figure 2.5B).The Bird 10K species were also separately aligned using MULTIZ [11] using the chicken genome as the reference, allowing us to make a comparison between the two resulting alignments. Cactus aligned more total bases to chicken than MULTIZ (an average of 69.4% of the chicken genome compared to an average of 64.9%, for an average increase of 47 Mb). Since, unlike Cactus, MULTIZ is reference-biased, the difference is more stark when looking at the number of bases aligned to a genome not used as the MULTIZ reference (an average of 79% of the zebra finch covered vs. 49.2%, for an average increase of 367Mb: see Figure 2.6).

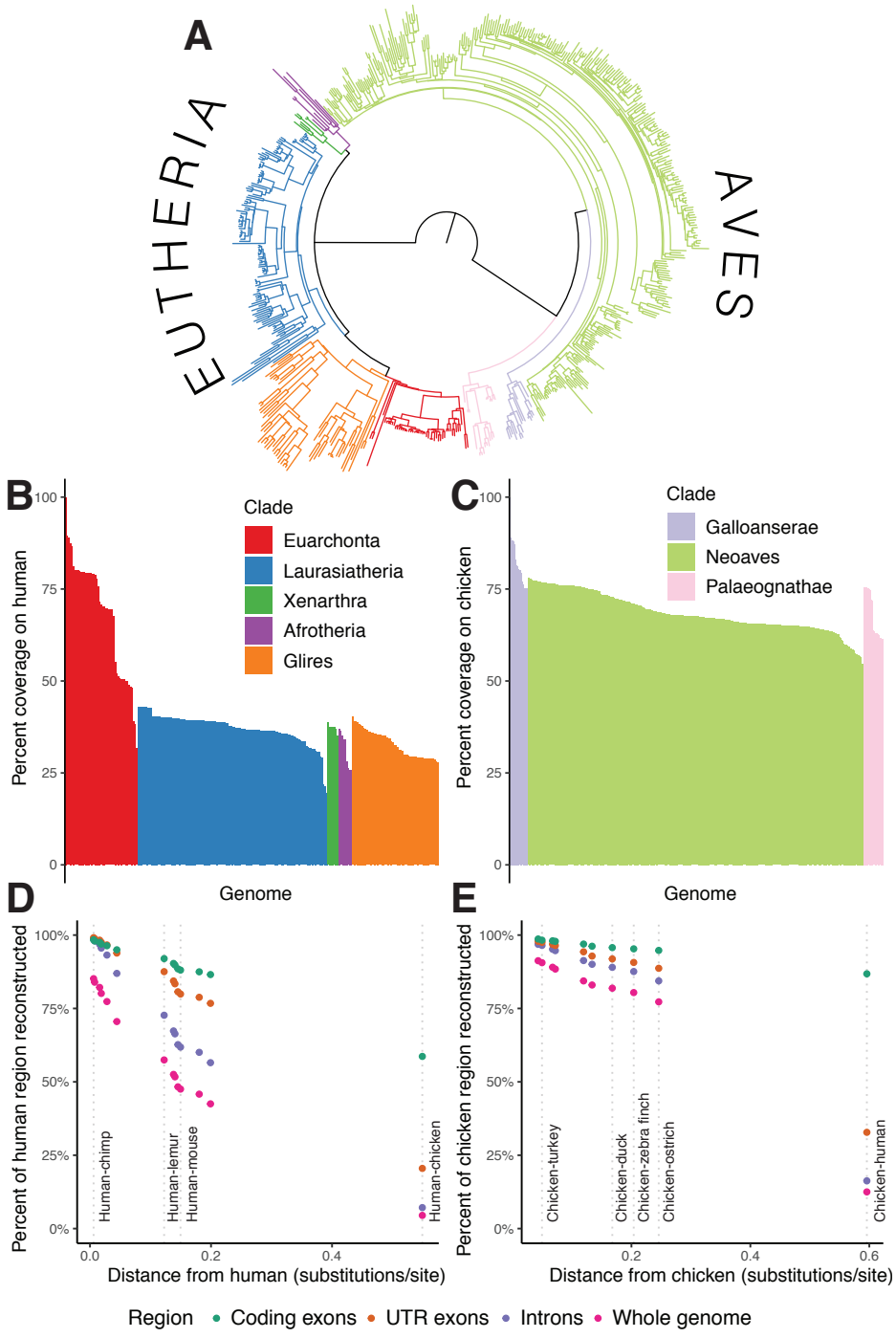
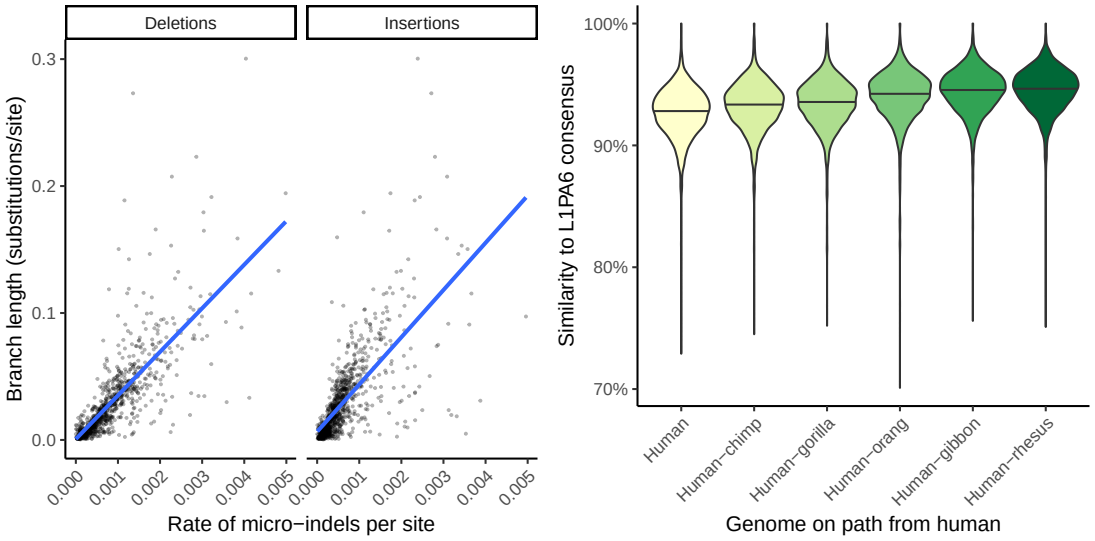


Figure 2.4: Results from the 600-way amniote alignment. A: The species tree relating the 600 genomes. Branches are colored by clades in the same way as figures B and C. B: Percent coverage on human within the eutherian mammals, grouped by clade from highest to lowest coverage. C: Similar to B, but for coverage on chicken within the avian alignment. D: Percent of various regions within the human genome mappable to each ancestral genome reconstructed along the path leading from human to the root. The positions of selected ancestors are labeled by dotted lines to indicate useful taxonomic reference points as context. E: Similar to D, but for the path of reconstructed ancestors between chicken and the root.



(a) **(b)**

Figure 2.5: A: Rates of micro-insertions and -deletions (micro-indels) along each branch within the 600-way, compared to conventional substitutions/site branch length. B: Violin plot showing the increasing similarity to consensus of L1PA6 elements within reconstructed ancestral genomes along the path to the emergence of modern L1PA6 elements (in the human-rhesus ancestor).

Alignment	# of genomes	Total bases	Instance-hours	Core-hours	Common ancestor size
200 Mammals	242	669 billion	68,166	1.9 million	1.73 Gb
Bird 10K	363	400 billion	5,302	0.2 million	1.13 Gb
Combined	605	1.07 trillion	73,692	2.1 million	181 Mb

Table 2.3: Aggregate statistics for the 600-way alignment. The increase in computational work for the mammal alignment over the bird alignment is largely caused by the increase in the pairwise alignment phase runtime, because it scales quadratically with the size of the genomes being aligned.

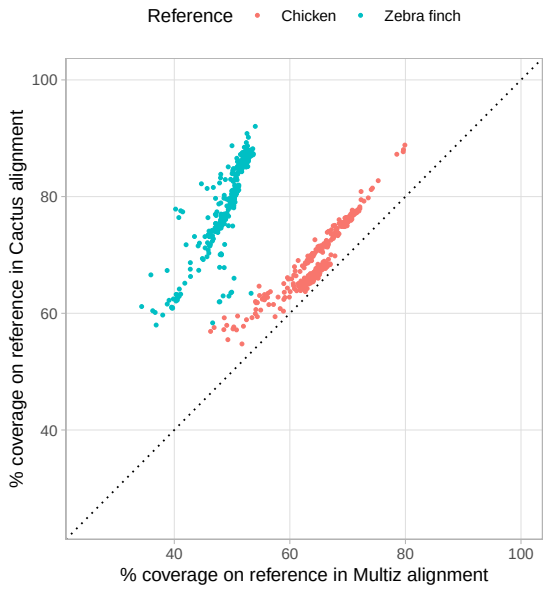


Figure 2.6: A comparison of coverage in the Cactus avian alignment compared to a chicken-referenced MULTIZ [11] alignment of the same genomes. Coverage of both alignments on chicken and zebra finch is shown to illustrate the effects of reference-bias on the completeness of the MULTIZ alignment.

2.4 Discussion

A few ambitious comparative genomics projects are already producing assemblies at the scale of tens to hundreds of species, and we anticipate that this scale of data will become much more common in the coming years. However, without a genome alignment it is impossible to relate these assemblies, and making an accurate genome alignment that large is difficult. We have demonstrated that Cactus can create alignments of hundreds of large genomes efficiently by producing an alignment relating over a trillion bases total. With this new development, we not only enable high-quality genome alignments for these projects, but also hope to set the stage for analysis of thousands to tens-of-thousands of genomes in the near future.Furthermore, as long-read technologies become cheaper and more widely accessible, assembly quality has been rising. The age of having only a few high-quality vertebrate assemblies, like human or mouse, is at its end. As more assemblies converge on the gold-standard, "reference" level of quality displayed by GRCh38 and GRCm38, a reference-free genome alignment becomes increasingly useful. A reference-biased alignment forces the user to view genome evolution through the lens of a single, distant reference. As the average assembly becomes ever more complete and accurate, this missed opportunity to analyze regions not present in the reference grows even worse as more data is ignored and does not contribute to the alignment. For this reason, we provide a reference-free alignment, allowing analysis of genome evolution throughout the entire tree rather than in comparison to one anointed reference.Cactus is also useful for comparison between assemblies of the same species, not just comparison between species. Often a sequencing effort will produce multiple *de novo* assemblies from different individuals, or

diploid assemblies from a single individual. Alignments of these assemblies are essential for many analyses, e.g. annotation of *de novo* assemblies [37]. Cactus is easily capable of capturing even the most complex structural variation, such as copy number variation, between these assemblies.Producing a genome alignment has usually been an arcane task, where parameters used to produce, chain, or filter the input local alignments can have an under-appreciated effect on the result. We provide Cactus as an integrated pipeline that can be used across many different compute environments, but especially thrives on modern cloud environments. It intelligently adjusts alignment parameters to maximize efficiency and accuracy depending on evolutionary distance. While genome alignment is a computationally intensive task, we have broken up the problem into small pieces that can work in heterogeneous clusters, playing to the advantages of both cheap CPU-rich machines and more expensive memory-rich machines.We have used Cactus to produce a 600-way alignment, which is, to our knowledge, the largest-yet genome alignment of vertebrates. This alignment is already proving useful for further downstream analysis. The Bird 10K [122] and 200 Mammals [43] consortia plan to use the alignment to analyze selection at unprecedented detail across avians and mammals, respectively.In the quest to make Cactus more efficient, optimizing the local alignment phase would offer the most return because the computational cost of the alignment is dominated by the generation of local alignments. Some less-sensitive local alignment programs are naturally more efficient than LASTZ, which is tuned for high sensitivity and long evolutionary distances. Making the local alignment phase a "pluggable" module, in which methods of generating the local alignments, or even the initial sequence graph, could be easily swapped out would be a fruitful avenue for experimentation. Cactus could potentially transition between using a less sensitive local aligner for closely related

sequence and a more sensitive aligner across long evolutionary distance, much in the same way that we change alignment parameters based on evolutionary distance today.As alignments become larger and more expensive to compute, it becomes much more important to be able to update them (by e.g. adding a new genome or updating an assembly) without recomputing the entire alignment. Cactus's progressive alignment framework, combined with special functionality in the HAL toolkit [49] makes it possible to make these changes very efficiently: costing only a single subproblem's worth of computation time, usually about 120 CPU days. However, there is currently an appreciable amount of manual work involved in the process of adding, removing, or updating an assembly within an existing alignment. Making this simpler and more automated would be an interesting future direction, one that would potentially allow a very large alignment resource to be used and updated for years, with collaborators adding in their genomes of interest cost-effectively.

2.5 Methods

2.5.1 Cactus

The Cactus pipeline is available at `https://github.com/ComparativeGenomicsToolkit/cactus`. The exact version of Cactus used for each of the analyses described above varies; the commit used in each analysis are available in the supplementary material.

2.5.1.1 Preliminary repeat-masking

Cactus requires input genomes to be soft-masked, but often repetitive sequence goes unmasked due to poor masking or incomplete repeat libraries for newly-sequenced species. This can negatively affect alignment runtimes (as alignments need to be enumerated to and from all copies of a repetitive sequence) and impact quality. For this reason, we mask overabundant sequence before alignment, using a strategy not based on alignment to repeat consensus libraries, but on over-representation of alignments. We first divide each genome into small, mutually overlapping chunks. For each chunk, we align it to itself and a configurable amount of other randomly sampled chunks (currently 20% of the total pool). To avoid combinatorial explosion due to unmasked repetitive sequence, we use a special mode of LASTZ [48] which stops exploring alignments from any region early if a maximum depth is reached. We then soft-mask any region covered by more than a certain configurable number of these alignments (currently set to 50).

2.5.1.2 Local alignment and outgroup selection

The alignment process for each subproblem begins with a series of local alignments generated using LASTZ [48]. The local alignments fall into two sets: a set of all-against-all alignments among the ingroup genomes, and a set of alignments from ingroup genomes to outgroup genomes.We have found outgroup selection to be absolutely crucial in creating an acceptable ancestral reconstruction: any missing data or misassembly in the outgroup that causes a true deletion in one of the ingroups to be misinterpreted as an insertion in others will mean that the ancestor contains less sequence than it ought to. This missing sequence in turn impacts the

alignment between the entire subtree below the reconstructed ancestor and the entire supertree above it: the missing sequence will never be aligned between the subtree and supertree. To avoid this we attempt to use multiple outgroup genomes in each subproblem (3 by default). Naively aligning each ingroup against multiple outgroups would significantly increase the computation time; to avoid this we note that in general any region already containing an outgroup alignment benefits very little from aligning an additional outgroup. Therefore, we iteratively align each ingroup against one outgroup at a time, pruning away any ingroup sequence already covered by the previous outgroup alignments. In this way the computational cost is reduced to be far less than naively aligning against the entire outgroup set, while still retaining nearly all of the benefit. In addition, we allow the user to designate certain genomes in the input as being of particularly high quality; these are chosen as outgroups if possible to avoid problems with missing data in regions like mitochondrial or sex chromosomes that are often missing from some assemblies but not others.

2.5.1.3 Ancestral genome reconstruction

The core of what makes the progressive alignment algorithm possible is the ancestral reconstruction generated in each subproblem. This assembly serves as a summary of each subproblem alignment; the alignable sequence between the genomes in the subtree below the ancestor, as well as that alignable between the subtree and the supertree above the ancestor, is all present in the ancestral reconstruction. The ancestral sequence contains a base for each column in all blocks which contain an alignment between two ingroups and/or an ingroup and an outgroup — any alignment purely between outgroups is discarded. The order and orientation of

the blocks relative to one another is chosen via a previously published algorithm for ordering a pangenome [90].The identity of the ancestral bases is inferred via maximum-likelihood on a Jukes-Cantor model [58] of evolution using Felsenstein's pruning algorithm [34] on the subtree of the guide tree induced by the genomes in the subproblem. These base-calls are generated as the alignment is being made, so they necessarily take only a part of the alignment information into account and may be different than the ideal base-calls would be if taking into account information across the entire alignment. However, we provide a tool, `ancestorsML`, distributed as part of the HAL toolkit [49], that re-estimates ancestral base-calls after completion of the alignment if desired.

2.5.1.4 Paralogy resolution

Previous beta versions of progressive Cactus relied on an outgroup-based heuristic to resolve duplication timing. This heuristic, which we term "single-copy outgroup filtering", separated collections of ingroup regions based on their similarity to outgroup regions, ensuring that at most one outgroup region could be present per block: the one most similar to the block's ingroup sequences. Assuming that the outgroup contains the proper number of copies and each ingroup copy is indeed most similar to an orthologous outgroup copy, this should function correctly. However, this method is very sensitive to incomplete outgroup assemblies (containing an incorrect number of copies of a duplicated region) or variation in similarity between closely related paralogs causing assignment to the wrong copy. As seen in Figure 2.3, this filtering method tended to resolve duplications far too early, often causing paralogs to be called as orthologs (for example, implicitly labeling 6.1% of human sequence as duplicated along the

chimpanzee lineage, which is certainly an overestimate).To remedy this problem, we developed an improved duplication-timing method, which we termed "best-hit filtering" in the earlier text. The method assigns, for every base in every input genome, a single *primary* pairwise alignment (the highest-scoring alignment involving that base, if it has been aligned) and a set of *secondary* pairwise alignments (all others involving that base). All primary alignments are added to the initial graph unconditionally, as they represent the most likely ortholog relationship (or in the case of multiple orthology, likely a random ortholog) (Figure A.5). The set of primary alignments represents a conservative set of alignment relationships that should include nearly no alignments to ancient paralogs. However, in regions that have undergone many rounds of lineage-specific duplications (which should all be aligned together in the restricted duplication-timing problem we describe above), the set of primary alignments will often by chance not align all copies together.For this reason, we also allow some of the secondary alignments into the initial graph, after adding the primaries, though with additional restrictions because the secondary alignments will inevitably contain some alignments to distant paralogs. We only allow in those secondary alignments that would not merge two existing blocks that both contain sequences from multiple species — this allows lineage-specific duplications to correctly land in the same block, while avoiding merging blocks from likely-paralogous alignments.

2.5.1.5 Removing recoverable chains

Due to the insensitive and approximate nature of the input local alignments, homologies are often missed in the input to the CAF algorithm. Alignment blocks that are "incomplete", i.e. contain some true homologies but miss others, can cause issues for the CAF algorithm: if these

incomplete blocks make it into the output cactus graph, the missing homologies can never be recovered by the BAR algorithm. This is because to preserve the structure of the cactus graph, BAR cannot modify existing alignment blocks, only add new ones. To remedy this issue, we developed a method to remove likely-incomplete blocks as part of the algorithm, which we term "removing recoverable chains". In short, this method runs as a post-processing step to the original CAF algorithm, removing blocks which contain only homologies that could recovered by the BAR algorithm extending from neighboring blocks. Adding this post-filtering step noticeably increases coverage, especially for distant genomes in large trees (Figure A.4). For further detail on the process, see Section A.0.8.

2.5.2 Adding a new genome to an existing alignment

There are three possible ways to add a genome into an existing alignment, depending on the desired phylogenetic position of the genome. Adding a genome as an outgroup is straightforward, since it follows the normal progressive process: the root of the existing alignment and the new genome can be aligned together into a supertree alignment, which the existing subtree alignment can be appended to. A genome can be added as a new child of an existing internal node by simply aligning the new child, its siblings, and its parent together, without inferring a new ancestral genome. Adding a genome by splitting an existing branch is the least straightforward, but is key if the topology of the alignment or the accuracy of the ancestral genomes is important. To add a genome to an existing alignment, two subproblems are required: one relating the new genome and its new sibling in the target tree, constructing the ancestral genome that will split the existing branch, and one relating this new ancestral genome, its sibling,

and its parent.After addition of a new genome as an ingroup (by adding it to a node or a branch), at most a single ancestral sequence is re-inferred. This prevents any information from the new genome from propagating to the rest of the tree. While this saves significant effort in recomputing other parts of the alignment, it also means that, occasionally, rare stretches of sequence in a newly added genome would not be properly aligned to distant outgroups because they were deleted or missing in the new genome's close relatives. Re-inferring the ancestral genomes on the path from newly added genomes to the root should address this issue if it appears.

Chapter 3

Applications of Cactus alignments

3.1 Introduction

In this chapter I briefly survey various comparative genomics projects I was involved in. All used Cactus in some form, though many of the earlier projects used a much earlier version with worse alignment quality. I present these collaborations not in chronological order, but in descending order of importance: beginning with those that had the most impact and/or where I had the most contribution, and finishing with those where I had relatively less of a contribution to the main thrust of the project. I omit many of those where my main contribution was running the alignment and possibly offering advice on further analysis.

3.2 Reconstruction of the archosaur genome

One of the first applications of the alpha version of progressive Cactus was in 2014, for the analysis of three new crocodilian assemblies (of *Alligator mississippiensis*, *Gavialis*

gangeticus, and *Crocodylus porosus*). This was done in collaboration with Phase 1 of the Bird 10K project, and published in Science as a package with those papers [45]. My contribution entailed the construction of a combined crocodilian and avian alignment as well as subsequent analysis. I ran the entirety of the alignment and downstream analysis (described below), as well as creating the tools to analyze micro-indel rates, synteny-break rates, and those to refine and evaluate the ancestral reconstruction. I generated two of the six figures in the main text; however, Dent Earl edited the figure panels and drastically improved their readability and style. Other aspects of the analysis, not downstream of the alignment, were performed by Ed Green, Ed Braun, and others, and are not described here. I briefly summarize my analyses below, but much more detail is available in the appendix (Section B).

The main result of the paper was to show that the crocodilian genome evolved at a much slower rate than their sibling archosaurs (birds). This was shown through several sources of evidence of mutation rate (ultraconserved elements (UCE), fourfold degenerate (4D) sites, gene synteny rates, and micro-indel rates). I contributed the analysis of 4D sites, gene synteny, and micro-indels, all of which relied on the alignment. All sources of evidence showed that crocodilian genomes mutated far more slowly than birds, especially in terms of genome rearrangements, even when normalizing by branch length measured in the traditional substitutions/site (Figure 3.1), which has been replicated in later work [107].

A secondary result of the paper was the detailed, base-level reconstruction of the archosaur genome (the most recent common ancestor of birds and crocodilians). The reconstruction was derived from the Cactus alignment and improved using a more accurate base-caller developed for this paper; results showing the accuracy of the base-calls as well as the preservation

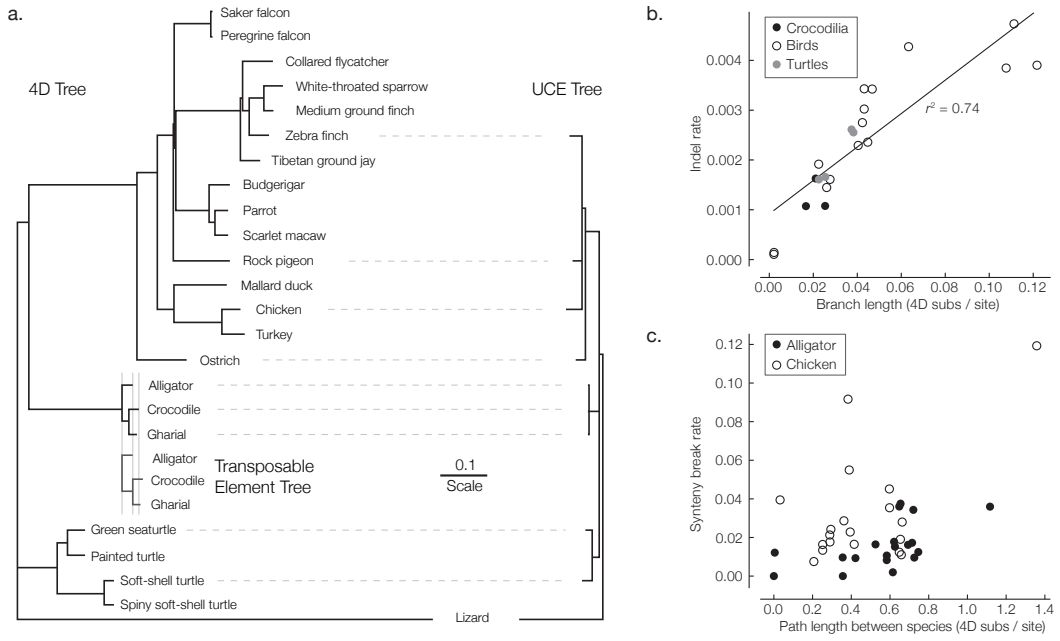


Figure 3.1: (A) Rates of substitution at 4D sites, transposable elements (TEs), and, for comparison, UCE-anchored loci. Scale bar denotes substitutions per site. (B) Indel rate versus 4D substitutions per site for each extant lineage. (C) Gene synteny breakage rate versus 4D substitutions per site, each measured with respect to either alligator or chicken.

of order-and-orientation relative to leaf genomes are shown in Figure 3.2. The improved effectiveness of recent versions of Cactus compared to 2014 is particularly obvious when considering this ancestor: while this paper described a 584 Mb assembly, newer alignments easily reach a 850 Mb reconstruction of the archosaur.

3.3 200 Mammals Project

After the progressive extensions to Cactus were proven to be highly effective for enabling clade genomics work, examining families of perhaps a dozen closely related genomes, I focused my work on much larger projects. I joined both the Bird 10K and 200 Mammals project, both of which analyze the largest-ever collection of genomes in their respective clades, and have been involved in the day-to-day analysis work of both for many years. Below, I outline the purpose of the 200 Mammals project and my contribution.

The 29 Mammals project [78], published in 2011, aimed to sequence and assemble new genomes from 20 previously unsequenced mammalian species, to bring the total number of mammalian genomes then available to 29. The main purpose of the project was to then use these 29 genomes to create a high-resolution annotation of conserved elements in the human genome. Because of the limited power available when using only 29 genomes to detect selection, these annotations were generated in 12-bp windows. The main analysis in this paper used a reference-biased MultiZ [11] alignment to detect constrained elements using SiPhy [41]. The project was able to detect 4.2% of the genome as being under negative selection [78]. Because a reference-biased alignment was used, the constrained elements were only produced for the

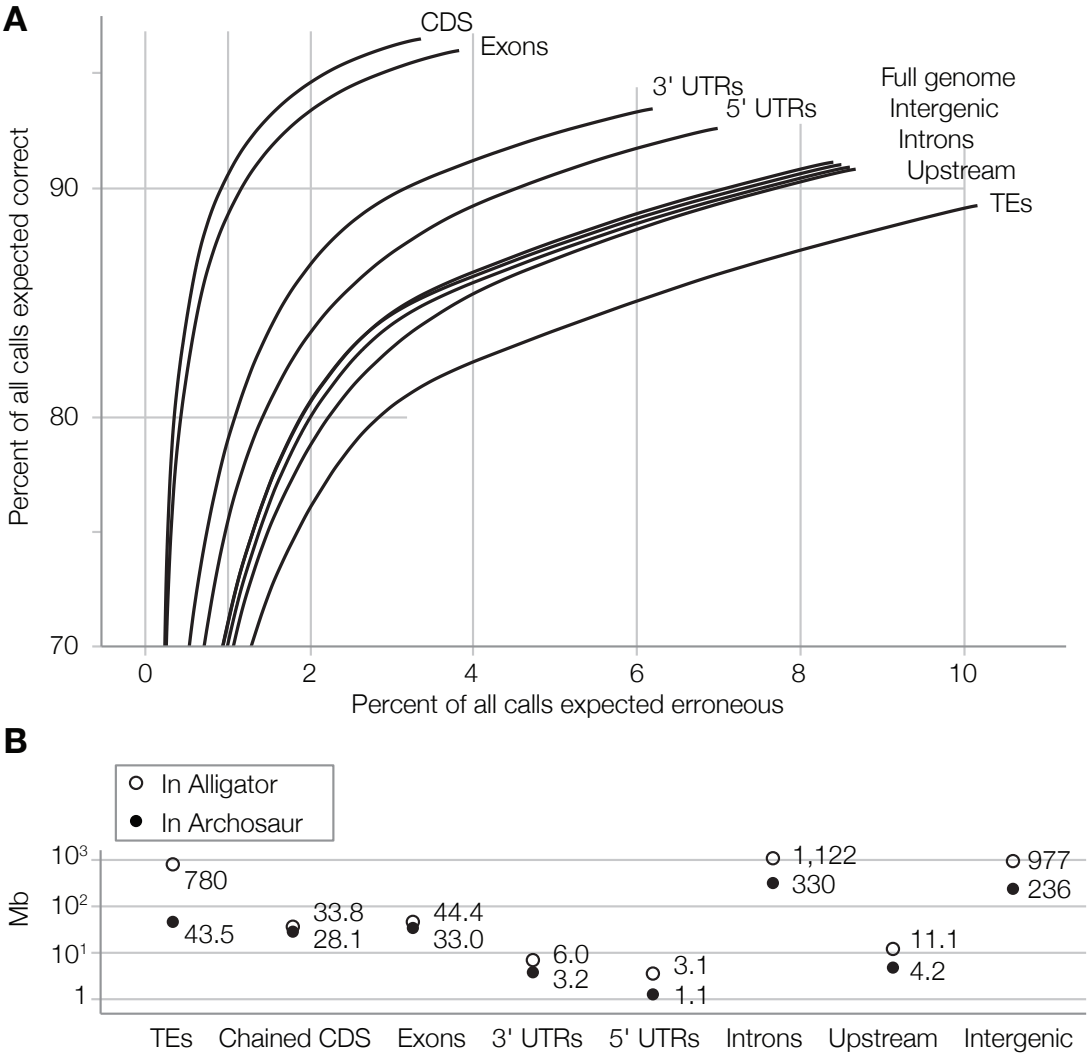


Figure 3.2: (A) Expected base reconstruction accuracy. (B) Total archosaur bases assembled in several annotated functional classes and numbers of bases in each category from the alligator genome.

human genome and not for any of the other 28 species present in the alignment.

The 200 Mammals project (200M) is a new effort to produce even-higher-resolution conserved-element annotations for the human genome as well as other eutherian mammals. Since the power to detect constrained elements is effectively proportional to the total branch length within the tree relating the aligned species [78], the way to achieve this goal is to sequence as many previously unsequenced species as possible, especially concentrating on those in relatively sparsely-sequenced portions of the tree. By sequencing and assembling 131 new placental mammal genomes, the project has brought the total number of sequenced eutherians well above 200, which should provide enough statistical power to enable 1bp-resolution constrained element annotations [78] in at least some regions of the genome. These new assemblies are all produced using DISCOVAR *de novo* [1] using a single 30X coverage Illumina HiSeq library per species, without further scaffolding. This method requires only a very small amount of DNA and is fairly inexpensive, allowing generation of over a hundred new assemblies at very low cost, but the resulting assemblies are highly fragmented, running from anywhere between 5kb and 350kb N50 (a median of 44.8 kb), with no scaffolding applied after the fact. Though this puts most of these new assemblies far below the median contiguity level for eutherian assemblies on Genbank (3.4 Mb as of June 2019), the size of the contigs does not impact the analysis of constrained elements much: as long as any given contig is long enough to be alignable and establish orthology, the fragmentation of the assembly matters surprisingly little. Though conserved sequence is less likely to be affected by reference-bias than unconstrained sequence, a reference-free alignment will still produce much better conservation annotations on non-reference species. For example, though human is a common reference with a very complete assembly, there has been a surprising

amount of conserved sequence lost even just on the human branch [84].

The vast majority of the project's analysis relies on cross-species comparison and therefore requires a genome alignment. Moreover, the project aims to also conduct analyses (such as conserved-element annotations) using multiple non-human reference genomes, so a reference-free alignment is also necessary for consistency of the alignment when viewed from different genomes. These reasons led the project to choose to use Cactus to produce the multiple alignment that will be used for the comparative aspects of the project.

The 200M project has not yet published its analysis paper(s), though I have contributed the 242-mammal Cactus alignment used for nearly all cross-species analysis as well as the neutral models used for all conservation and acceleration analysis. I also contributed to aspects of its marker paper which has been submitted to Nature and is currently under review [43]. Most importantly, I also contributed an analysis of single-base-pair conserved elements, one of the main goals of the project, which I describe below. Though the project's embargo on releasing cross-species results ahead of publication prevents me from presenting this analysis in the Cactus paper or other publications until the 200M paper is published, they have kindly allowed the results to be previewed in this thesis. The results are produced using the exact same pipeline used for the Bird 10K conservation analysis, described in Section 4.5. That analysis will be submitted to Nature soon and has similar findings, modulo obvious differences due to genome size and phylogenetic position; for this reason, I only briefly summarize the main points and key figures from the 200M work.

The 200M alignment is one of the largest genome alignments ever created, and therefore is in a unique position to supply information about selection within placental mammals.

I generated conservation and acceleration scores across all alignment columns projected onto the human genome using phyloP [101], and transformed them into single-base-pair conserved and accelerated elements at an expected false discovery rate (FDR) of 5% [10]. The resulting conserved elements were able to cover 4.3% of the human genome at a single-base-pair resolution — similar to the 4.2% covered by the 29 Mammals data, though with an order-of-magnitude more resolution provided by the (nearly) order-of-magnitude larger number of species (Figure 3.3). For a comparison of what the 200M alignment offers compared to existing best-in-class resources, I ran the same pipeline on the 100-way conservation scores offered by the UCSC browser [46]. The results for the 100-way were far more conservative (1.1% covered by 1-bp conserved elements), demonstrating the need for truly large genome alignments. As expected, functional regions were highly likely to contain conserved columns relative to the genome as a whole (Figure 3.4), though notably the proportion of non-coding regions covered by conserved elements is much larger in the 200M data than in the 100-way. This is likely due to the fact that the additional genomes from the 200M alignment increase the power to detect weak selection (Figure 3.5).

3.4 Mouse Genomes Project

I contributed to the Mouse Genomes Project, a project that sequenced and analyzed 16 laboratory mouse strains [77], producing several alignments and evaluating assembly quality as the project progressed and refined its assembly methods. The results from these alignments informed assembly decisions over the years of 2014 and 2015. I continued to produce alignments for the group even as my day-to-day involvement with the project decreased, resulting in the

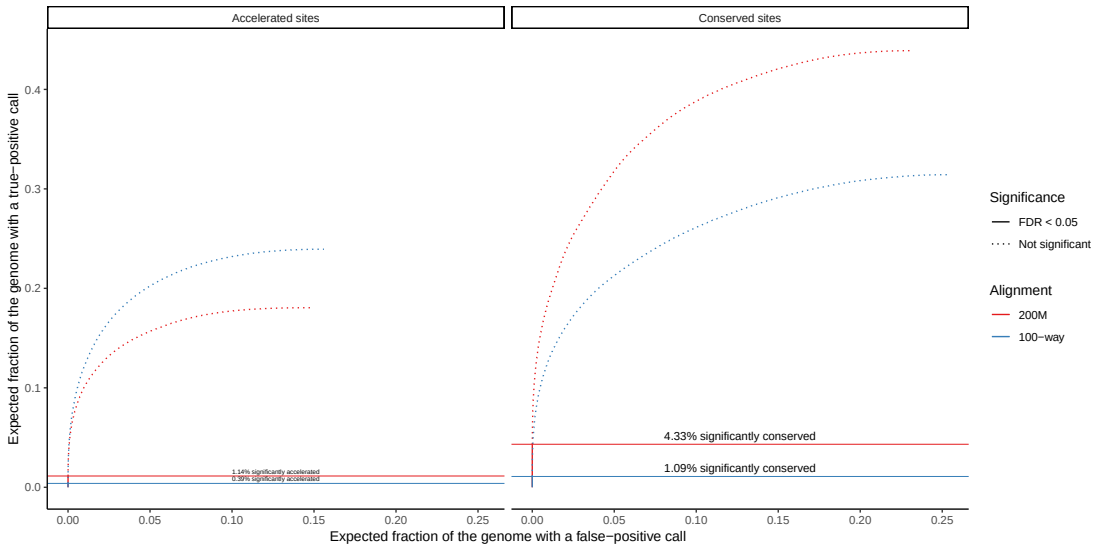


Figure 3.3: Proportion of alignment columns labeled as accelerated (left) and conserved (right). The cumulative portion of the genome with a positive or negative conserved/accelerated call is shown, starting from the column with the smallest p-value and proceeding to the columns with the highest p-values. The dotted lines show the path after hitting the FDR 0.05 p-value cutoff, and the horizontal lines show the proportion of the genome whose calls are significant with FDR 0.05.

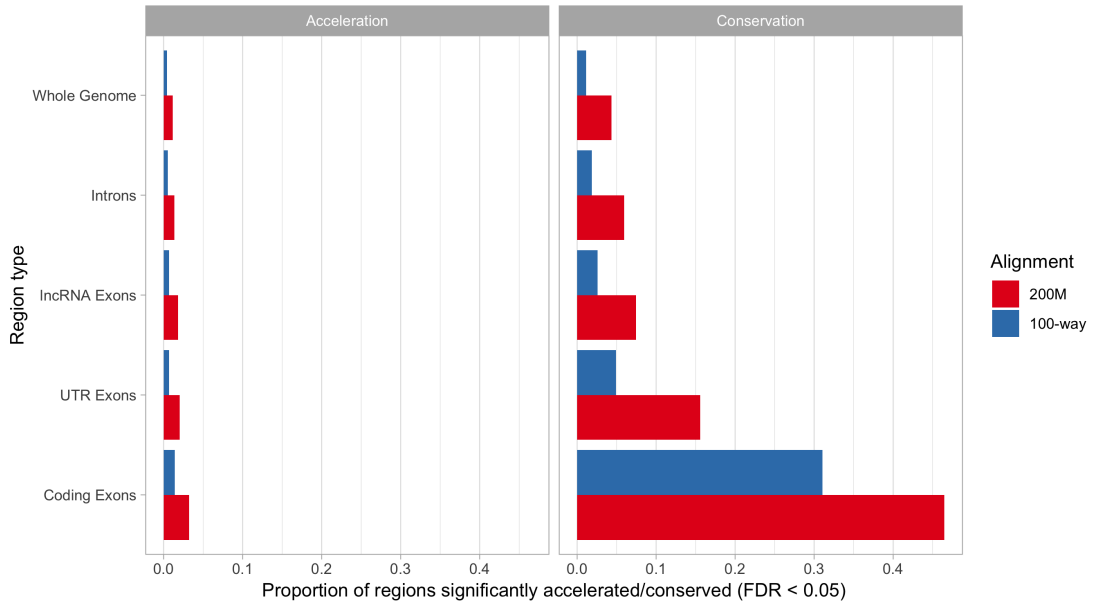


Figure 3.4: Proportion of various functional regions of the genome that are covered by single-bp conserved (right) or accelerated (left) elements in the 200M alignment compared to the public 100-way alignment.

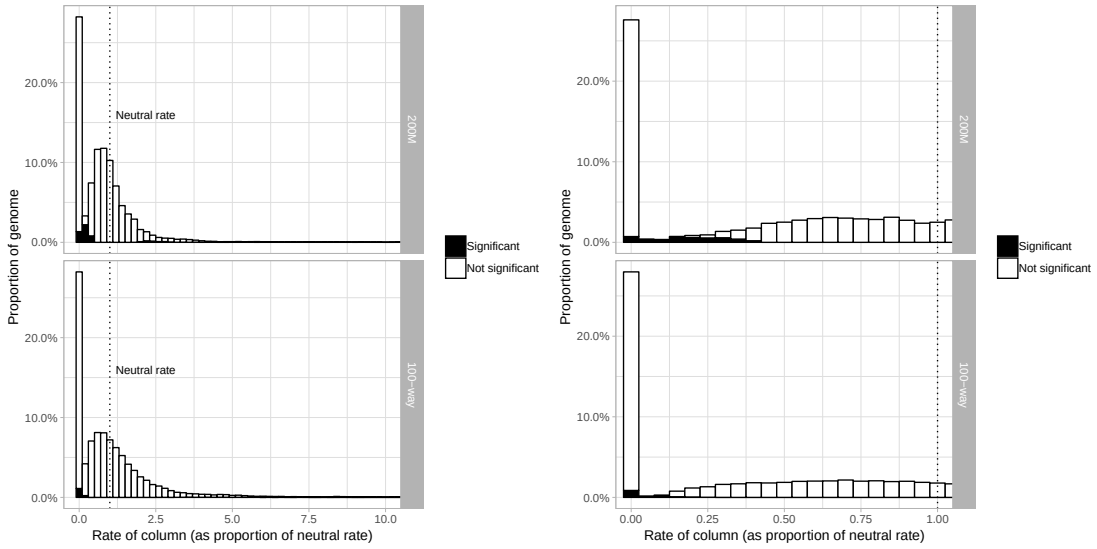


Figure 3.5: A: Histogram of rate of columns in the 200M alignment compared to the 100-way. B: The same histogram, but zoomed in to show detail in only the conserved columns.

final alignment described in the manuscript. I also created (with Ian Fiddes) the assembly hub produced as a result of the project and available on the UCSC Genome Browser [46].

3.5 Shasta / T2T

Shasta is a new method for assembling genomes using nanopore data. The manuscript describing Shasta is currently in preparation. I contributed to this project by aligning and annotating (using CAT [37]) dozens of human assemblies from Shasta and competing assemblers. These results indicate Shasta, especially post-polishing, is able to represent genic regions at an accuracy better than, or at least competitive with, other long-read assemblers that take an order of magnitude more runtime. My results also guided selection of parameters for MarginPolish, a

polishing method released alongside Shasta.

I also aligned and annotated several iterations of a new CHM13 assembly being developed by the Telomere to Telomere (T2T) consortium with the goal of creating at least one assembly with end-to-end representation of every base in the human genome, even in centromeric, telomeric, or otherwise highly repetitive regions. I contributed comparison to other CHM13 assemblies that will appear in the manuscript (in preparation).

Chapter 4

Densely sampling genomes across the diversity of birds increases power of comparative genomics analyses

4.1 Preamble

The Bird 10K Project (B10K) [122] is an ambitious sequencing and analysis project aiming to assemble all bird species within the next few years. The project is proceeding in four major phases, each aimed at filling in a different one of the four major taxonomic ranks within birds: one phase for sequencing at least one species within each *order* of birds, then one phase for sequencing at least one within each *family*, then finally a *genus* phase and *species* phase. Each phase will involve a roughly order-of-magnitude increase in scale from the previous, putting B10K at the forefront of large-scale comparative genomics. This massive scale will offer unprecedented insights into avian genome evolution.

The first, ordinal phase sequenced 45 previously unsequenced avian species, bringing the total number of avian assemblies available to 48. The group analyzed the resulting 48 avian genomes and was massively successful, resulting in the release of dozens of papers revealing details of avian evolution, including improving our understanding of the highly-controversial avian species tree [55, 123]. The second, family phase is currently in progress. 237 species were newly sequenced for this phase of the project, bringing the total number of species with genomes available for analysis to 363 after including various assemblies from Genbank [65] or those contributed from collaborators.

The remainder of this chapter is the text for the release of the assembly, annotation, and alignment for the current phase of the Bird 10K project, which will be submitted to Nature in the coming weeks. I am a co-first author on the paper along with Josefin Stiller, Yuan Deng, and Shaohong Feng, and performed the entirety of the alignment and conservation analyses, contributed the sections describing those analyses, as well as made significant edits to the rest of the text of the paper. I have also further edited the current text of the paper to better fit into this thesis.

4.2 Introduction

In an era of phylogenomics where large whole-genome sequencing projects are increasingly populating the tree of life and characterizing genomic biodiversity [73, 66, 32], there is high demand for establishing efficient comparative genomics workflows to deal with these enormous and often heterogeneous datasets. Recent comparative phylogenomic efforts with

relatively small-scale taxon sampling have demonstrated the immense biological insights that can be obtained from comparative analysis of genomes [123, 55, 20, 78, 59]. However, sparse sampling of a few species may confound phylogenetic inference [103] and can necessarily only capture a fraction of the genomic diversity. Here, we report a significant step towards denser representation of avian phylogenetic diversity by analyzing a total of 363 genomes from 92% of bird families, including 268 newly sequenced genomes produced by family phase program of the Bird 10,000 Genomes Project (B10K), making it the largest multi-species vertebrate genome dataset to date. We show that a novel pipeline leveraging a reference-free whole-genome aligner identifies orthologous regions in greater numbers and more consistently across species than previously possible. The alignment also allows us to recognise genomic novelties specific to particular bird lineages. This unprecedentedly dense phylogenomic sampling significantly enhances the power to detect evolutionarily constrained positions down to individual base pairs, resulting in a more-than-doubled estimate of the proportion of the avian genome conserved. Our results demonstrate that increasing the diversity of genomes in comparative analyses can unveil more shared and lineage-specific variation and can quantitatively improve the dissection of genomic characteristics. In addition to the phylogenetic and comparative analyses of this dataset underway by the B10K, we anticipate that these genomic resources will assist species conservation and offer new perspectives on evolutionary processes in cross-species comparative analyses.

4.3 Genome release

For Phase II of the B10K (the "family phase"), we included a total of 363 species representing 93% (218 of 236) of avian families. Samples were selected to broadly cover the overall diversity of Aves and to subdivide the long branches (Figure 4.1). The current sampling more than triples taxonomic span from 63 to 218 bird families, of which 155 families were sequenced for the first time and 75 families have genomes available from multiple species. Of the 18 missing families, eight were not represented because of poor-quality assemblies, while 10 were lacking appropriate samples. We chose a short read sequencing strategy, which has the important advantage of being applicable to most of museum specimens. This strategy allowed us to sequence a broad variety of species, including old samples (the oldest collected in 1982), samples from all continents, and museum gems such as the Henderson Crake (*Zapornia atra*), that occurs on a single Pacific island and of which we sequenced tissue of one of the few vouchered specimens. We include 68 species that are listed in some category of concern by the IUCN RedList of Threatened Species, including two Critically Endangered birds, the Plains-wanderer (*Pedionomus torquatus*) and the Bali Myna (*Leucopsar rothschildi*), the latter with less than 50 adults remaining in the wild.

A total of 268 genomes are newly released with 18.9 trillion base pairs (bp) of raw data and 291 billion bp of assemblies for immediate use by the community. Of those, 236 were species specifically chosen for the B10K (after filtering from initially 272 species). A total of 49 genomes were contributed to the B10K by individual research groups, of which 17 have already been made available and the remaining 32 are newly released here. Together with 78 publicly

available genomes, a final dataset of 363 species was constructed. These genomes have been generated with a variety of methods (454/Sanger, Illumina, PacBio) and therefore show a range of assembly contiguity. Most new genomes were sequenced to 35x-374x coverage, with 11% having over 100x coverage and assembly qualities (average scaffold N50 = 2.25 Mb, contig N50 = 45.32 kb) are comparable to previously published bird genomes . Genomic completeness assessed by BUSCO [118] was high (average 95%), which shows that most genomes are well suitable for comparative analysis.

Protein gene models across the 363 avian genomes were predicted using a homology-based method with a uniform reference gene set including gene models from the chicken, zebra finch, human and published transcriptome sequencing for some birds. This approach predicted on average 15,464 protein coding genes, ranging from 9,909 in *Neodrepanis coruscans* to 19,174 in zebra finch. The mitochondrial genomes of 321 of 363 species were de novo assembled [25], with 210 samples (65.42%) fully circularized and annotated [86], and 201 species (62.61%) had the complete complement of 37 genes.

We constructed a whole-genome alignment of all 363 genomes using an updated version of the reference-free aligner Cactus [95]. Unlike many other whole-genome aligners, Cactus is in principle equally useful for all genomes, rather than being biased toward sequence present in the reference genome. This gives Cactus two advantages: it can identify lineage-specific insertions and deletions and it reconstructs ancestral sequences. It can also produce more complete genome alignments, which were normally broken by the repeat elements in other alignment methods. For instance, Cactus aligned 981 Mb (93.7%) of the chicken genome and 1.17 Gb (94.8%) of the zebra finch genome to at least one other species. The proportion is

much greater for functional sequence: e.g. for chicken genes identified by BUSCO, 97.5% had an alignment to turkey covering the majority of their bases (92.5% to ostrich). Compared to a commonly used reference-based method MULTIZ (Blanchette et al. 2004), Cactus aligned a similar proportion of coding regions (24.5 vs. 24.6 Mb of MULTIZ) but aligns 3% more of intronic regions (275.3 vs. 267.0 Mb) and 4% more of intergenic regions (700.0 Mb vs. 670.8 Mb).

4.4 Increased power to detect orthologs using a whole-genome alignment

High quality homology is of crucial importance for comparative studies, be it for uncovering phylogenetic relationships or for studying the genomic landscape of evolutionary change [114]. Orthology is straightforward when there is a one-to-one relationship between the genes of compared species that can be shown by reciprocal best hits (RBH). In Phase I of the project, orthologs were identified with a method based on RBH [123]. However, duplication events can lead to more complex patterns of one-to-many or many-to-many orthologs [68]. In such cases, sequence similarity may not distinguish the original and derived copies [91]. Copies are however often embedded into different genomic contexts, which makes positional information valuable to obtain the pairwise relationship of one-to-many/many-to-many orthologs [47, 40, 24]. Based on these ideas, we have developed a new pipeline to identify orthologs that first locates potential homologous regions on the Cactus whole-genome alignment and then refines ortholog relationships by incorporating conserved gene synteny and sequence similarity. We identified

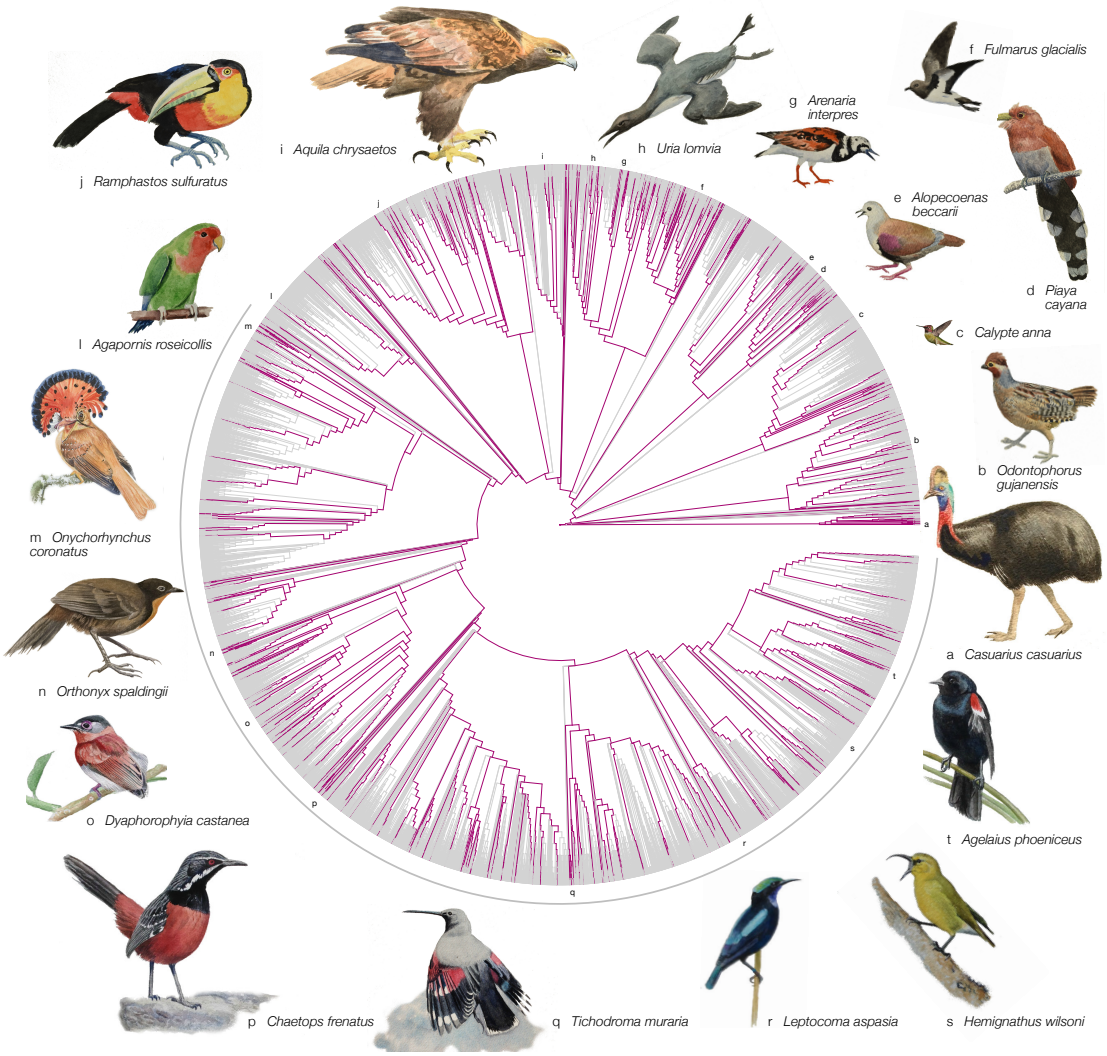


Figure 4.1: Shown are 10,135 bird species on the mega-phylogeny that synthesizes taxonomic and phylogenetic information [13], of which 363 species (purple highlights) now have reference genomes available. This corresponds to 92% of all bird families now having genomes available. Drawings illustrate select examples of species with available genomes. The grey half circle indicates the ultra-diverse Passeriformes with 6,063 species.

16,179 orthologs, of which 15,807 remained after removing the orthologs that were absent in 99% of the species (>359 species). The Cactus-based method consistently identified on average 8% more orthologs (n=1066) than the Phase I method in each species (Figure 4.2), and identified genes more consistently across all species (occupancy: 82% vs. 78%). When setting all other conditions equal, we also found that sampling more birds together with the Cactus-based method increased the number of detected orthologous proteins by 12% compared to the earlier 48 birds ortholog set [123]. Based on the family-phase orthologs, we obtained 129.33 Mb conserved orthologous introns in total, $\tilde{1}2\%$ of an average avian genome, which was 6.7 times higher than the intronic regions identified in the earlier 48 birds.

The use of the Cactus alignment not only ensures the identification of the classical one-to-one orthologs, but also improves the power to distinguish cases of co-orthologs (Figure 4.3). As transcriptional factors, FoxP subfamily proteins have been proved to be the key elements for vocal learning in birds [85]. Thus, distinguishing their co-orthology relationship could help us to further analyze their evolutionary roles in the genetics of language, especially the function of FoxP3 in passerines. Another advantage of Cactus alignments is that we could obtain the ancestral sequences at each node in the phylogeny, which enables the detection of both shared ancestry and novel specific sequences of any lineage. Thus, we introduced a pipeline to obtain the avian pan-genome, a collection of the shared and lineage-specific diversity at the root of a set of bird species. We executed the pipeline on Passeriformes (173 genomes) and identified 5,958 Passeriformes-specific genes, which have orthologs only within Passeriformes (though losses are possible) and are inferred in the reconstructed ancestral "genome" of the Most Recent Common Ancestor (MRCA) of Passeriformes. Some Passeriformes-specific genes were retained in a large

number of species. Among the top, we found by Swissport annotation two feather keratin related genes (retained in 100 and 104 of 173 Passeriformes, respectively) and three nuclear encoded mitochondrial genes (retained in 131, 115 and 105 of 173 Passeriformes, respectively). Gene DNAJC15, one of the top candidates, has many copies in bird genomes and is thought to be associated with the biogenesis of mitochondria [109] and fertilization as a member of heat shock proteins [121]. With the Cactus alignment, we first located a unique copy in Passeriformes lineage at the position shown in Figure 4.4.

4.5 Single-base-pair resolution annotations of purifying selection

The diversity of species represented in the 363-species Cactus alignment also gives much more statistical power to detect weak conservation. A slower rate of mutation than expected in a given region between a set of species is often an indicator of purifying selection [27]. For this reason, annotations of conserved elements are useful for investigating function within the genome [84]. We created conservation scores using phyloP [101] for each basepair of the 363-species Cactus alignment on the chicken genome. We compared these results against phyloP scores we derived from two similar alignments: a 77-way alignment including avians as well as other vertebrate outgroups (the largest publicly available avian alignment, obtained from the UCSC Genome Browser [46]) and a 53-way alignment containing only avians (a subset of the 77-way). We tailored our scoring method to account for a key factor in avian genome evolution: the difference between the rates of evolution in micro-, macro-, and sex chromosomes. We scaled our model of the neutral rate of mutation (which is used to evaluate the degree of the

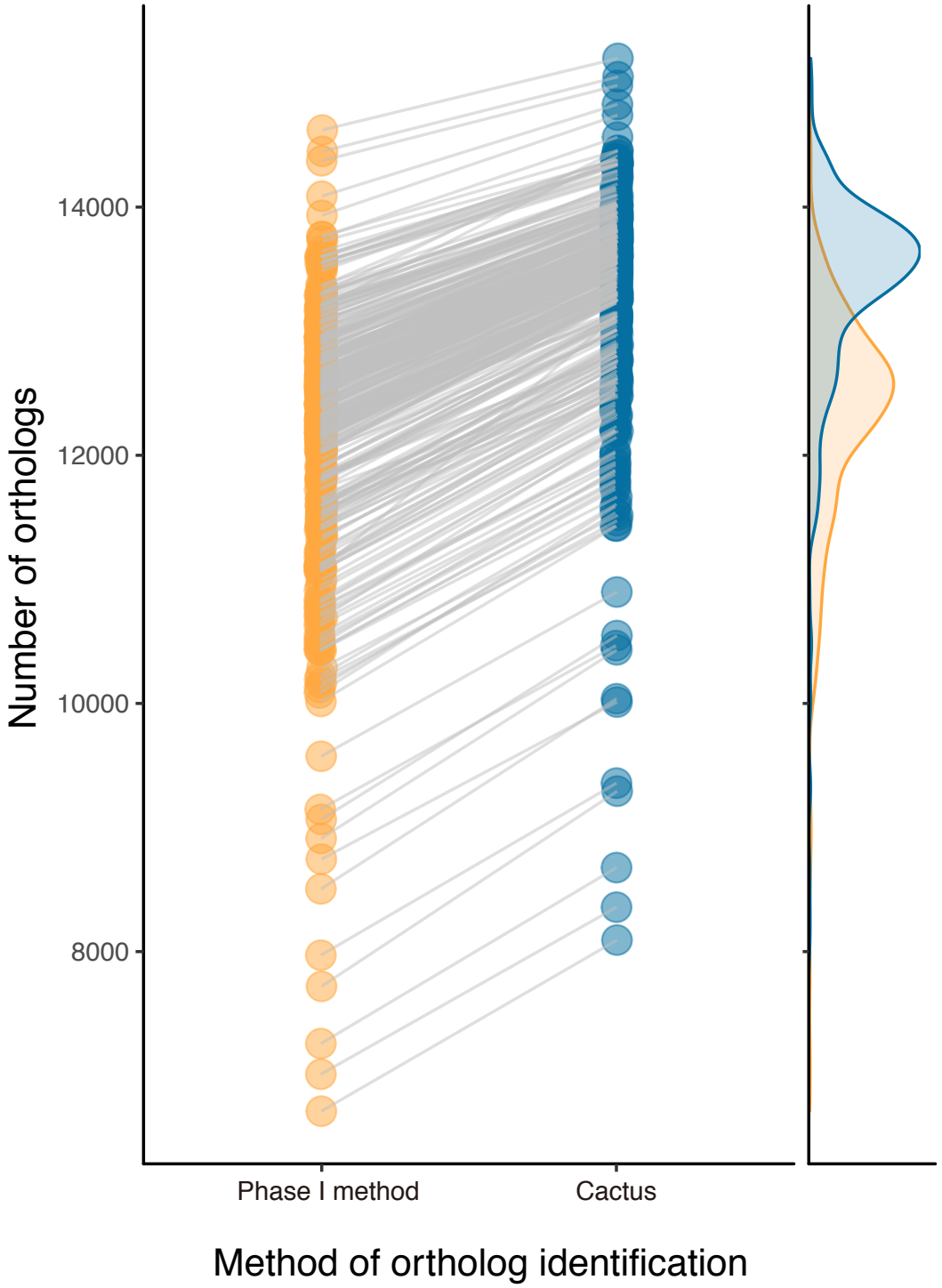


Figure 4.2: The novel ortholog pipeline that uses the Cactus whole-genome alignments identifies more orthologs than the conventional RBH pipeline. The lines connect the annotations of the same species with the two methods.

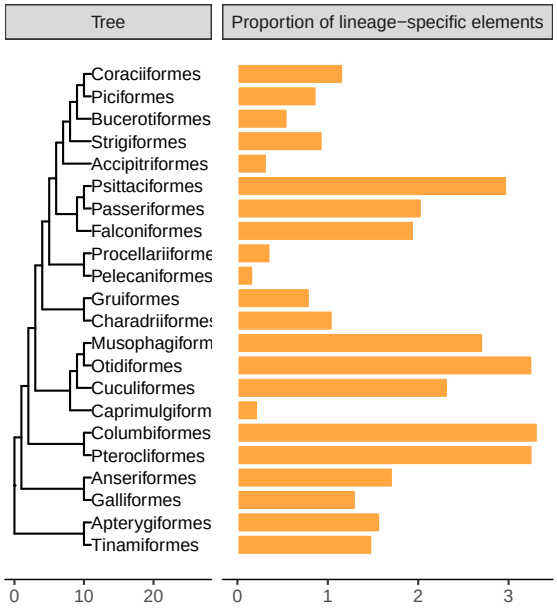


Figure 4.3: The one-to-one ortholog relationship of three co-orthologs genes FOXP1, FOXP2 and FOXP4 in chicken and zebra finch. The first two genes in chicken can be assigned to their respective corresponding genes in zebra finch through the ortholog pipeline used in phase 1 and this phase, but the last gene can only be identified correctly in the Cactus-based method.

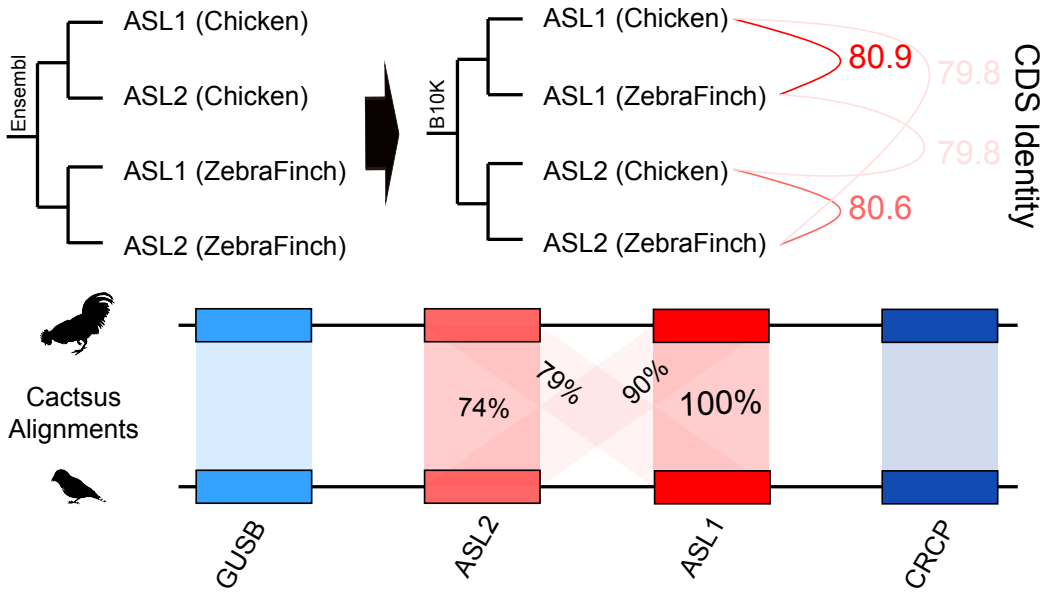


Figure 4.4: Proportion of lineage-specific regions in bird orders. Lineage-specific sequences are only found in members of the clade and are reconstructed in the ancestral genome. Values shown for clades with more than 2 representatives.

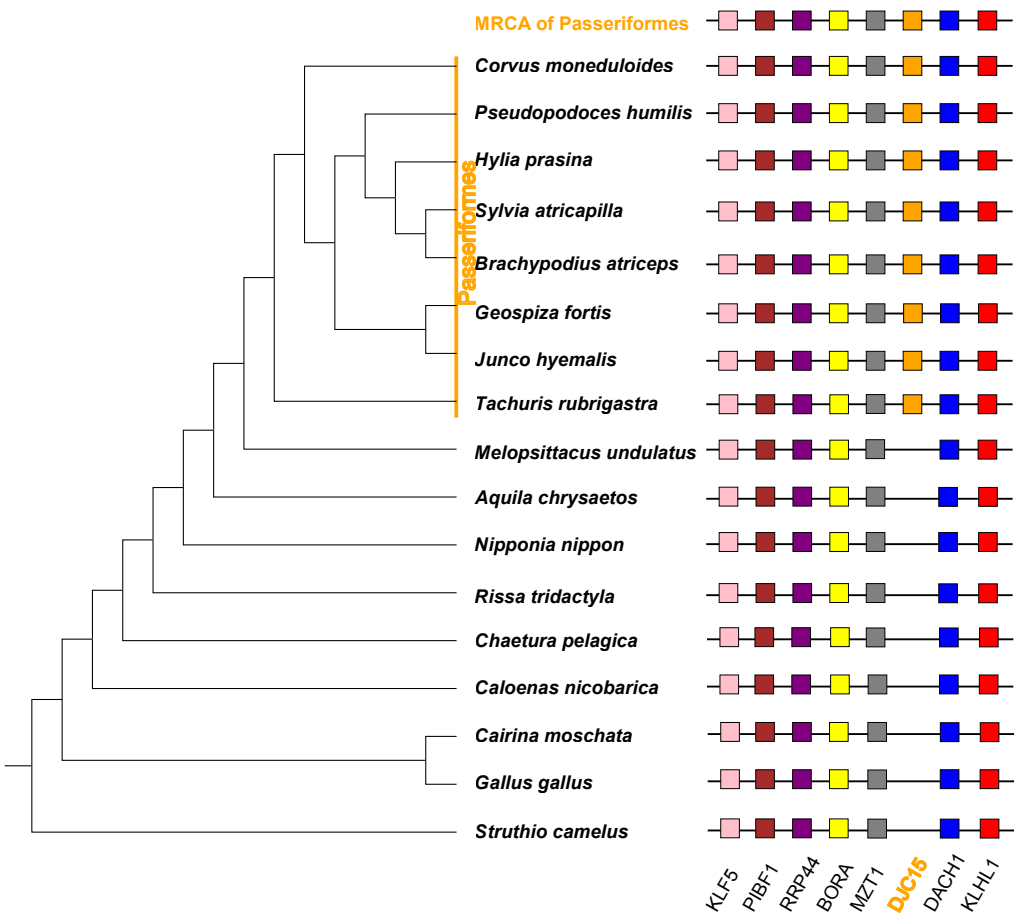


Figure 4.5: Chromosomal organization of a putative lineage-specific gene (DNAJC15) and its surrounding genes in Passeriformes and non-Passeriformes. DNAJC15 can be found in 131 of 174 passerine species can be located in the reconstructed "genome" of the MRCA of Passeriformes, but cannot be found in any non-Passeriformes. This gene is therefore likely a Passeriformes-specific gene.

departure from the neutral, unconstrained rate for each site) to match the neutral rate observed in each of micro-, macro-, and sex chromosomes, and used these three models when generating the conservation scores. We found that the neutral rate within sex chromosomes is 16% faster than in macrochromosomes, and that the neutral rate within macrochromosomes is 9% faster than in microchromosomes, consistent with the fast-Z [81] and fast-macro [7] hypotheses as well as our earlier findings [123].

Though our previous comparison for 48 bird genomes detected that at least 7.5% of the chicken genome was conserved [123] at 10bp resolution, this ratio was reached by integrating across multiple adjacent bases, trading off a low resolution for a necessary increase in statistical power. The statistical power to detect conserved elements is roughly proportional to the total branch length within the tree relating aligned species [21]. Our dense sampling of avians results in a total branch length of 16.5 expected substitutions per site, compared to 9.9 within the 77-way and 4.3 within the 53-way. This increase in branch length causes an enormous increase in our ability to detect negative selection, rendering for the first time a site-by-site conserved element annotation that covers a substantial portion of the genome. We transformed the phyloP scores described above into calls of significantly conserved single-base-pair elements at an expected FDR [10] of 5%. Our alignment provides ample increases in the number of bases detectable as conserved at single-base-pair resolution relative to the browser alignments that contain fewer taxa (13.2% of the chicken genome in the 363-way vs. 3.8% in the 77-way and 2.1% in the 53-way Figure 4.6]). While the additional branch length afforded by the vertebrate outgroups in the 77-way offered the ability to detect a larger number of constrained sites than the 53-way, it still fell short of creating a high-coverage 1-bp resolution annotation, falling short of the 7.5%

that we had estimated using 10-bp elements. In contrast, the denser phylogenetic sampling allows us to not only find constrained elements covering a greater proportion of the genome than our earlier work, but also at the highest possible resolution.

These results give insight into weakly conserved functional regions of the genome. Though our alignment is able to detect 62.4% of bases within coding exons as conserved, higher than the 34.3% within the 77-way and 18.6% within the 53-way, the increase is proportionally much larger in non-coding regions of the genome such as lncRNAs obtained from NONCODE [33] (16.2% vs. 4.8% and 3.2%) and untranslated exons (30.1% vs. 8.8% and 6.0%) (Figure 4.7). These increases in the proportion of the genome under selection detectable at a single-base level come largely from an increase in ability to detect weakly conserved sites. This indicates that while non-coding regions are less strongly conserved than coding regions, much of their sequence is still under some amount of constraint. The fastest-evolving columns detectable at this level of significance evolved at 52% of the neutral rate for the 363-way alignment, compared to 26% for the smaller 77-way alignment (Figure 4.8). The 53-way alignment provided only enough power to detect conserved bases that were completely unmutated within all avians (0% of the neutral rate).

4.6 Discussion

This dataset makes birds now a system with unparalleled genomic resources. The genomes will serve the community both individually, to investigate species-specific traits and to support conservation efforts of the sequenced and their relatives, but also collectively in

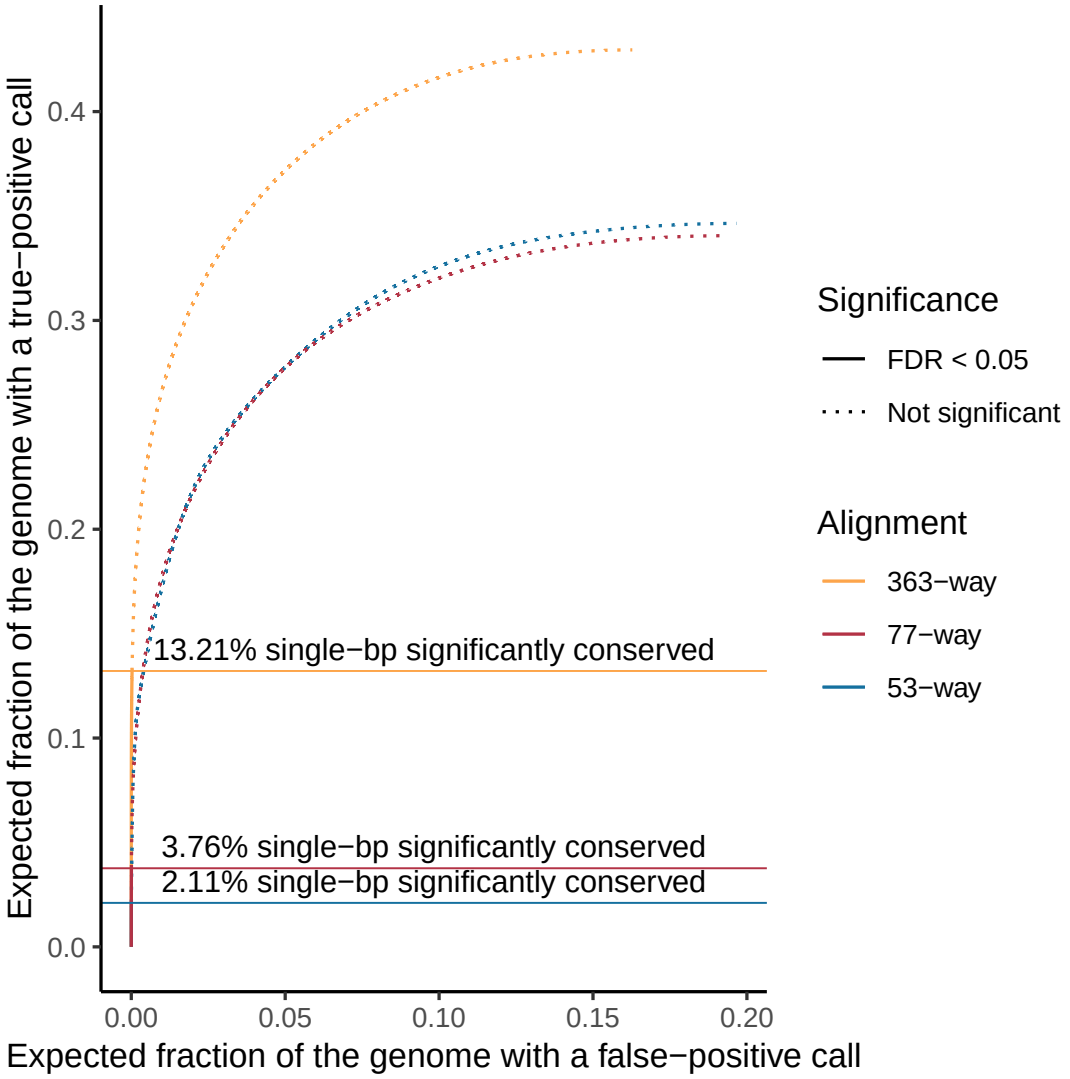


Figure 4.6: Proportion of alignment columns labeled as conserved. The cumulative portion of the genome with a conserved call is shown, starting from the column with the smallest p-value and proceeding to the columns with the highest p-values. The dotted lines show the path after hitting the FDR 0.05 p-value cutoff, and the horizontal lines show the proportion of the genome whose calls are significant with expected FDR 0.05.

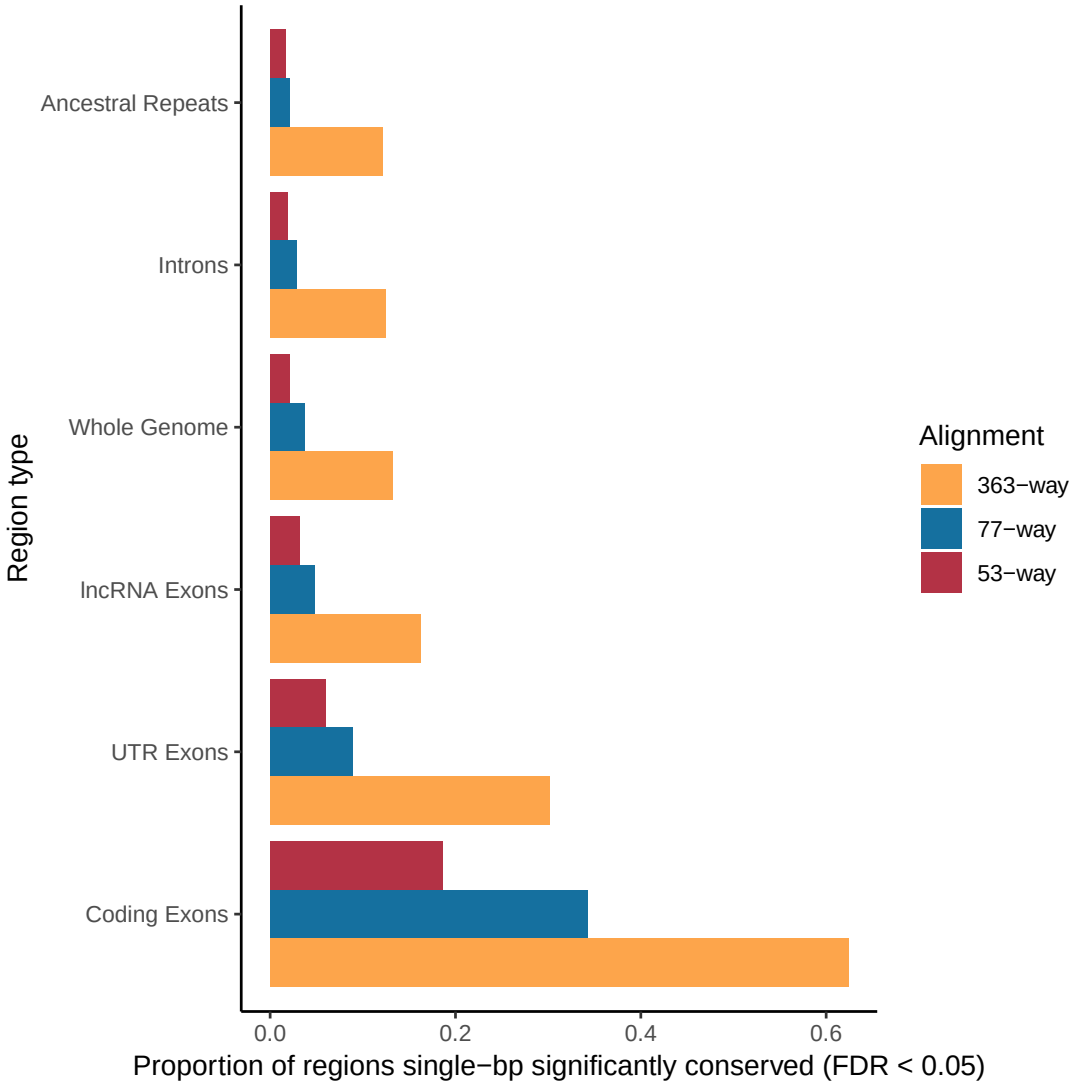


Figure 4.7: Proportion of various functional regions of the genome that are covered by single-bp conserved elements in the B10K alignment compared to the existing 77-way resource.

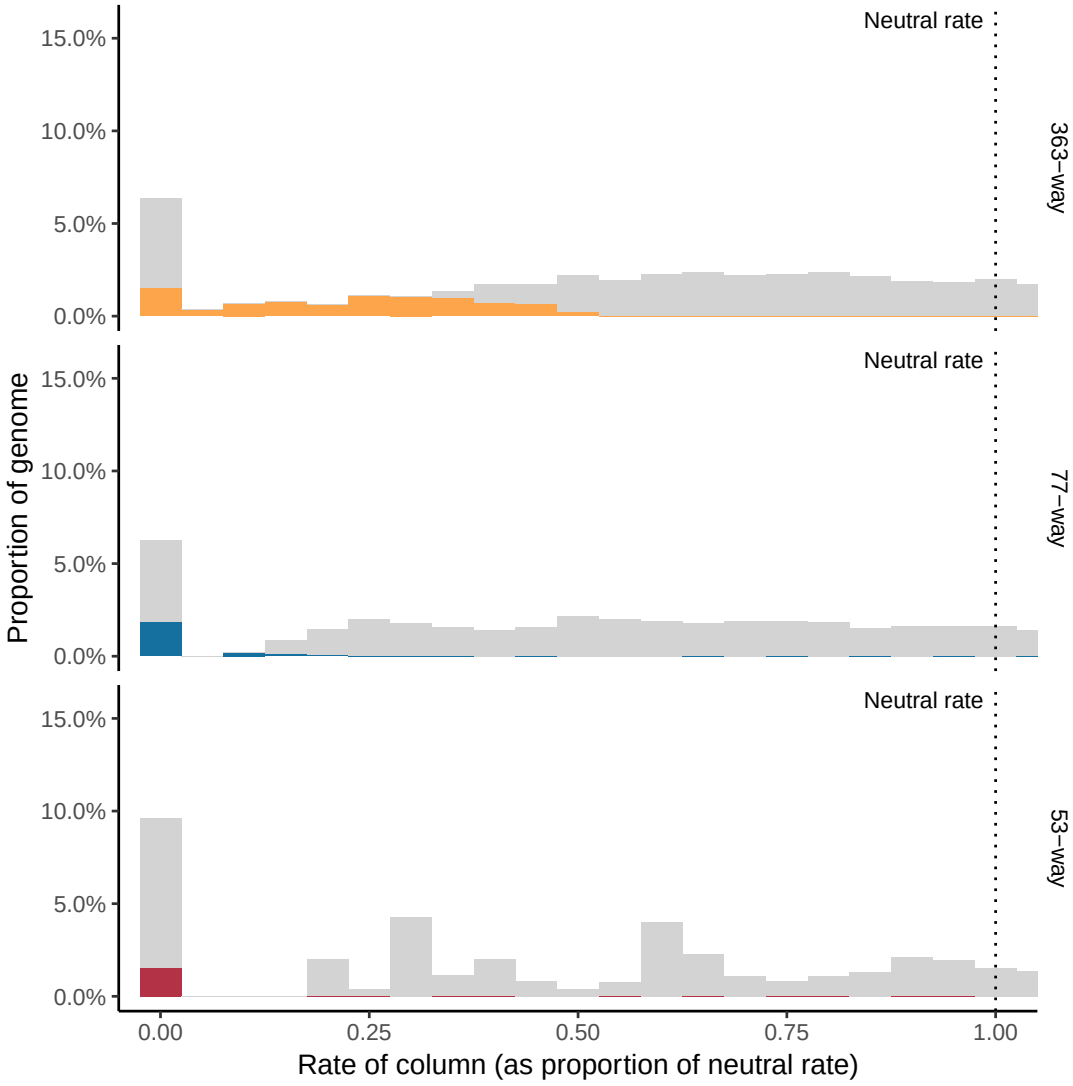


Figure 4.8: Histogram of the rate of alignment columns relative to the neutral rate (here labeled 1.0), focused only on columns evolving slower than neutral. A rate of 0.0 indicates no mutations observed at all in the columns. Note that the 0.0 column has a relatively high proportion of non-significantly conserved columns because of recent insertions, which will often have no observed changes but offer very limited statistical power due to being present in only a few species.

cross-species comparisons to gain new perspectives on evolutionary processes and genomic diversity. The B10K consortium's goal is not just to merely sequence but to explore the full information content contained in these genomes, for which we welcome new collaborations. The B10K consortium is already analyzing this dataset pursuing its main goals:

1. **Build the new timetree of birds.** We are investigating how scaling up both sequence length and the number of terminals impacts phylogenetic resolution of the notoriously difficult to resolve early neoavian divergences, as well as more shallow branches. Dense sampling breaks large phylogenetic distances between taxa and provides added anchors for fossil calibrations, which will allow for new insights into the evolutionary history of birds. Accessing more genomic sequence of different functional categories (coding, non-coding, ultraconserved elements) will allow us to better understand the distribution of phylogenetic conflict across the genome [105].

2. **Investigate the genomic underpinnings of bird innovations in morphology, physiology, and life history.** We have built a comprehensive, expert-curated trait database that is used to link genes and regulatory regions and lineage-specific genomic innovations to phenotypes in cross-species comparative analyses. We will investigate evolutionary constraint in birds with unparalleled power to identify highly conserved regions and those under accelerated evolution.

3. **Genetic diversity and demography.** We are investigating historical trajectories of population sizes across all birds during the Pleistocene era to describe their N_e dynamics, model their strategies to adapt to previous climate change, infer the ecomorphological factors that

influence demography, and predict future diversity trends under current climate change.

Chapter 5

Discussion

This thesis has described an effort to scale comparative genomics up to hundreds of genomes. The results from this work have demonstrated both the effectiveness of the approach and the potential unlocked by such large-scale analysis. The progressive extensions to Cactus I presented have made reference-free genome alignments at the scale of hundreds to thousands of species possible. Cactus has been thoroughly tested and benefited from incremental scale-ups, tested across a range of successful comparative genomics projects, ranging from alignments of small clades of $\tilde{5}$ genomes and finishing with the largest genome alignments yet created. Along the way these alignments me to do enabled significant comparative genomics work to take place, including analyzing the rate of evolution of the archosaur genome and generating human and avian conserved element annotations at unprecedented resolution and sensitivity. Moreover, the influence of this data will continue: dozens of research groups worldwide will use the Bird 10K and 200 Mammals alignments for further analysis in the next year.

However, future work will still be necessary to continue bringing comparative genomics

forward. Third-generation sequencing and assembly technologies [31, 53, 54, 119] are already bringing new, more accurate genome assemblies at an impressive speed; in my opinion, a major challenge facing comparative genomics is to scale up to actually make use of these genomes in cross-species analysis. While the work presented in this thesis demonstrates that it is possible to do comparative genomics at large scale, there is more to be done in order to truly catch up to the expected rate of incoming assemblies.

First, on the alignment side, Cactus is currently fairly computationally demanding and therefore costly (120 CPU-days per assembly and around $100–200 per assembly on the AWS cloud). Therefore, though large alignments are possible, they are significant undertakings, requiring at the least a large cluster, and in practice benefits from an autoscaling cloud environment. The computational effort is largely expended in the local alignment stage of Cactus, which uses the LASTZ aligner [48] and scales quadratically with genome size. The local alignment effort could, however, be reduced for closely related genomes such as in same-species comparisons, where divergence is so low that MinHash-based approaches are sensitive enough to be practical replacements for BLAST[4]-esque local aligners [74, 52].

Further, though creating alignments of hundreds of species is possible, even using large genome alignments can carry its own challenges. As one aspect, with increasing numbers of species per alignment column, the chance that at least one entry contains an assembly or alignment error increases. For example, many of the Bird 10K assemblies were assembled using a version of SOAPdenovo [76] that produces many false tandem duplications; as a result of this and the sheer number of affected assemblies, the chance that any given column happens to hit is involved in a duplication is high. Conventional single-copy-filtering of the Bird 10K therefore

removes far more columns than in smaller alignments. Furthermore, remote access of large alignment files for visualization purposes remains challenging. I recently completed work (along with Mark Diekhans) to develop a new file format which improves the speed of accessing our reference-free indexed alignment format, HAL [49], by up to an order of magnitude. However, remote-access alignment formats like HAL or bigMAF are still quite slow due to the large number of network round-trips required when seeking within the file, even despite the indexed nature of the files. One possible solution is to, rather than host a web server that merely knows how to serve parts of files, host a "smart" server process that can parse an RPC call requesting to view a subset of the alignment, do the required seeking and gathering of the alignment data on local disk (which will be much lower-latency than across a network link), then return the response to the RPC call over the network. Early tests suggest that such a strategy would reduce runtimes for remote-access HAL queries from several minutes to mere seconds.

The advances I describe also prompt questions about the desired semantics of genome alignment. For example, it is usually axiomatic that a genome aligner's goal should (at least by default) to be to align orthologous regions to one another [6]. Cactus takes it for granted that it should align *all* orthologous regions to one another. However, across a great deal of evolutionary distance, that can cause subtle problems for many use-cases. For example, even though a new lineage-specific pseudogene in chimp would indeed be orthologous to its source gene in human, aligning the human copy to *both* the orthologous gene and orthologous pseudogene in chimp confuses and frustrates users who are not interested in pseudogenes. Most users would prefer aligning to the clear *parent* of the duplication, if possible, rather than one of the *daughter* copies. Applying the concept of *positional orthology* [24] (sometimes called *toporthology*) to genome

alignment could improve this situation. Integrating positional orthology into genome alignment would enable separate investigation of a single orthologous locus as well as the existing behavior of aligning all conventional orthologs.

The single-bp conservation analysis I presented in Chapter 4 demonstrates the utility of large genome alignments for examining selection. The methods (e.g. phyloP [101]) I used in examining conservation have remained largely unchanged for years, because they do their job well. However, I believe that large genome alignments, spanning many clades, offer opportunities to push for new methods to detect selection. Most importantly, across alignments that span many hundreds of millions of years of genome evolution, a large part of conserved regions will be conserved in only part of the tree. Furthermore, their degree of conservation may differ, resulting in slightly different rates over time (heterotachy [79]). Detecting lineage-specific selection automatically across the entire tree is currently somewhat difficult: the current suggested method is to look for lineage-specific selection along a single branch at a time [101]. Though an HMM-based method for detecting lineage-specific selection along multiple branches has been developed, called DLESS [111], the model only allows for a single gain or loss event across the entire tree, and a single "conserved" and "non-conserved" rate. Further pushing the boundaries and creating methods to simultaneously estimate variation in mutation rate both among and between sites (possibly using a model similar to the "covarion" model proposed for codon evolution [38]) should prove very useful in the coming era as datasets of thousands or tens of thousands of genomes become commonplace.

The work I have presented in this thesis, particularly the progressive extension to Cactus, has pushed forward the state of the art of comparative genomics, and will continue to

play a role in the field. My work with large genomics projects like Genome 10K, 200 Mammals, Bird 10K, and the Vertebrate Genomes Project has convinced me that large alignment resources pay dividends. Some parts of the 600-way alignment I describe in this thesis will undoubtedly be used in other projects, though unfortunately some aspects remain under embargo. What is clear is that a public vertebrate alignment resource (iteratively updated using Cactus as new assemblies are made public) is sorely needed. The 600-way I described could be thought of as a first draft of that effort. Results from simulated data in the Cactus paper have shown that maintaining an alignment by iteratively updating it as new assemblies arrive is relatively inexpensive while offering accuracy nearly as high as a full realignment would. The 200 Mammals group is already discussing iteratively updating their alignment in at least one additional phase as new assemblies come in, and I hope that other groups will follow the same pattern.

Appendix A

Supplementary Information for Cactus

A.0.1 Evaluation on simulated data

20 primate genomes were simulated using Evolver [30], managed using the evolverSimControl (`https://github.com/dentearl/evolverSimControl`, commit `b3236deb`) pipeline. The root genome used was derived from 30 megabases selected from the hg19 genome, and is available at `http://courtyard.gi.ucsc.edu/~jcarmstr/datastore/progressiveCactusEvolverSim.tar.gz` along with the Evolver configuration files that were used. The species tree used for the simulation was obtained from a catarrhine subtree of the 100-way alignment tree available on the UCSC browser. The tree used was, in Newick format: `((((((Human:0.00655,Chimp:0.00684)anc0e:0.00122,Bonobo:0.00784)anc1e:0.003,Gorilla:0.008964)anc2e:0.009693,Orangutan:0.01894)anc3e:0.003471,Gibbon:0.02227)anc4e:0.01204,(((((Rhesus:0.004991,Crab_eating_macaque:0.005991)anc5e:0.001,Sooty_mangabey:0.001)anc6e:0.005,Baboon:0.003042)anc7e:0.01061,(Green_monkey:0.027,Drill:0.03)anc8e:0.002)anc9e:0.003,((Proboscis_monkey:0.0007,Angolan_colobus:0.0008)`

anc10e:0.005,(Golden_snub-nosed_monkey:0.0007,Black_snub_nosed_monkey:0.00
08)anc11e:0.004)anc12e:0.009)anc13e:0.02)anc14e:0.02183,(((Marmoset:0.03,S
quirrel_monkey:0.01035)anc15e:0.01065,White-faced_sapajou:0.009)anc16e:0.0
1,Nancy_Mas_night_monkey:0.01)anc17e:0.01)anc18e;The alignments were generated using Cactus commit 51eb980b. The input files (the simulated genomes as well as input files and Cactus configuration file) are available at http://courtyard.gi.ucsc.edu/~jcarmstr/datastore/progressiveCactus.EvolverSim.CactusInput.tar.gz. A non-default configuration (included in the dataset) was used to change the alignment filtering in both runs to better support the high degree of polytomy in the star-tree runs. Four sets of 2, 6, 10, and 20 genomes were used, each of which were run three times to generate runtime estimates.The runtime statistics were gathered using the `toil stats` command (the overall `Clock` time was used, which represents CPU time spent across all jobs). To generate the recall and precision statistics, MAFs were exported for each run (using `hal2maf` with the `--onlyOrthologs` option using the rhesus genome as a reference) and compared to the Evolver MAF using mafComparator (https://github.com/dentearl/mafTools, commit 82077ac3).

A.0.2 Adding a new genome to the simulated alignment

We evaluated the accuracy of adding a genome to an existing alignment by creating a new alignment of 19 of the 20 simulated genomes described above (holding out the "Crab_eating_macacque" genome), then adding it back in after the fact. All alignments for this analysis were generated using Cactus commit 49e80082.To add the crab-eating macaque back in as the child of an existing node (the add-to-node strategy), we ran a single new alignment with the

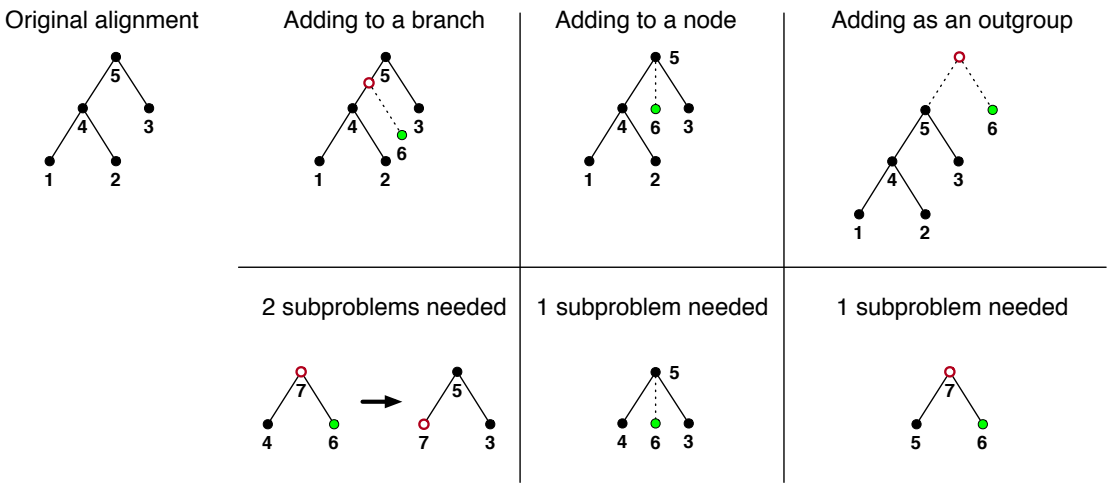


Figure A.1: Methods of adding a genome to a Cactus alignment. The top row shows the different ways of adding a new genome given its phylogenetic position, and the bottom row shows what subproblems would need to be computed for the new genome to be properly merged into the existing alignment. Green circles represent a new genome, and red circles represent newly reconstructed genomes.

tree `(Rhesus:0.006, Crab_eating_macaque:0.007, Sooty_mangabey:0.001)anc6e;`. The anc6e genome from the original, held-out alignment was used as a unreconstructed ancestral input sequence. We set the "runMapQFiltering" option in the config file to "0" and the "alignmentFilter" option to "singleCopyOutgroup", since these options produce a better alignment of polytomies. We merged the resulting HAL file into a new copy of the existing alignment via the command `hal ReplaceGenome<copyofheld-outalignment>anc6e--topAlignmentFile<held-outalignment>--bottomAlignmentFile<add-to-nodealignment>`.To add the macaque by splitting a branch (the add-to-branch strategy), we ran two separate alignments. We ran the first with the tree `(((Rhesus:0.004991, Crab_eating_macaque:0.005991)anc5e:0.001, Sooty_mangabey:0.001)anc6e:0.005, Baboon:0.003042)anc7e;` (with the `--root anc5e` option so that only a single subproblem was run), generating a newly reconstruted anc5e ancestor. We then ran a second alignment with the tree `(anc5e:0.001, Sooty_mangabey:0.001)`

anc6e;, again providing the anc6e assembly from the original alignment rather than inferring a new reconstruction. (We note that these two subproblems could have been run in a single alignment invocation, resulting in the same amount of alignment work but a slightly more complicated merging process.) To merge these two add-to-branch intermediate alignments into a full alignment, we first removed the Rhesus genome from a new copy of the held-out alignment. We then ran `halAddToBranch <held-out alignment> <first add-to-branch alignment> <second add-to-branch alignment> anc6e anc5e Rhesus Crab_eating_macaque 0.001 0.006`.We evaluated the performance of these new alignments using mafComparator in the same way as described in Section A.0.1. In the interest of narrowly determining accuracy of alignments involving the newly added genome, we counted only aligned pairs involving the Crab_eating_macaque genome when calculating precision, recall, and F1 scores.

A.0.3 Evaluation of the effect of the guide tree

The guide-tree analysis was performed on a set of 48 bird genomes originally published in 2014 [55]. To reduce the amount of alignment work required, we subsetted these genomes down to the size of only a single chromosome, chicken chromosome 1 (by removing any contig or scaffold which had less than 20% of its sequence alignable to chicken chromosome 1). We used Cactus commit `36304707` for all alignments in this analysis.The Prum and Jarvis topologies were adapted from [103] and [55], respectively. The "permuted" topology was generated starting from the Jarvis topology, via 3 randomly chosen subtree-prune-regraft operations followed by 3 random nearest-neighbor-interchange operations. Each of these three topologies had branch-length estimates performed using phyloFit from the PHAST package [50] based on

Alignment	URL
Jarvis	https://s3.amazonaws.com/alignment-output/cactus48BIRDS_jarvis14.hal
Prum	https://s3.amazonaws.com/alignment-output/cactus48BIRDS_prum15.hal
Consensus	https://s3.amazonaws.com/alignment-output/cactus48BIRDS_consensus.hal
Permuted	https://s3.amazonaws.com/alignment-output/cactus48BIRDS_permute.hal

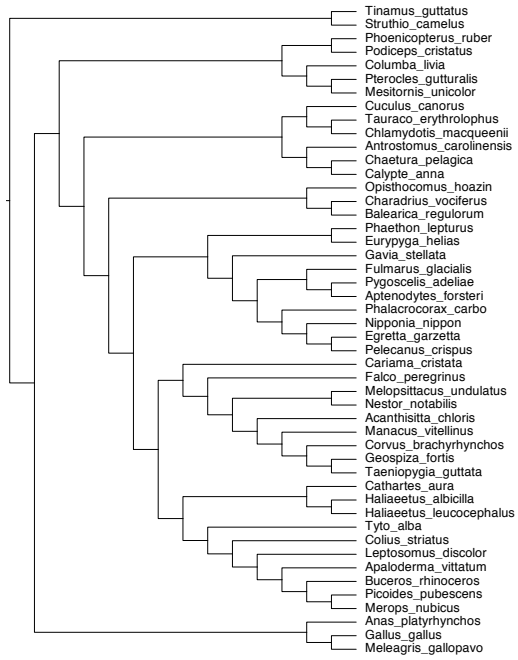
Table A.1: Alignments used in the guide-tree analysis.

fourfold-degenerate sites of BUSCO orthologs. Finally, the "Consensus" tree was produced as a strict consensus of the Jarvis and Prum trees (collapsing all groupings that were not the same in both trees) using the `ape::consensus` method from the APE R package [94]. The branch-lengths for this tree were generated from the fitted branch lengths for the two input trees, using the `consensus.edges` function of the phytools R package [106]. The four final trees that were used in the four Cactus alignments are shown in Figure A.2, and available in supplementary data in Newick format.

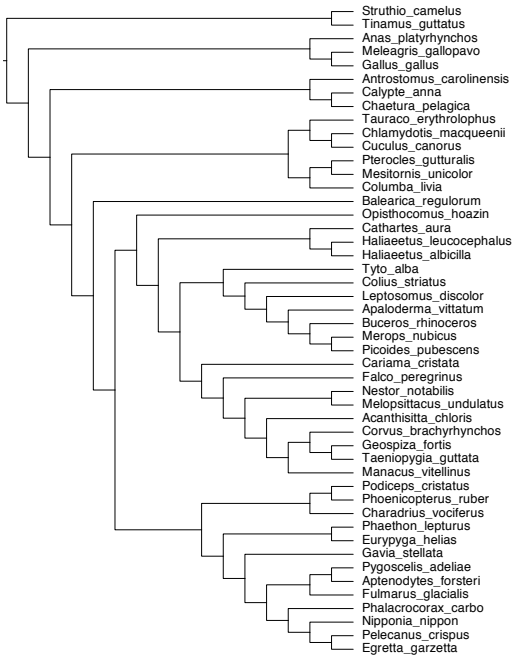
A.0.4 Paralogy-filtering evaluation

A.0.4.1 Alignment of 12 Boreoeutherian genomes

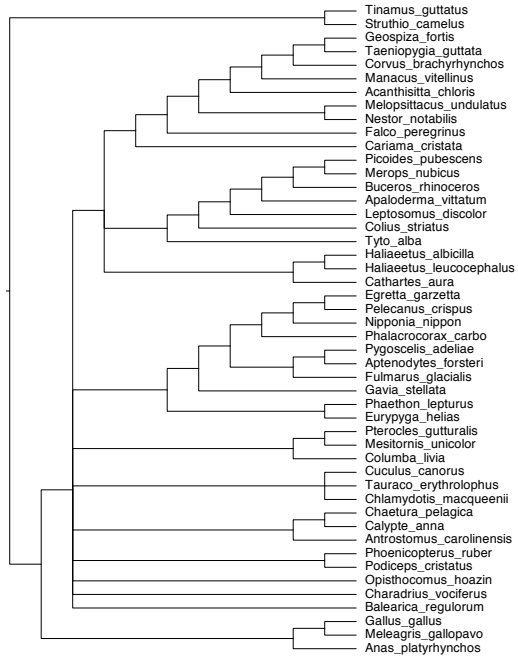
We ran two versions of Cactus (commits `450da74` [best-hit filtering] and `aca859f` [outgroup filtering]) using the following tree: `((((Human:0.006969,Chimp:0.009727):0.025291,Rhesus:0.044568):0.07,Tree_shrew:0.19):0.03,(Kangaroo_rat:0.17,(Mouse:0.072818,Rat:0.081244):0.11):0.150342):0.02326,((Dog:0.07,Cat:0.07):0.087381,((Pig:0.06,Cow:0.06):0.104728,Horse:0.05):0.05):0.04);` Coverage statistics from the resulting alignments were obtained using the `halCoverage` tool.



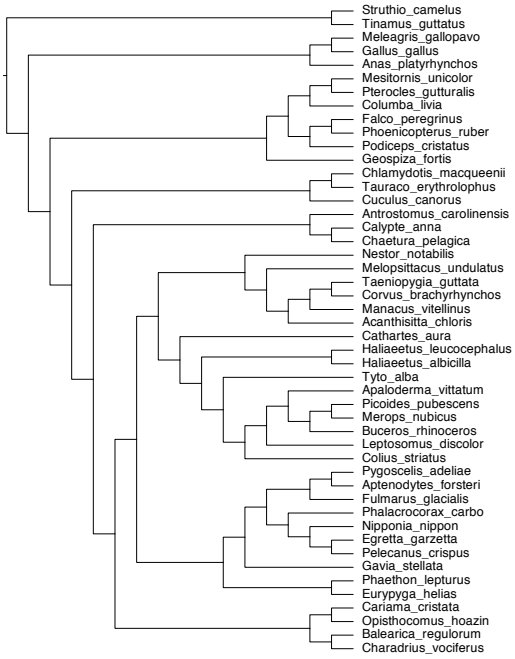
(a) Jarvis



(b) Prum



(c) Consensus



(d) Permuted

Figure A.2: Guide trees used in the guide-tree influence analysis.

Genome	Transcript projections filtered during initial pass	
	Chimpanzee	Gorilla
Outgroup filtering	43709	31678
Best-hit filtering	13567	15765

Table A.2: Number of transcripts filtered out in the initial `pslCDnaFilter` step of CAT, which attempts to remove paralogs and processed pseudogenes.

Genome	Coding genes missing from final set		Coding transcripts missing from final set	
	Outgroup filtering	Best-hit filtering	Outgroup filtering	Best-hit filtering
Chimpanzee	1716	1612	6244	5872
Gorilla	1829	1647	6469	6100

Table A.3: Number of human genes / transcripts that have no assigned ortholog in the "consensus" CAT gene set across the different alignments.

A.0.4.2 Annotation using CAT

We produced two alignments using Cactus on the UCSC hg38, panTro6, and gorGor5 assemblies using the same Cactus versions mentioned above. We ran the CAT pipeline at commit `7a8c7e24`, using the GENCODE V30 gene set [39]. We projected the transcripts solely via transMap without the use of the AUGUSTUS modes. Multiple-mapping statistics as well as the gene composition of the final gene set were taken from the `filter_tm_metrics.json` file in the CAT output.

A.0.4.3 Duplication-timing evaluation

The duplication-timing evaluation was performed using a custom pipeline (`https://github.com/joelarmstrong/treeBuildingEvaluation`) designed to sample columns from a HAL file and evaluate their trees against an independently re-estimated tree of the same region.

For this analysis we used the the two 12-boreoeutherian alignments described above, sampling 10,000 columns from the human genome. The comparison trees were built from a context of 1000 bases around the entries in each sampled column using FastTree [102] 2.1.10 and the `-gtr -nt` options. Only duplicated columns were counted in the final output (columns containing no duplications did not count in the results). The coalescence pairs were evaluated using the `--onlySelf` option, meaning that only pairs that included the sampled site were counted in the results. To avoid weighting columns with a high number of copies per genome more than columns with a low number of copies per genome, only a single coalescence was randomly sampled per column.

A.0.5 Micro-indel events within the 600-way

We extracted all insertion and deletion events by running the `halBranchMutations` tool on every branch in the 600-way alignment. The ungapped insertion and deletion calls (represented by "I" and "D" respectively within the output file) were filtered so that only calls spanning less than 20bp (in the child for insertions, and the parent for deletions) were counted. The rate for each branch was then obtained by dividing the count of these micro-indel events by the total amount of sequence present in the child.

A.0.6 Generation of the 600-way alignment

The 200 Mammals (200M) alignment was composed of two sets of genomes: newly assembled DISCOVAR assemblies and Genbank assemblies. The DISCOVAR genomes were masked with RepeatMasker [112] commit `2d947604`, using Repbase [8] version 20170127

as the repeat library and CrossMatch as the alignment engine. The pipeline used is available at `https://github.com/joelarmstrong/repeatMaskerPipeline`. The guide-tree topology was taken from the TimeTree database [71], and the branch lengths were estimated using the least-squares-fit mode of PHYLIP [35]. The distance matrix used was largely based on distances from the 4d site trees from the UCSC browser [46]. To add those species not present in the UCSC tree, approximate distances estimated by Mash [93] to the closest UCSC species were added to the distance between the two closest UCSC species. The final guide tree is embedded in the HAL file, and available using the `halStats --tree` command.The 363 assemblies in the B10K alignment comprised four sets: 236 newly sequenced species for the "family" phase of the project, assembled using SOAPdenovo2 and AllpathsLG, 42 assemblies already sequenced from the "order" phase of the project, 36 assemblies taken from GenBank, and 49 assemblies contributed by other research groups. For the avian guide-tree, we used a tree that the B10K consortium derived as preliminary data from ultraconserved elements.Both alignments were run on the AWS cloud over the course of 3 weeks for the avians and 2 months for the mammals, using a maximum of 240 `c3.8xlarge` instances and 20 `r3.8xlarge` instances. Because Toil's autoscaling mode was used, this capacity was only fully utilized during the initial phase of the alignment, when the potential for parallelism was at its highest.The 600-way alignment was formed by aligning the two roots of the B10K and 200M alignments, using the xenTro9 (frog), latCha1 (coelacanth), and danRer11 (zebrafish) assemblies as outgroups. This created a "linker" alignment connecting the roots of the two alignments. The B10K and 200M alignments were then added to this linker alignment using the `halAppendSubtree` command.

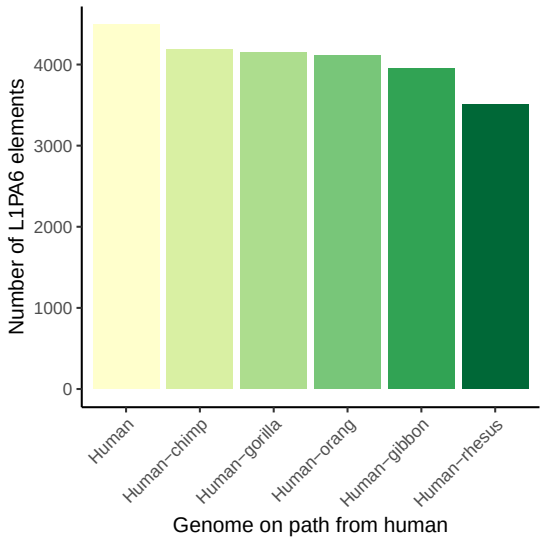


Figure A.3: Number of L1PA6 elements within ancestral genomes.

A.0.7 Repetitive elements within ancestral sequences

We ran RepeatMasker [112] on all ancestral assemblies of human within the 600-way alignment (using RepBase [8] version 20170127, selecting the "primate" repeat library and choosing CrossMatch as the alignment engine). We additionally ran the same pipeline against human (as existing annotations used the "Homo_sapiens" repeat library). All ancestors up to human-rhesus had over 78% of the human complement of L1PA6 elements (Figure A.3).

A.0.8 Removing recoverable sequence

The original CAF algorithm described in [96] was focused on removing small rearrangements, while retaining as much of the original alignment relationships as possible in the filtered cactus graph. However, because the input local alignments are insensitive, the original alignment relationships are likely to have missed certain homologies. This can result

in what we term *incomplete blocks*: blocks that contain some alignment relationships but are missing others, i.e. are proper subsets of what the corresponding "true" alignment block. In our anchor-and-extend process, once a block becomes an anchor it can never be modified. As a result, these incomplete blocks will remain incomplete: they prevent the true alignment relationship from being found, even if an adjacent syntenic anchor block is complete and contains all desired alignment relationships. These problematic incomplete blocks become more prevalent at longer evolutionary distances: the local aligner will miss more true homologies at increasing distances, causing more incomplete blocks and in turn a far worse alignment.To remove these incomplete blocks, Cactus originally relied on a heuristic that identified blocks that were "likely" to be incomplete, removing blocks which did not have alignment relationships between all ingroups. However, this heuristic performed poorly in the presence of deletions or missing data: any large deletion in one ingroup could cause huge stretches of the other ingroup(s) to be left unaligned. To remedy this, we have developed a new alteration to the CAF algorithm, one that now focuses on maximizing the potential size of the alignment graph *after* extension as opposed to *before* extension. We call this addition *removing recoverable chains*, because it identifies chains in the cactus graph that represent alignments which could be recovered by the extension process.The algorithm is applied as a post-processing step after the CAF process described in [96], which proceeds as normal. After the cactus graph is created and filtered, the algorithm identifies *recoverable blocks*. Each block is composed of segments, each of which represent a non-overlapping region of a sequence and which strand is being aligned; we briefly review the necessary terminology, but see [95] for additional context. We call a segment *a left-adjacent* to another segment *b* if *a* represents the positive strand and *b* comes before *a* in their sequence

and there is no other segment between them. Similarly, we call a left-adjacent to b if a is on the negative strand and a comes before b in their sequence ordering with no other intervening segment. If a is left-adjacent to b, then b is *right-adjacent* to a.A block is called *recoverable* if, in the case that the block were removed, all its regions would be contained entirely within a single end alignment in the BAR extension phase. The end alignments are identified by looking at all unaligned sequence between the adjacent segments of a single *end* of a block: in short, two end alignments are created for every block, one for all sequence between each segment and its left-adjacent segment, and similarly for the right-adjacent segments. In practice, this means that for some block A, it is recoverable if all its segments are all left- or right-adjacent to segments from the same block $B \neq A$.Whether a block is recoverable depends only on its immediate neighboring blocks. However, it is interesting to consider the maximum set of recoverable blocks, and, by contrast, of unrecoverable blocks — these unrecoverable blocks represent a minimal set of anchors that can be extended from to recover the alignment relationships from the original sequence graph as well as potential additional alignment relationships.Since the chains and nets within the cactus graph represent a hierarchy of the rearrangements implicit in the alignment, they are helpful for finding this a smaller set of anchors to extend from. We consider what anchors could provide recoverability to a block: if a block A's segments would lie within the end alignment of B if all the recoverable blocks between B and A, including A, were destroyed, we call A *recoverable given* B. The relationship is transitive: if block A is recoverable given block B, and B is recoverable given C, then A is recoverable given C. All blocks in a chain are recoverable given each other, since all blocks in a chain are collinear with each other, potentially with intervening rearrangements located further down the chain/net hierarchy. Similarly, if any

block in a chain is recoverable given another block above the chain in the chain/net hierarchy, the entire chain is recoverable given that block. Due to this fact, in order to determine the recoverability status of all blocks, we only have to examine the blocks at the ends of chains and their immediate neighbors, rather than every block.Though in principle we would need to keep only one block within even unrecoverable chains (since all other blocks within the chain would be recoverable given that single block), to save computational effort in realignment we only destroy or keep entire chains as a unit. In the same spirit, to avoid spending needless effort when the chain is recoverable but very likely is not *incomplete*, we apply a heuristic and do not remove chains that contain the same number of copies in all ingroups and outgroups.After identifying and removing all recoverable blocks, some blocks previously marked unrecoverable may become recoverable (because adjacent blocks were removed). For this reason, we run the process of identifying and removing recoverable chains multiple times in a loop, until either no recoverable chains are identified or a limit on the number of cycles is reached. The structure of the cactus graph may change after removing recoverable blocks, so we recompute the cactus graph after every removal step. The process that is followed is described in Algorithm 1.

A.0.9 Improvements from removing recoverable sequence

To quantify the effect that the process of removing recoverable chains (described above) had on real alignments, we ran alignments on a set of 9 Euarchontoglires genomes with the feature turned on and off. The tree used was:`((((((human:0.00877,gorilla:0.008964):0.009693,orang:0.01894):0.015511,rhesus:0.037601):0.07392,tarsier:0.1114):0.034014,tree_shrew:0.19114):0.002,(kangaroo_rat:0.171759,(chinese_ham`

Algorithm 1 Recoverable-chain destruction

```
function REMOVERECOVERABLECHAINS(G, n)
    for 1 ... n do
        cactusGraph ← CreateCactusGraph(G)
        RecoverableChains ← ∅
        for chain C in cactusGraph do
            if

    ▷ A single adjacent end offers the potential for recoverability
    (|C.leftAdjacencies| = 1 or |C.rightAdjacencies| = 1)
    ▷ Shared adjacencies indicate a non-recoverable rearrangement
    and C.leftAdjacencies ∩ C.rightAdjacencies = ∅
    ▷ Links between chain ends indicate a non-recoverable duplication
    and C.leftEnd ∉ C.rightAdjacencies
            then
                RecoverableChains ← RecoverableChains ∪ {C}
            end if
        end for
        if |RecoverableChains| = 0 then
            break
        else
            Destroy each chain in RecoverableChains
        end if
    end for
end function
```

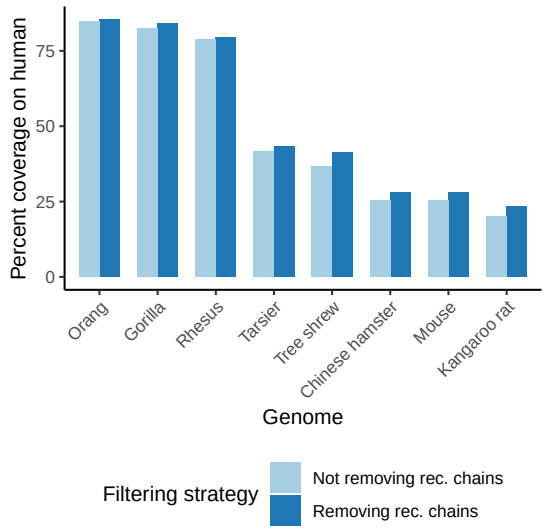


Figure A.4: Coverage (on the human genome) from alignments with and without removing recoverable chains after the CAF process. While the coverage is increased overall across all genomes when removing recoverable chains, the increase is relatively larger in more distant species.

`ster:0.14,mouse:0.132282):0.11015):0.114051)euarchontoglires:0.020593,(cow:0.18908,dog:0.13303):0.032898);` We used Cactus commit `56874bde`, with the `--root euarchontoglires` option so that cow and dog were used only as outgroups. Coverage on human increased for all genomes when recoverable chains were removed, especially for those most distant from human (Figure A.4). This likely reflects the fact that though the losses caused by not removing recoverable chains in any single subproblem are relatively small, they can compound to be quite significant in large alignments, since many subproblems are involved in creating the alignment between distant species (such as human and mouse, which are separated by 7 internal nodes in this tree).

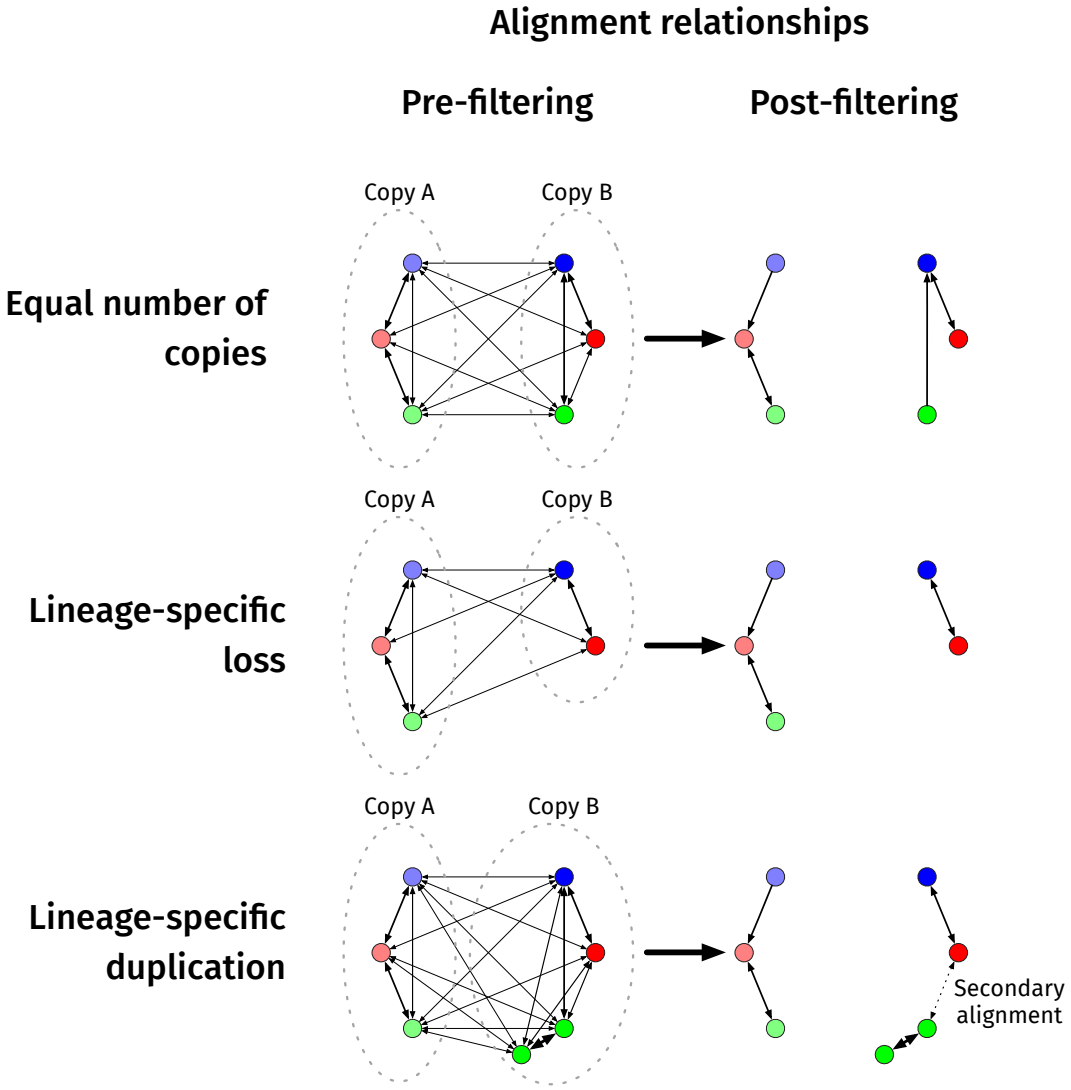


Figure A.5: A visualization of the best-hit filtering method. Here, each node of the directed graph indicates a single base, and edges represent pairwise alignment relationships (the color of the node indicates the species the base belongs to, and higher thickness of edges represents higher scores of the pairwise alignments). Since Cactus's alignment columns represent the transitive closure of the input pairwise alignment relationships, the final alignment relationships will be represented by connected components within this graph. Taking the single best hit (so that this graph contains at most one outgoing edge per base) results in the correct separation between copies if orthologous copies have higher score, but some lineage-specific duplications require secondary, non-best-hit alignments to bring together orthologs from different species.

Appendix B

Supplementary Information for archosaur reconstruction

B.1 Whole Genome Alignment and Ancestral Genome Reconstruction

The whole genome alignment of 23 taxa was computed using progressive-Cactus (`https://github.com/glennhickey/progressiveCactus`) using its default parameters and the phylogeny shown in Figure 3.1A. The genome assemblies used are listed in Table B.1. The topology of the phylogeny was derived by manually merging a subtree of the UCE trees with results from the avian phylogeny sister paper (76) along with published phylogenies for passerine birds (77), parrots, (78), and turtles (79). We used a 512 CPU cluster for running the local alignment jobs and a 1 terabyte shared memory machine with 64 cores for computing the CAF and reconstruction algorithms (75, 80). Ancestral reconstruction of all internal nodes was

performed as part of this process. To improve the ancestral base calls we used the `ancestorsML` tool in the HAL tools library (`https://github.com/glennhickey/hal`) (81) to call bases by maximum likelihood, using the general reversible continuous time nucleotide substitution model implementation from the PHAST package (60). To parameterise the model and estimate branch lengths for the topology (shown in Figure 3.1) we used the phyloFit tool (82) on conserved fourfold degenerate sites in alligator genes, as described below. We also stored the posterior probability of these base calls given the model, and these results were used to calculate the expected accuracy of base calls in the archosaur genome. These results are shown in Figure 3.2A.

B.2 Whole Genome Alignment Analyses

The following gives technical definitions of the WGA analyses performed. A *whole genome alignment* (WGA) is formally a partitioning of the residues within a set of genomes into a set of aligned columns, each of which represents a set (technically an equivalence class) of residues inferred to be homologous. Given a chosen subset of genomes X within the WGA, a *non-duplicated column* for X is a column containing one or zero residues from each of the species in X. Similarly, given a chosen subset of species X within the WGA, a *single copy column* for X is a column containing exactly one residue from each of the species in X.

B.2.1 Percentage Identity

For a pair of genomes their *percentage identity* is the proportion of single copy columns for the pair that do not contain a wildcard character representing either genome and in which

the nucleotide from both genomes is identical. The percentage identity value reported therefore includes the maximum number of columns where there is no apparent ambiguity about the ancestry of the residues. Table B.2 shows the percentage identity between each of the three crocodilian genomes. Near identical results were produced using Mummer (173).

B.2.2 Fourfold Degenerate Codon Substitution Rates

In the WGA, a column is *conserved fourfold degenerate* if the column contains a residue that is a fourfold degenerate site in an annotated transcript in a given reference genome, and the two previous bases (in the opposite direction to the direction of transcription) in every leaf genome are such that each leaf genome site in the column is fourfold degenerate. Coding transcripts in alligator were filtered to only those with all the columns in their coding exons non-duplicated among crocodylia, chicken, zebra finch, and Carolina anole. The sites in alligator that corresponded to conserved fourfold degenerate columns in non-duplicated alligator coding sequences were then extracted. The program halPhyloPTrain (available in the HAL tools library) was used on these sites to estimate substitution rates for every branch in the WGA.

To validate these branch lengths, the non-duplicated alligator coding transcripts described above were also exported in Phylip format and further processed to remove regions that might bias the estimation of neutral rates before analysis in RAxML version 8.0.0 (64). We retained codons only if they were present in all taxa and did not have any internal gaps. This was done because the WGA allowed some indels of length one or two in coding regions; this typically occurred in regions where one or more of the sequences were poorly assembled (based upon visual inspection). The rationale for this stringent filtering is provided below in the section

on 4-fold degenerate sites. This resulted in a matrix of 144,733 nucleotides and no missing data. Branch lengths were estimated for this alignment using the `-f e` option in RAxML with the `GTRGAMMA` model and the phylogeny used for other WGA analyses (see above). The tree length was 3.180126 and ML estimates of the parameters describing 4D site evolution were:

Γ distribution shape parameter	
α	1.658157
GTR rate parameters (normalized to the G-T rate)	
rate A-C	0.750544
rate A-G	4.203959
rate A-T	0.591019
rate C-G	1.387873
rate C-T	2.267481
Equilibrium base frequencies	
$\pi(A)$	0.280429
$\pi(C)$	0.252743
$\pi(G)$	0.167161
$\pi(T)$	0.299667

The resulting phylogeny with ML estimates of branch lengths is shown in Figure 3.1. A similar analysis was conducted by after increasing the stringency of the filtering to require that the amino acid encoded by the *conserved fourfold degenerate column* is itself conserved. This reduced the length of the alignment to 114,709 nucleotides but it had a negligible impact upon the branch length or parameter estimates. Estimates of the rates based upon 4D sites are presented in Table B.3.

B.2.3 Transposable Element Substitution Rates

Any transposable element Y defines a nonempty subsequence $x_i, x_{i+1}, \ldots, x_j$ of a chosen reference genome (here the common ancestral genome of the extant crocodilian genomes

in our analysis). We call Y syntenic with respect to a subset of genomes X, if:

1. The residues in $x_{i-m}, x_{i-m+1}, \ldots, x_{j+m+1}$ are all members of non-duplicated columns for X, where m is a flanking parameter (set to 2kb in this analysis; this ensures that the phylogeny is apparently unambiguous across the element).

2. For each pair of contiguous residues in $x_{i-m}, x_{i-m+1}, \ldots, x_{j+m+1}$, if the columns they are contained in both contain residues from another genome in X, then those residues in the other genome are in the same order and orientation as in the chosen reference genome and are separated by no more than 100 bases in the other genome (this ensures that no rearrangements other than indels less than 100bp in length and substitutions have been observed to effect the sequences). We use the set of single-copy columns that contain residues from syntenic transposable elements to calculate the substitution rate in transposable elements.

To estimate rates we used the strand symmetric general reversible continuous time substitution model implemented in the Phast package continuous time substitution model (82), using the halPhyloPTrain program on the single-copy columns from syntenic transposable elements in the common ancestor of Crocodylia. Table B.4 below shows the TE substitution rates.

B.2.4 Micro Insertion and Micro Deletion Rates

For a pair of genomes A and B, a *clean insertion* in A with respect to B is a nonempty subsequence $x_i, x_{i+1}, \ldots, x_j$ of a sequence in A such that:

1. The residues in $x_i, x_{i+1}, \ldots, x_j$ are not aligned to any residues in B.

2. The residues in $x_{i-k-1}, \ldots, x_{i-1}$ and $x_{j+1}, \ldots, x_{j+k+1}$ are each aligned in single copy columns for $\{A, B\}$, where k represents a number of cleanly aligned residues flanking the insertion. This condition ensures no duplications that suggest ambiguity in the phylogeny. After some testing, in this analysis, $k = 5$, though other larger values of k produce similar results.

3. The corresponding residues in B aligned to $x_{i-k-1}, \ldots, x_{i-1}$ and $x_{j+1}, \ldots, x_{j+k+1}$ are in the same order and orientation in B as in A. This ensures the structural change is indeed an insertion rather than a more complex rearrangement.

4. None of the residues in the alignment columns containing $x_{i-k-1}, \ldots, x_{j+k+1}$ represent the wildcard character. This avoids labeling scaffold gaps as insertions.

A *clean insertion* in A with respect to B is, reversely, a *clean deletion* in B with respect to A. A clean indel (insertion or deletion) is a *micro* event if the inserted or deleted subsequence is less than or equal to 10 residues in length. A *clean adjacency* is either a clean insertion or deletion, or equivalent to a clean insertion or deletion in which the inserted subsequence has zero length; a clean adjacency represents a place where there could have been a clean indel, but potentially none was observed.

Let an induced subtree of the phylogeny connecting a chosen genome A, its sister genome and their closest outgroup genome, be termed a *three-leaf subtree* for A. For a chosen genome A with corresponding three-leaf subtree, a clean insertion, deletion or adjacency with respect to its three-leaf subtree is a clean branch insertion, deletion or adjacency, respectively,

in A with respect to both the sister and outgroup genomes of the three-leaf subtree. Note this definition discounts clean insertions, deletions or adjacencies which differ between the outgroup and sister genomes, i.e. the indel subsequence has to have the same length in both other species. Defining events with respect to three-leaf subtrees gives confidence in the categorization of the event as an insertion, deletion or clean adjacency.

The insertion and deletion rates reported are the ratio of clean micro insertion or deletion events per clean adjacency. Normalising by clean adjacency proved necessary to factor in coverage differences between assemblies. Table B.5 shows the measured rates of clean insertions and deletions in each of the leaf taxa of the WGA.

B.2.5 Gene Synteny

For a pair of genomes A and B and pair of genes X and Y on A, we say the pair X and Y are *candidates for synteny* if X and Y both:

1. map uniquely from A to B (no evidence of duplication in the other species),
2. map to the same scaffold on B,
3. map at least 90% of their sequence to B,
4. reciprocally preserve their structures to B (i.e. X and Y must be preserved from A to B, and their images on B must be preserved back from B to A, see below for technical definition of preservation).

If X and Y are candidates for synteny and they are in the same order and orientation on B as on A, they are *syntenic*, otherwise they are *broken*. If X or Y are not candidates for

synteny, they are considered invalid and the pair is neither considered syntenic nor broken. The ratio of broken/syntenic pairs is the *gene pair synteny rate*. This carefully constructed analysis was necessary to unambiguously identify orthologous pairs of gene and minimise assembly differences impacting the results, though it is still likely to be somewhat affected by assembly composition and errors.

B.3 Archosaur Reconstruction Analyses

Below we detail analyses used to assess the reconstructed archosaur assembly.

B.3.1 Estimating Potential Missing Sequence in the Archosaur Assembly

Due to the parsimony based simultaneous progressive alignment and reconstruction approach used to construct the WGA, any sequence in alligator that has a homolog outside the crocodilian lineage must have a homolog in the reconstructed archosaur. This implies that alligator sequences without an ortholog in the archosaur do not have an ortholog outside the crocodilian lineage. Misalignment and assembly errors will tend to reduce the quality of the ancestral reconstruction (usually by missing sequence), and thus lower the amount of sequence aligned. To estimate how much sequence should have been included in the archosaur reconstruction but was not, alligator fragments that did not align to archosaur at all were aligned against a selection of other leaf genomes (Table B.6) using LASTZ (version 1.03.52) with the following parameters: `[multiple,unmask]`, which ensure that any repetitive sequence will not be unaligned due to heuristics that by default avoid the alignment of soft-masked sequence.

While most fragments (87%) mapped to crocodile, the substantial majority do not map to these outgroup genomes (e.g. 91% of unmapped fragments do not align to anywhere in chicken) using LASTZ. This suggests that the reconstructed genome, despite its small size, already approaches the maximum possible size for a reconstructed genome given current alignment techniques. The small minority of regions that do map to out-group genomes are largely repetitive (mapping to many places in the outgroup genomes), suggesting the reconstruction of repetitive elements is an area of future improvement.

B.3.2 Element Categories for Archosaur Analysis

To avoid issues with double-counting elements and bases in the mapping, phyloP, and structure-preservation analyses, the BED files for these categories were pre-processed. Gene and chained-CDS categories were processed to select only the longest transcript where there was overlap between multiple elements on the same strand. All other categories had their elements merged together where overlapping to avoid multiple-counting.

The gene annotations used were as described above for the alligator and, for the chicken, RefSeq gene annotations available from the assembly hub (see below), or at `http://hgwdev.cse.ucsc.edu/~jcarmstr/permanent/galGal4_refSeq.bed`.

B.4 Selection Analysis

The halPhyloP tool, available from the HAL tools library, was used to generate phyloP scores for all columns in the alignment. The input branch lengths were determined by running

phyloFit on conserved alligator fourfold degenerate sites, as described above. Using the WGA, each column was lifted over to each genome, creating a phyloP wiggle track for each of the 23 leaf and 22 ancestral genomes in the WGA.

B.5 Order Preservation

An element, either exon, UTR, intron, etc., is defined by an interval of a genome. For a pair of genomes A and B and element Y in A, for a pair of successive residues in Y that align to a sequence X in B, we say their *adjacency is preserved* if the corresponding residues aligned in X are in the same order and orientation and separated by less than 100bp. We say the structure of Y is preserved if for all such pairs the adjacency is preserved and at least 25% of bases in Y align to X, and X is the sequence in B where the majority residues in Y align without self alignment (self duplication). This definition ensures that to have preserved structure at least a reasonable minority of bases must align to a single sequence and be organized as in the reference genome. If Y is a coding sequence (CDS), comprising the coding portions of a gene's exons, it is treated as a single element, except that residues in the interstitial introns are ignored, and introns are allowed to change in size by up to 100kb in X.

B.6 Extant Mapping Controls

The proportion of elements and adjacencies that were preserved is shown for alignments between alligator to archosaur (Figure B.1), alligator to chicken (Figure S18) and chicken to archosaur (Figure S19). These comparative controls show we get similar, but uniformly slightly

higher results mapping chicken, rather than alligator, genes to the archosaur (presumably due to differences in gene sets, as the evolutionary distance is expected to be greater), and substantially higher mapping and order and orientation preservation results mapping extant annotations (either alligator or chicken) to archosaur than mapping alligator annotations to chicken, or vice versa.

B.7 Assembly Hub

A Comparative Assembly Hub for the WGA is available for the UCSC genome browser (174) at `http://genome.ucsc.edu/cgi-bin/hgHubConnect` (locate the "Croc and Bird Hub" link). From it, it is possible to browse the genomes, annotations and alignments, and download (via the Table Browser), portions of the WGA, the sequences of the reconstructed genome as well as the alligator, gharial and crocodile gene sets.

UCSC Genome ID	Common name	Species name
falChe1	Saker falcon	*Falco cherrug*
falPer1	Peregrine falcon	*Falco peregrinus*
ficAlb2	Collared flycatcher	*Ficedula albicollis*
zonAlb1	White-throated sparrow	*Zonotrichia albicollis*
geoFor1	Medium ground finch	*Geospiza fortis*
taeGut2	Zebra finch	*Taeniopygia guttata*
pseHum1	Tibetan ground jay	*Pseudopodoces humilis*
melUnd1	Budgerigar	*Melopsittacus undulatus*
amaVit1	Puerto Rican parrot	*Amazona vittata*
araMac1	Scarlet macaw	*Ara macao*
colLiv1	Rock pigeon	*Columbia livia*
anaPla1	Mallard duck	*Anas platyrynchos*
galGal4	Chicken	*Gallus gallus*
melGal1	Turkey	*Meleagris gallopavo*
strCam0	Ostrich	*Struthio camelus*
allMis2	American alligator	*Alligator mississippiensis*
croPor2	Crocodile	*Crocodylus porosus*
ghaGan1	Gharial	*Gavialis gangeticus*
cheMyd1	Green seaturtle	*Chelonia mydas*
chrPic1	Painted turtle	*Chrysemys picta bellii*
pelSin1	Soft-shell turtle	*Pelodiscus sinensis*
apaSpi1	Spiny soft-shell turtle	*Apalone spinifera*
anoCar2	Carolina anole	*Anolis carolinensis*

Table B.1: Genome assemblies used to construct the WGA.

Genome pair	Percent ID
Alligator, crocodile	92.9%
Crocodile, gharial	95.7%
Alligator, gharial	93.4%

Table B.2: Percentage identity for each pair of crocodilian genomes.

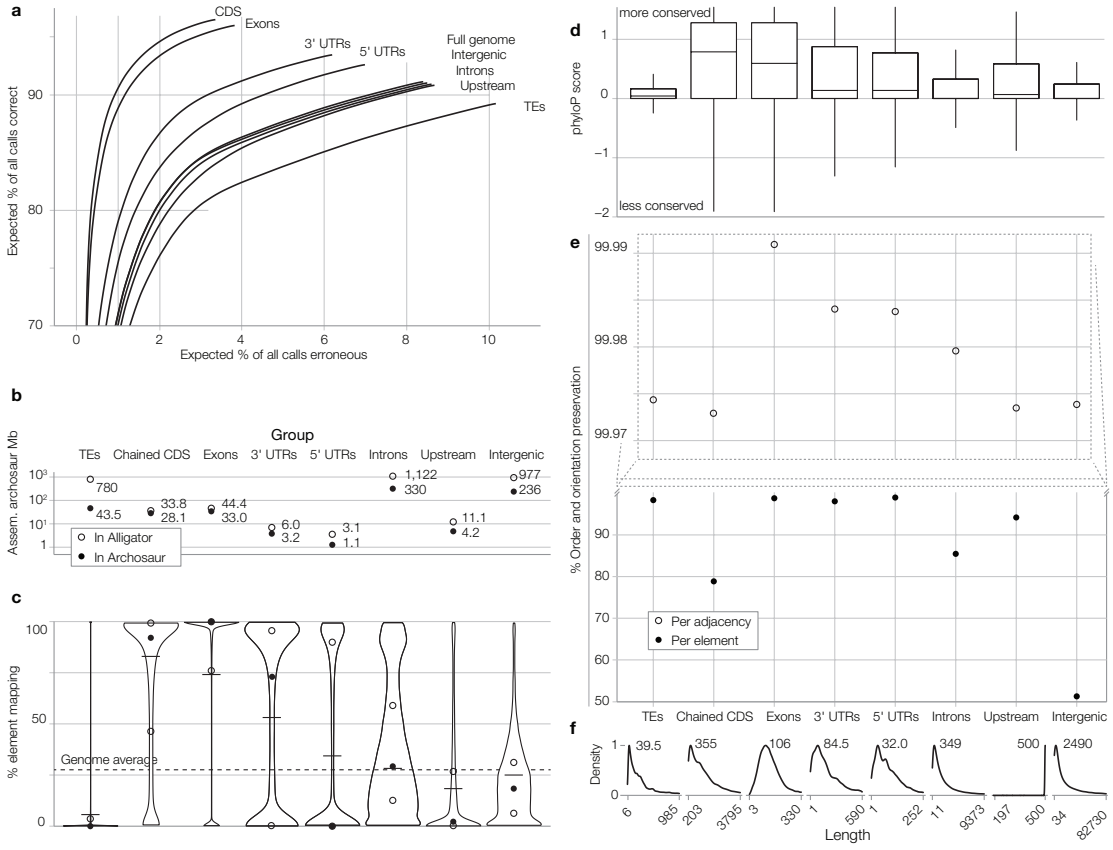


Figure B.1: Analyzing the archosaur assembly using projected alligator annotations. a) Expected base reconstruction accuracy. b) Total archosaur bases assembled. c) Proportion of annotations mapping as violin plots, with box plot (circles) and average (line) inset. d) Conservation scores (phyloP). e) Order and orientation preservation of annotated elements. f) For comparison, length distributions of annotations on alligator. Note: 'Chained CDS' category includes complete CDS, with introns spliced out. This is expansion of the analysis presented in Figure 3.2.

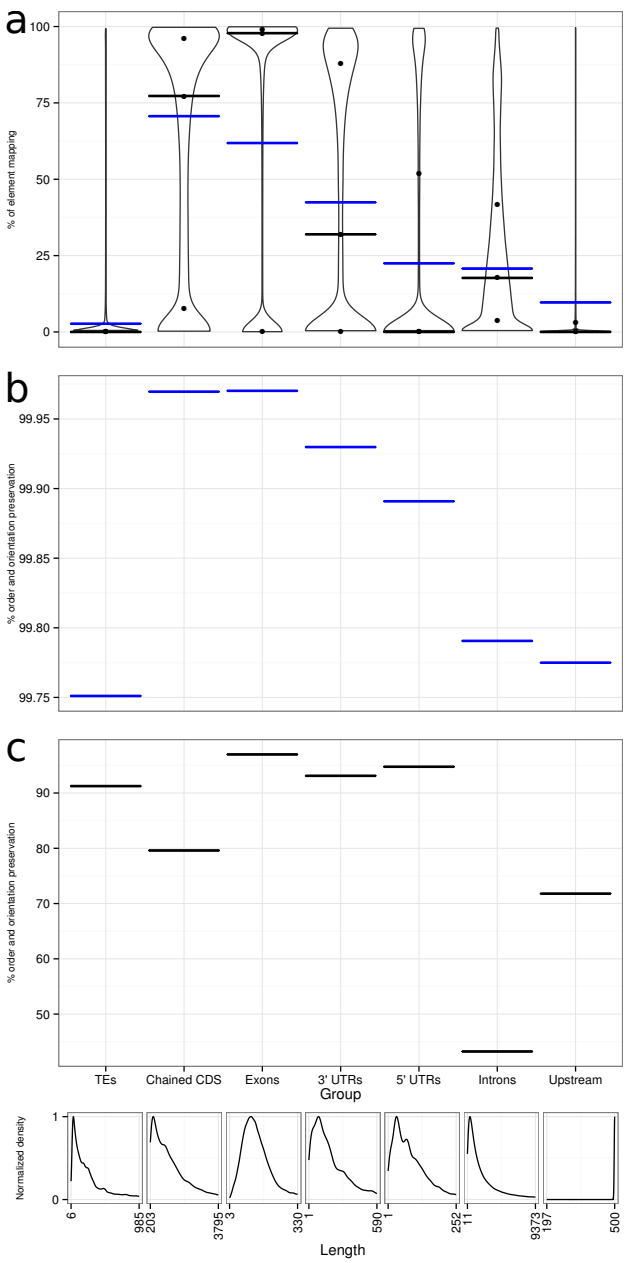


Figure B.2: Mapping and order-and-orientation-preserving statistics from alligator to chicken in the alignment. A) Violin plot for the percent of an element that maps to the target genome. Blue lines represent the average mapping for the category, and dots show inner quartiles. B) Percent of adjacencies preserved. C) Percent of elements preserved.

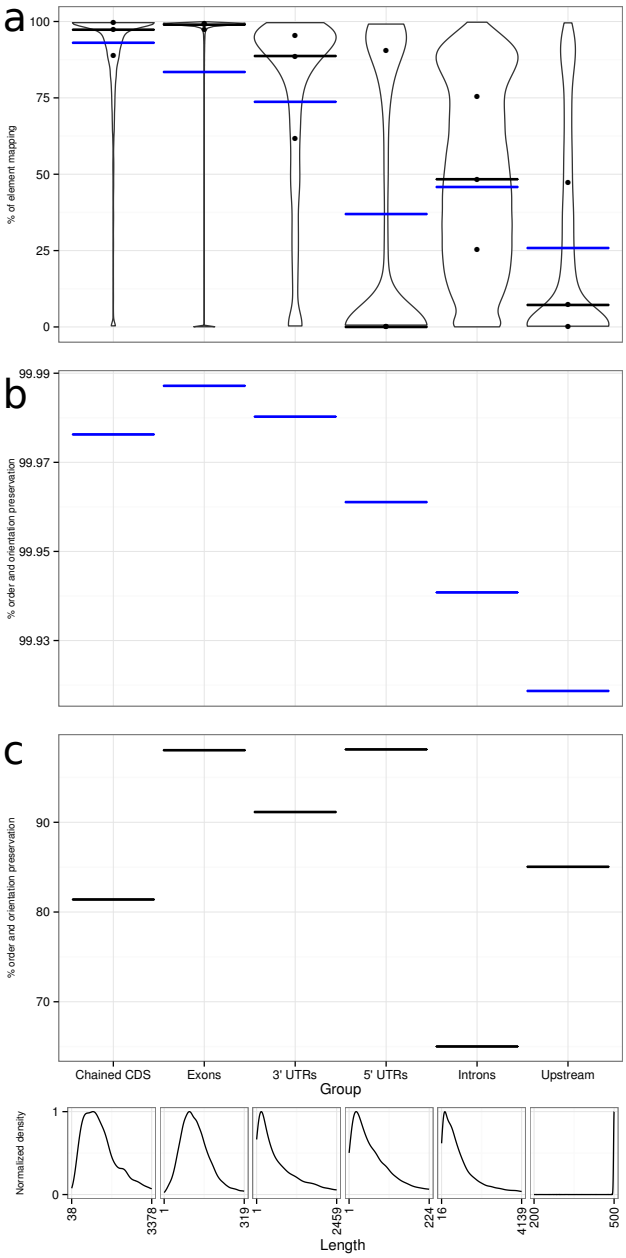


Figure B.3: Mapping and order-and-orientation-preserving statistics from chicken to archosaur in the alignment. A) Violin plot for the percent of an element that maps to the target genome. Blue lines represent the average mapping for the category, and dots show inner quartiles. B) Percent of adjacencies preserved. C) Percent of elements preserved.

Genome	4D substitution rate	
	Filtering strategy 1	Filtering strategy 2
Alligator	0.0263	0.0254
Crocodile	0.0221	0.0211
Gharial	0.0172	0.0167
Crocodile-gharial common ancestor	0.0147	0.0144

Table B.3: 4D site substitution rates for the branches directly above each crocodilian genome.

Genome	TE substitution rate
Alligator	0.0260
Crocodile	0.0246
Gharial	0.0188
Crocodile-gharial common ancestor	0.0260

Table B.4: TE substitution rates for the branches directly above each crocodilian genome. Note that the tree can be arbitrarily rooted between the alligator and the crocodile-gharial common ancestor.

Leaf genome	Clean micro-insertion rate	Clean micro-deletion rate
Parrot	0.000436	0.001174
Mallard duck	0.001804	0.002931
American alligator	0.000617	0.00046
Spiny soft-shell turtle	0.000533	0.001075
Scarlet macaw	0.000448	0.000999
Green sea turtle	0.000969	0.001583
Painted turtle	0.000944	0.001669
Rock pigeon	0.001239	0.002664
Crocodile	0.000567	0.001061
Saker falcon	0.000045	0.000059
Peregrine falcon	0.000057	0.000086
Collared flycatcher	0.001203	0.003072
Chicken	0.000866	0.001426
Medium ground finch	0.000616	0.0013
Gharial	0.000381	0.000692
Turkey	0.001042	0.002382
Budgerigar	0.000666	0.001691
Soft-shell turtle	0.000512	0.001147
Tibetan ground jay	0.000978	0.002045
Ostrich	0.002045	0.001801
Zebra finch	0.001048	0.002381
White-throated sparrow	0.000794	0.001956

Table B.5: Clean micro-insertion and -deletion rates for leaf genomes in the WGA. Note that the Carolina anole does not appear in this table, since it has no three-leaf subtree.

Genome	Alligator fragments missing in archosaur with >= 1 blast hit	Alligator fragments missing in archosaur with >= 10 blast hits	Alligator coding exons missing in archosaur with >=1 blast hit	Alligator coding exons missing in archoaur with >= 10 blast hits
Chicken	440/5000 (8.8%)	328/5000 (6.6%)	3484/22666 (15.3%)	1556/22666 (6.9%)

Table B.6: Alligator elements that did not map to archosaur in the WGA were aligned against multiple leaf genomes using LASTZ. Elements were restricted to >30 bp to prevent spurious alignment, and repeat-masking was ignored in both the element and the target genomes.

Appendix C

Supplementary Information for selection analysis

C.1 Neutral model

To estimate the degree of conservation or acceleration within a column means evaluating the departure from a "neutral" rate of evolution — this rate is described using a neutral model. We estimated a neutral model based on ancestral repeats using an automatic pipeline for estimating neutral models (`https://github.com/joelarmstrong/neutral-model-estimator`). We extracted the ancestral genome from the alignment representing the ancestor of all birds, and ran RepeatMasker [112] to find avian repeats present in that genome (using the species library "aves"). A random sample of 100,000 bases within these repeats was used to extract a MAF, which was used as input to the `phyloFit` program from the PHAST [50] package to create the neutral model (using a general reversible model of nucleotide substitution, `REV`).

The PHAST framework allows only at most a single entry per genome per column, while the output MAFs may contain alignments to multiple copies. To resolve this, `maf_stream` (`https://github.com/joelarmstrong/maf_stream`) was used to combine multiple entries per genome into a single entry (using `maf_stream dup_merge consensus`).

Sex-determining chromsomes are known to evolve at a slightly different rate than autosomes (the fast-X / fast-Z hypothesis) [81, 17, 123]. Furthermore, micro- and macro-chromosomes in birds have been shown to evolve at different rates as well [7, 123]. To remove any potential differences in neutral rates among these chromosomes as a factor, we generated a second set of neutral models which represent the neutral rate on these three chromosome sets (we call this set the "3-rate model"). These models were generated by mapping the ancestral repeat sample described above from the root ancestral genome to the chicken genome, then separating the resulting bases into macro-, micro-, and sex-chromosome bins. For our purposes, we defined micro-chromosomes as any autosomal chromosomes other than chr1-8. We then scaled the ancestral-repeats model described above separately for each of the 3 bins using `phyloFit --init-model <original model> --scale-only`. This 3-rate model was used for all results and figures in the main paper, as well as those in this supplement by default unless specifically mentioned otherwise.

C.2 Conservation/acceleration scores and significance calls

We estimated conservation and acceleration scores for the B10K alignment using phyloP [101, 50] run with the `--method LRT` and `--mode CONACC` scoring options. We ran

this twice using the two neutral model sets described above. When estimating the scores using the 3-rate model we ran each chromosome separately, using the corresponding scaled model belonging to the proper set (macro-, micro-, or sex-chromosomes). Though the HAL toolkit contains a tool that emits phyloP scores, that tool works on the basis of alignment-wide columns, which combine all lineage-specific duplications into a single column: this is undesirable since some alignment-wide columns containing homologies between two or more paralogs may be resolvable into multiple orthologous columns when viewed from chicken. Therefore, we instead ran phyloP on a MAF export referenced on the chicken genome (using the `hal2maf` tool with the `--onlyOrthologs` option). These MAFs were post-processed using the `maf_stream` command in the same way as described in Section C.1.

We obtained the 77-way MULTIZ alignment from the UCSC Genome Browser [46] site (`http://hgdownload.soe.ucsc.edu/goldenPath/galGal6/multiz77way/maf/`). Rather than use the phyloP scores provided by the browser, which were trained on fourfold-degenerate sites using a single neutral model, we estimated new scores using a 3-rate model in the same manner as the 363-way. The 55-way scores were generated by simply providing the avian subtree of the 77-way tree (using the `--tree` option) when fitting the neutral model. Though the resulting scores are based on a different version of the chicken assembly than we used for the primary analysis (galGal6 instead of galGal4), most analysis did not need assembly coordinates. For two aspects of the analysis (the region-specific analysis and Figure C.8) we needed a common coordinate system, so we lifted these scores to galGal4 using the `liftOver` tool (16.2 Mb, 1.5% of the total, were unable to be lifted over).

The two score sets largely agreed on the direction of acceleration/conservation, with the

values in the 363-way set being generally more extreme due to the additional power (Figure C.8).

PhyloP scores represent log-encoded p-values of acceleration. We transformed these scores into p-values, then into q-values using the FDR-correcting method of Benjamini and Hochberg [10]. Any site that had a q-value less than 0.05 was deemed significantly conserved or accelerated (see Figure C.1 for proportions of accelerated/conserved regions).

We extracted the significant accelerated and conserved sites from the phyloP wiggle files using the Wiggletools [120] command `wiggletools gt <threshold> abs`, where the appropriate score threshold was taken from Table C.1.

C.3 Intersection with functional regions of the genome

We split RefSeq genes (obtained via the RefSeq gene track on the galGal4 UCSC browser [46]) into sets of coding exons, untranslated (UTR) exons, and introns. We also downloaded a lncRNA gene set from NONCODEv5 [33] to obtain lncRNA regions and mapped all repeats from the root genome (mentioned in Section C.1) to chicken to get ancestral repeat regions. All of these regions were made mutually exclusive by removing overlaps with all other region types. Finally, 100,000 bases were randomly sampled from each of these mutually-exclusive regions and used to extract a corresponding distribution of scores for each region from the wiggle file. The results are shown in Figure C.2, Figure C.3, and Figure 4.7.

C.4 Distribution of rate of alignment columns

Finding the distribution of rates of alignment columns (relative to the neutral rate) is necessary for determining what strength of conservation is needed for significance. We sampled 10,000 sites at random from each of the galGal4 (for the 363-way) and galGal6 (for the 77-way) assemblies. For the 363-way, a MAF was exported containing the columns for each of these sites using `hal2maf`, while for the 77-way, the `mafFrags` program was used to obtain the columns from the UCSC browser database. The `--base-by-base` mode of phyloP was used to obtain the "scale" parameter for each column, which represents a best-fit multiplier of the neutral model applied to all branch lengths in the tree. (For the 363-way, we divided the columns within the MAF into three separate files according to their bin within the 3-rate model, and used the appropriate model for each resulting MAF.) The results are shown in Figure 4.8, Figure C.5, and Figure C.7.

C.5 Realignment of conserved sites

Our conservation and acceleration calls fundamentally rely on information from the alignment. For this reason, errors in the alignment could potentially cause erroneous acceleration or conservation calls.

We tested the degree to which alignment choices for a given region affect our conservation calls. We sampled 1,000 sites randomly selected from the set of conserved sites on chicken and extracted their columns from the alignment. For each species in each column, we extracted a 2kb region surrounding the aligned site into FASTA format, resulting in 1,000 FASTAs, one

Score set	Lowest significant score	% of sites sig. conserved
55-way (3-rate model)	±2.506	2.1%
77-way (3-rate model)	±2.392	3.7%
363-way (3-rate model)	±1.957	13.2%
77-way (single neutral model)	±2.215	7.0%
363-way (single neutral model)	±1.826	18.3%

Table C.1: Significance thresholds and coverage of conserved site for expected FDR 0.05 in the different phyloP score sets.

for each column. We then realigned these FASTAs using MAFFT [60] and used phyloP on the resulting region to extract a new score for the column containing the chicken site that was originally sampled. The distribution of differences in score (of the realigned score relative to the original score) is shown in Figure C.6: 52% of scores were exactly identical, while 93% were within a range of 1.0 from the original score value (i.e. an order of magnitude in p-value). 8.4% of conserved sites had a realignment score that dropped below the significance threshold after realignment; however, most of these cases were only slightly above the threshold to begin with (median original score of 2.26, mean 2.41).

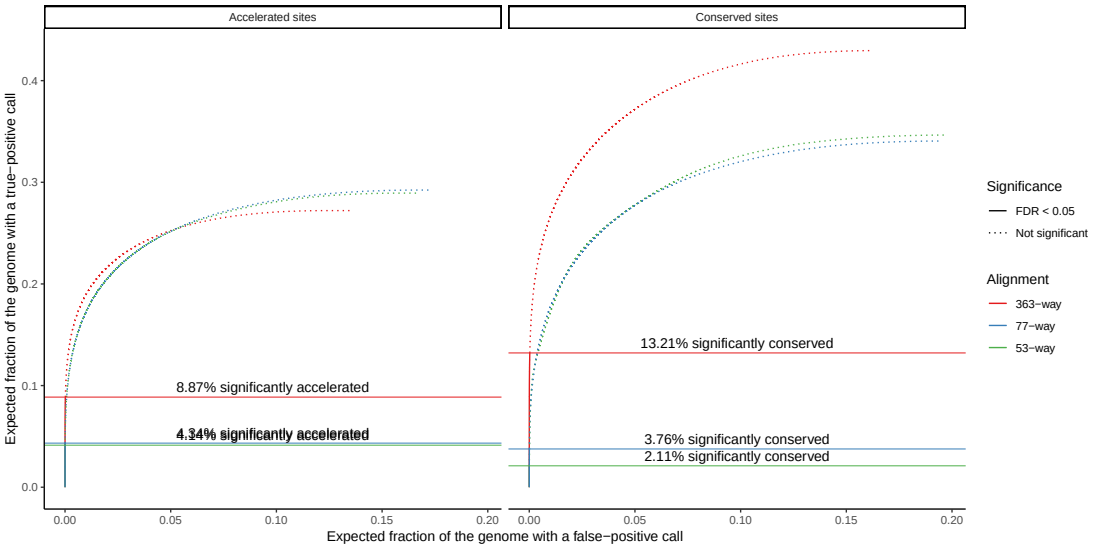


Figure C.1: Acceleration (left) and conservation (right) within B10K alignment columns on chicken compared to the 77-way. Similar to Figure 4.6, but includes accelerated columns.

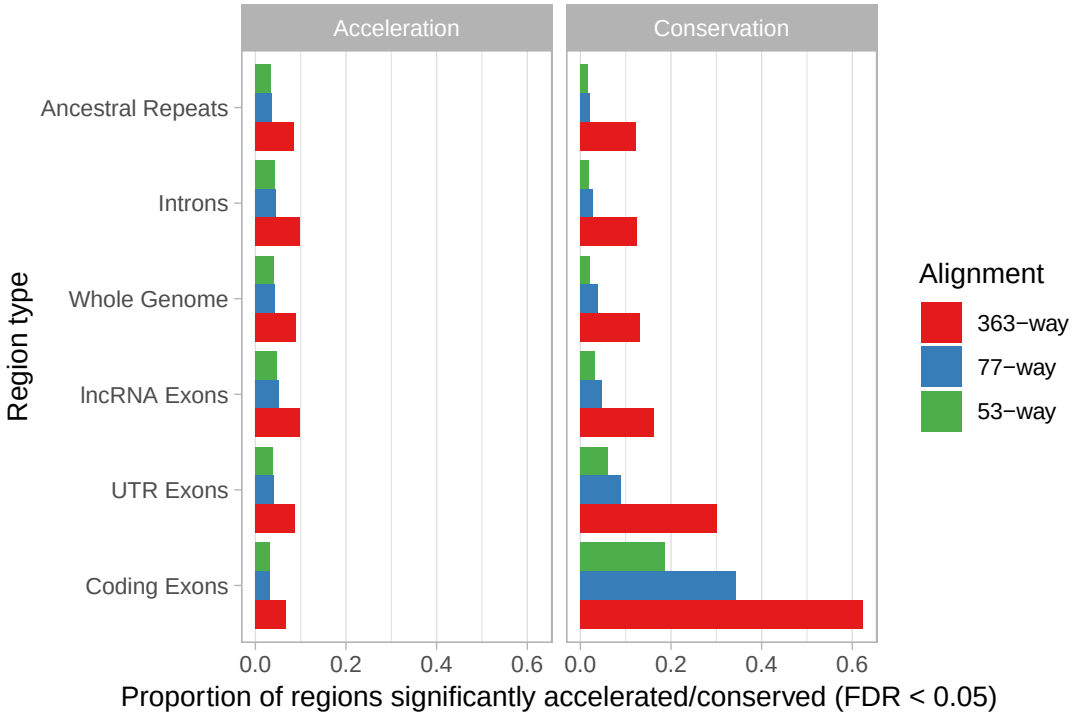


Figure C.2: Proportion of chicken functional regions covered by significantly accelerated/conserved sites. Similar to Figure 4.7, but includes accelerated columns.

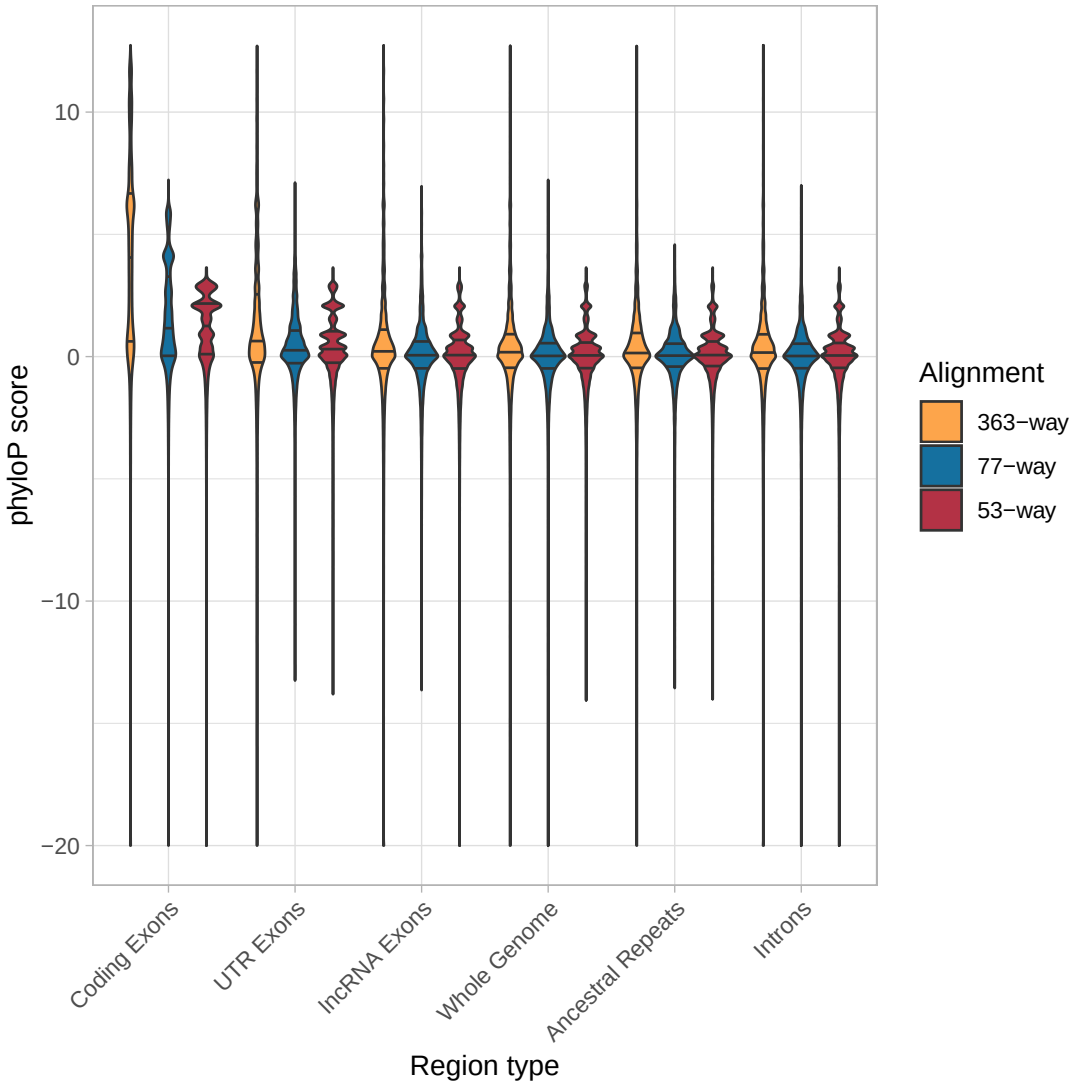


Figure C.3: Distribution of conservation / acceleration scores within different functional region types across the two alignments.

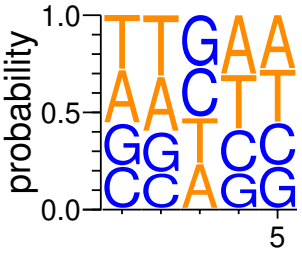


Figure C.4: Motif of 2bp on either side surrounding a random sample of conserved sites.

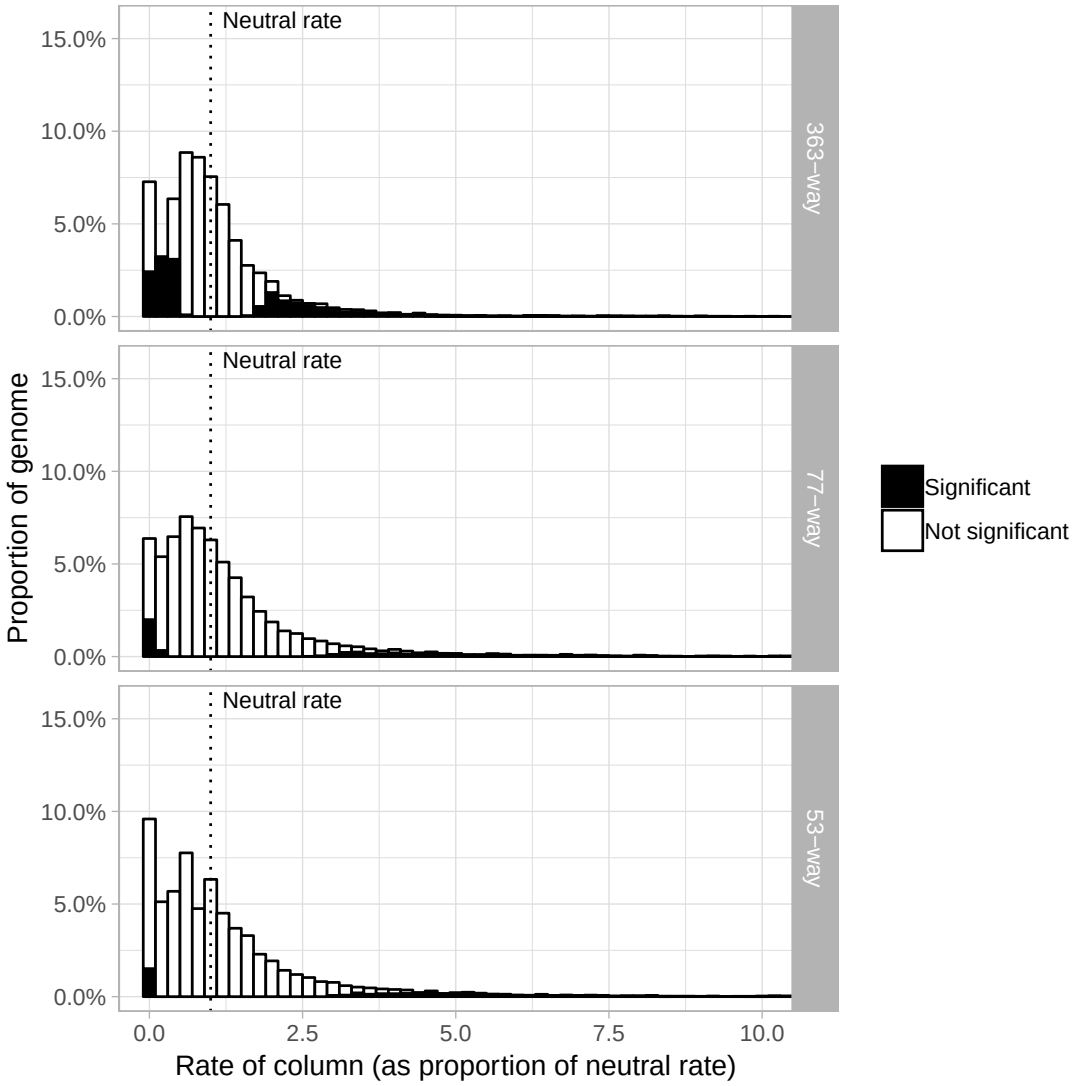


Figure C.5: Larger histogram of chicken column rates (similar to Figure 4.8, but including accelerated columns ending at a rate of 10 times the neutral rate).

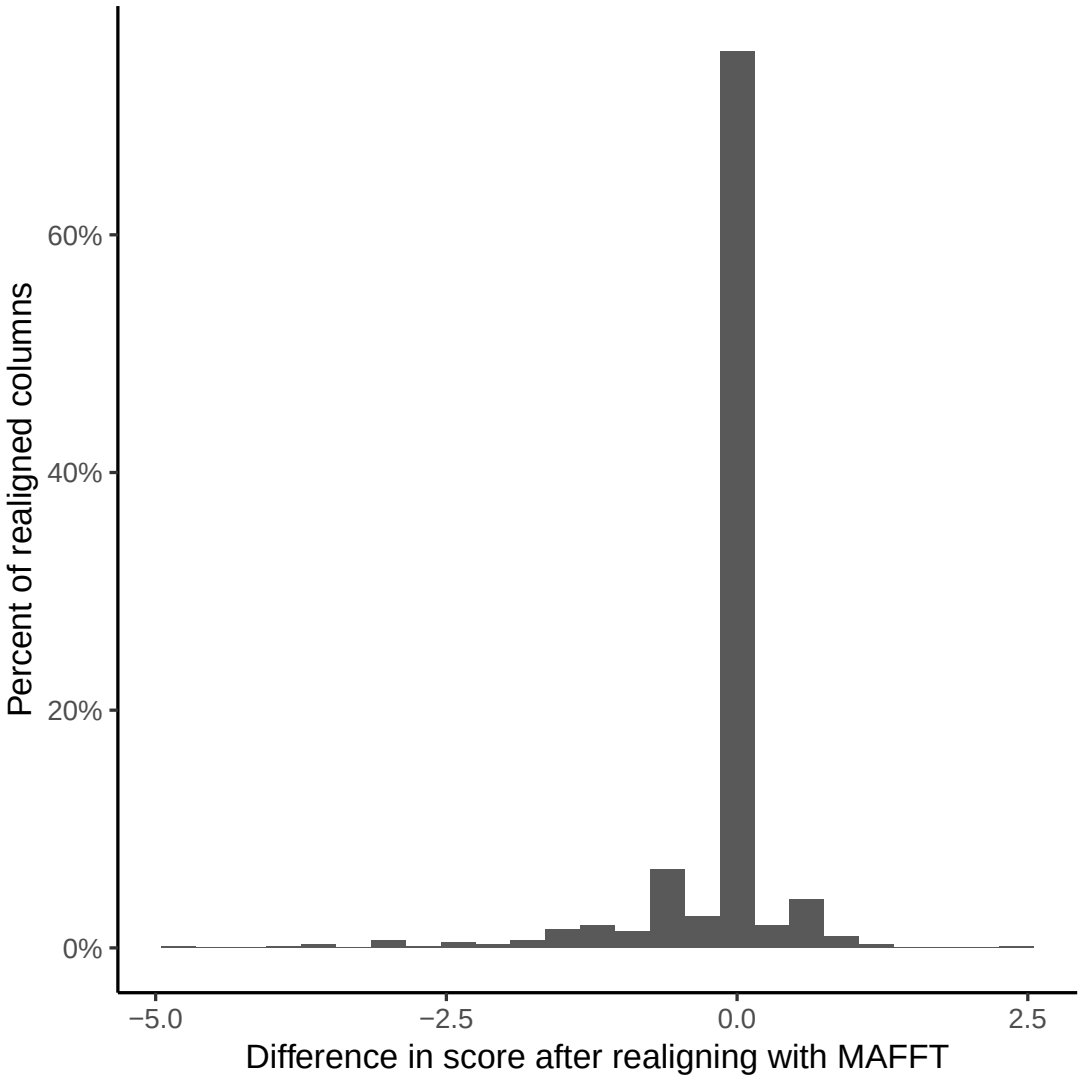


Figure C.6: Difference in phyloP scores (compared to original scores) after realignment with MAFFT for a random sample of significantly-conserved sites.

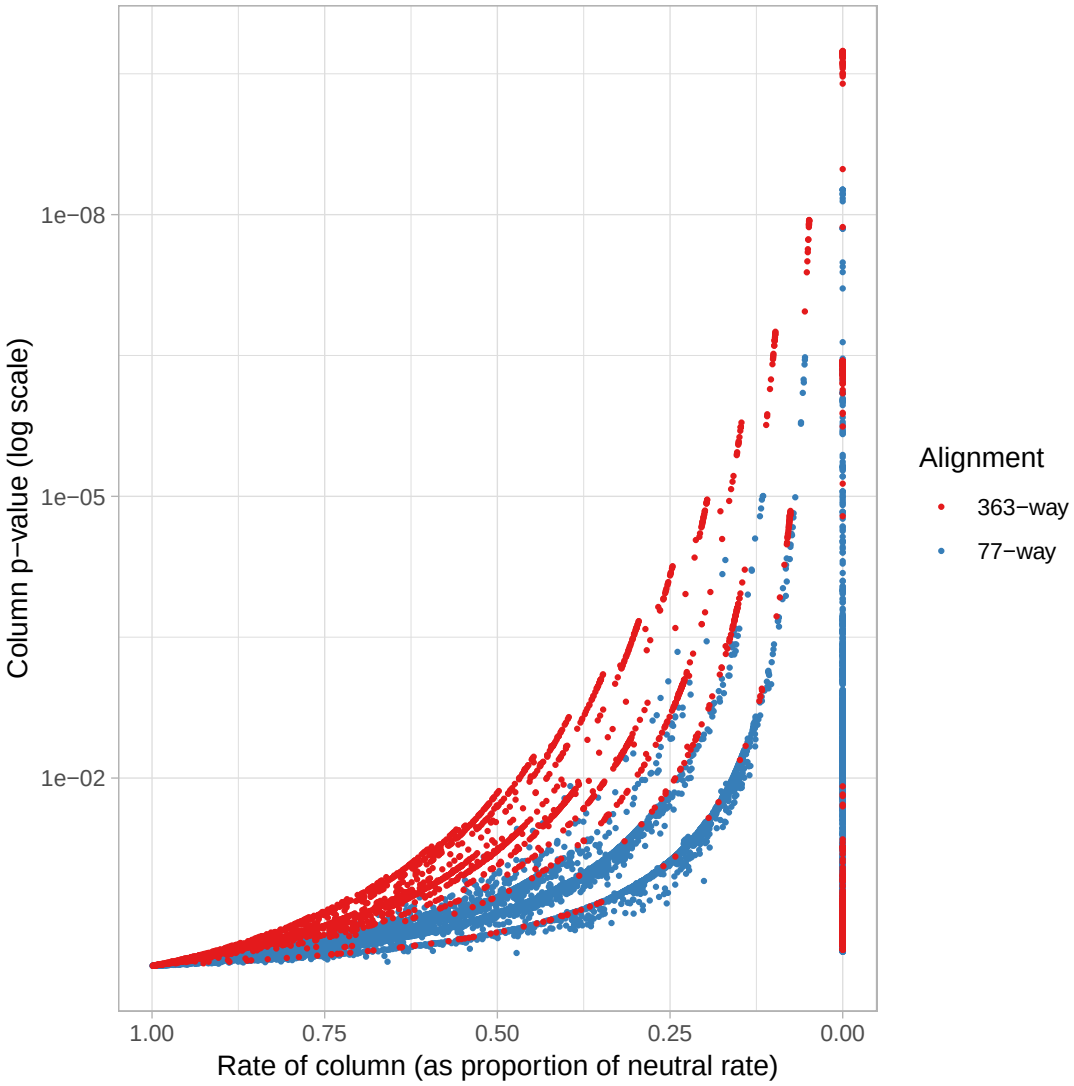


Figure C.7: Scatterplot of p-value vs. scale (rate of column relative to the neutral rate) in the B10K 363-way and the browser 77-way.

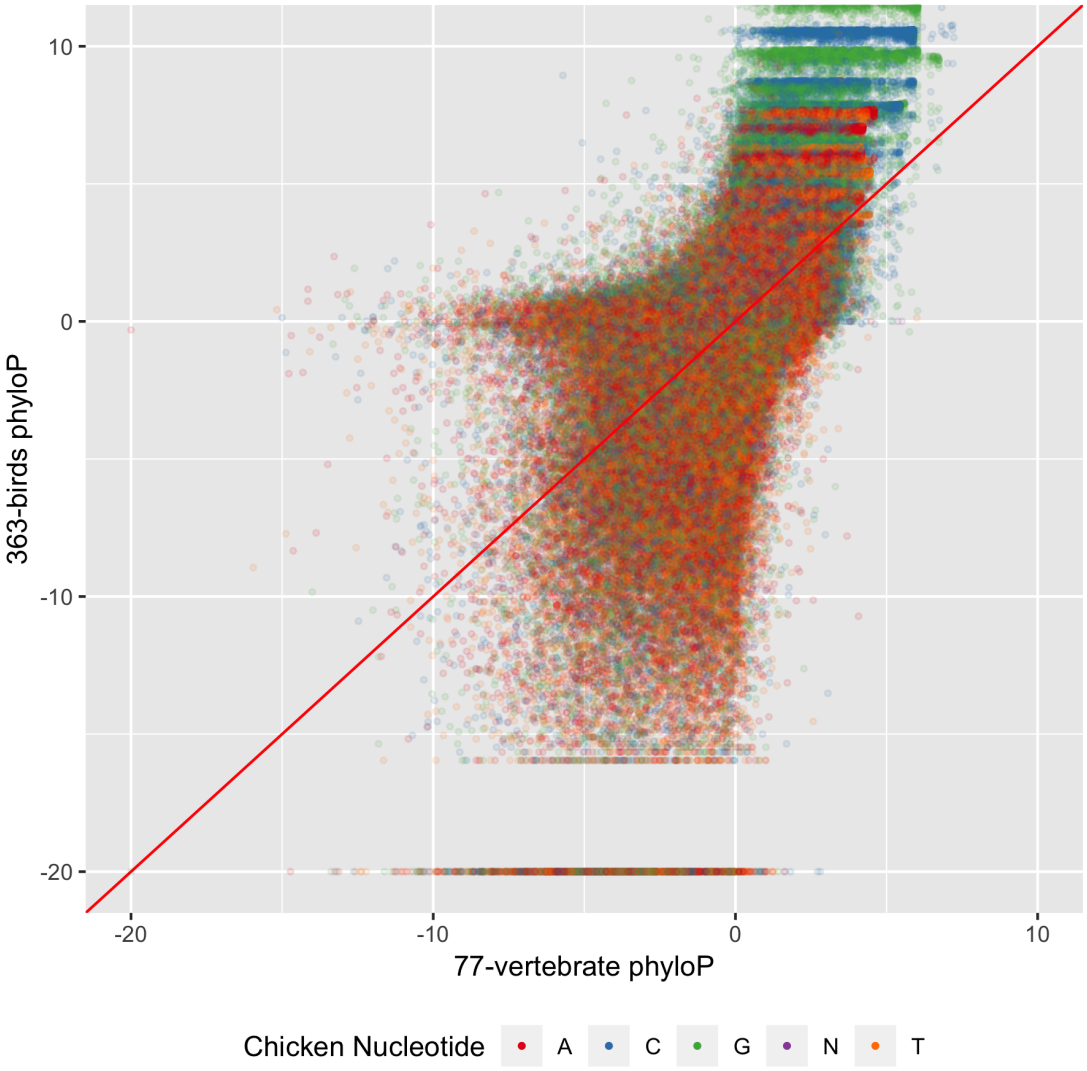


Figure C.8: Comparison of phyloP scores between the B10K 363-way and the browser 77-way on the same set of columns.

Bibliography

[1] DISCOVAR: Assemble genomes, find variants. `https://software.broadinstitute.org/software/discovar/blog/`. Accessed 2017-10-20.

[2] Bronwen L. Aken, Premanand Achuthan, Wasiu Akanni, M. Ridwan Amode, Friederike Bernsdorff, Jyothish Bhai, Konstantinos Billis, Denise Carvalho-Silva, Carla Cummins, Peter Clapham, Laurent Gil, Carlos García Girón, Leo Gordon, Thibaut Hourlier, Sarah E. Hunt, Sophie H. Janacek, Thomas Juettemann, Stephen Keenan, Matthew R. Laird, Ilias Lavidas, Thomas Maurel, William McLaren, Benjamin Moore, Daniel N. Murphy, Rishi Nag, Victoria Newman, Michael Nuhn, Chuang Kee Ong, Anne Parker, Mateus Patricio, Harpreet Singh Riat, Daniel Sheppard, Helen Sparrow, Kieron Taylor, Anja Thormann, Alessandro Vullo, Brandon Walts, Steven P. Wilder, Amonida Zadissa, Myrto Kostadima, Fergal J. Martin, Matthieu Muffato, Emily Perry, Magali Ruffier, Daniel M. Staines, Stephen J. Trevanion, Fiona Cunningham, Andrew Yates, Daniel R. Zerbino, and Paul Flicek. Ensembl 2017. *Nucleic Acids Research*, 45(D1):D635–D642, 2017.

[3] Bronwen L. Aken, Sarah Ayling, Daniel Barrell, Laura Clarke, Valery Curwen, Susan Fairley, Julio Fernandez Banet, Konstantinos Billis, Carlos García Girón, Thibaut Hourlier, Kevin Howe, Andreas Kähäri, Felix Kokocinski, Fergal J. Martin, Daniel N. Murphy, Rishi Nag, Magali Ruffier, Michael Schuster, Y. Amy Tang, Jan-Hinnerk Vogel, Simon White, Amonida Zadissa, Paul Flicek, and Stephen M. J. Searle. The ensembl gene annotation system. *Database*, 2016, 2016.

[4] Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[5] Samuel V. Angiuoli and Steven L. Salzberg. Mugsy: Fast multiple alignment of closely related whole genomes. *Bioinformatics*, 27(3):334–342, 2011.

[6] J. Armstrong, I. T. Fiddes, M. Diekhans, and B. Paten. Whole-Genome Alignment and Comparative Annotation. *Annu Rev Anim Biosci*, Oct 2018.

[7] E. Axelsson, M. T. Webster, N. G. Smith, D. W. Burt, and H. Ellegren. Comparison of the chicken and turkey genomes reveals a higher rate of nucleotide divergence on microchromosomes than macrochromosomes. *Genome Res.*, 15(1):120–125, Jan 2005.

[8] W. Bao, K. K. Kojima, and O. Kohany. Repbase Update, a database of repetitive elements in eukaryotic genomes. *Mob DNA*, 6:11, 2015.

[9] S Batzoglou. The many faces of sequence alignment. *Brief Bioinform*, 6(1):6–22, 2005.

[10] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[11] Mathieu Blanchette, W. James Kent, Cathy Riemer, Laura Elnitski, Arian F A Smith, Krishna M. Roskin, Robert Baertsch, Kate Rosenbloom, Hiram Clawson, Eric D. Green, David Haussler, and Webb Miller. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Research*, 14(4):708–715, 2004.

[12] Nick Bray, Inna Dubchak, and Lior Pachter. AVID: A global alignment program. *Genome research*, 13(1):97–102, 2003.

[13] J. W. Brown, N. Wang, and S. A. Smith. The development of scientific consensus: Analyzing conflict and concordance among avian phylogenies. *Mol. Phylogenet. Evol.*, 116:69–77, 11 2017.

[14] Michael Brudno, Chuong B. Do, Gregory M. Cooper, Michael F. Kim, Eugene Davydov, Eric D. Green, Arend Sidow, and Serafim Batzoglou. LAGAN and Multi-LAGAN: Efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, 13(4):721–731, 2003.

[15] Michael Brudno, Sanket Malde, Alexander Poliakov, Chuong B. Do, Olivier Couronne, Inna Dubchak, and Serafim Batzoglou. Glocal alignment: Finding rearrangements during alignment. *Bioinformatics*, 19(SUPPL. 1), 2003.

[16] Michael Brudno and Burkhard Morgenstern. Fast and sensitive alignment of large genomic sequences. *Proceedings / IEEE Computer Society Bioinformatics Conference. IEEE Computer Society Bioinformatics Conference*, 1:138–147, 2002.

[17] B. Charlesworth, J. A. Coyne, and N. H. Barton. The relative rates of evolution of sex chromosomes and autosomes. *The American Naturalist*, 130(1):113–146, 1987.

[18] Cedric Chauve, Nadia El-Mabrouk, Laurent Guéguen, Magali Semeria, and Eric Tannier. Duplication, Rearrangement and Reconciliation: A Follow-Up 13 Years Later. *Models and Algorithms for Genome Evolution*, pages 47–62, 2013.

[19] Jian-Qun Chen, Ying Wu, Haiwang Yang, Joy Bergelson, Martin Kreitman, and Dacheng Tian. Variation in the Ratio of Nucleotide Substitution and Indel Rates across Genomes in Mammals and Bacteria. *Molecular Biology and Evolution*, 26(7):1523–1531, 03 2009.

[20] L. Chen, Q. Qiu, Y. Jiang, K. Wang, Z. Lin, Z. Li, F. Bibi, Y. Yang, J. Wang, W. Nie, W. Su, G. Liu, Q. Li, W. Fu, X. Pan, C. Liu, J. Yang, C. Zhang, Y. Yin, Y. Wang, Y. Zhao, C. Zhang, Z. Wang, Y. Qin, W. Liu, B. Wang, Y. Ren, R. Zhang, Y. Zeng, R. R. da Fonseca, B. Wei, R. Li, W. Wan, R. Zhao, W. Zhu, Y. Wang, S. Duan, Y. Gao, Y. E. Zhang, C. Chen, C. Hvilsom, C. W. Epps, L. G. Chemnick, Y. Dong, S. Mirarab, H. R. Siegismund, O. A.

Ryder, M. T. P. Gilbert, H. A. Lewin, G. Zhang, R. Heller, and W. Wang. Large-scale ruminant genome sequencing provides insights into their evolution and distinct traits. *Science*, 364(6446), 06 2019.

[21] G. M. Cooper, M. Brudno, E. D. Green, S. Batzoglou, and A. Sidow. Quantitative estimates of sequence divergence for comparative analyses of mammalian genomes. *Genome Res.*, 13(5):813–820, May 2003.

[22] Aaron C E Darling, Bob Mau, Frederick R Blattner, and Nicole T Perna. Mauve : Multiple Alignment of Conserved Genomic Sequence With Rearrangements. *Genome Research*, pages 1394–1403, 2004.

[23] Aaron E. Darling, Bob Mau, and Nicole T. Perna. Progressivemauve: Multiple genome alignment with gene gain, loss and rearrangement. *PLoS ONE*, 5(6), 2010.

[24] Colin N. Dewey. Positional orthology: Putting genomic evolutionary relationships into context. *Briefings in Bioinformatics*, 12(5):401–412, 2011.

[25] N. Dierckxsens, P. Mardulyn, and G. Smits. NOVOPlasty: de novo assembly of organelle genomes from whole genome data. *Nucleic Acids Res.*, 45(4):e18, 02 2017.

[26] P. Dobrynin, S. Liu, G. Tamazian, Z. Xiong, A. A. Yurchenko, K. Krasheninnikova, S. Kliver, A. Schmidt-Kuntzel, K. P. Koepfli, W. Johnson, L. F. Kuderna, R. Garcia-Perez, M. d. Manuel, R. Godinez, A. Komissarov, A. Makunin, V. Brukhin, W. Qiu, L. Zhou, F. Li, J. Yi, C. Driscoll, A. Antunes, T. K. Oleksyk, E. Eizirik, P. Perelman, M. Roelke, D. Wildt, M. Diekhans, T. Marques-Bonet, L. Marker, J. Bhak, J. Wang, G. Zhang, and S. J. O'Brien. Genomic legacy of the African cheetah, Acinonyx jubatus. *Genome Biol.*, 16:277, Dec 2015.

[27] J. A. Drake, C. Bird, J. Nemesh, D. J. Thomas, C. Newton-Cheh, A. Reymond, L. Excoffier, H. Attar, S. E. Antonarakis, E. T. Dermitzakis, and J. N. Hirschhorn. Conserved noncoding sequences are selectively constrained and not mutation cold spots. *Nat. Genet.*, 38(2):223–227, Feb 2006.

[28] Inna Dubchak, Alexander Poliakov, Andrey Kislyuk, and Michael Brudno. Multiple whole-genome alignments without a reference organism. *Genome Research*, 19(4):682–689, 2009.

[29] Dent Earl, Ngan Nguyen, Glenn Hickey, Robert S Harris, Stephen Fitzgerald, Kathryn Beal, Igor Seledtsov, Vladimir Molodtsov, Brian J Raney, Hiram Clawson, et al. Alignathon: a competitive assessment of whole-genome alignment methods. *Genome research*, 24(12):2077–2089, 2014.

[30] R. C. Edgar, G. Asimenos, S. Batzoglou, and A. Sidow. Evolver: a whole-genome sequence evolution simulator. `https://www.drive5.com/evolver`.

[31] J. Eid, A. Fehr, J. Gray, K. Luong, J. Lyle, G. Otto, P. Peluso, D. Rank, P. Baybayan, B. Bettman, A. Bibillo, K. Bjornson, B. Chaudhuri, F. Christians, R. Cicero, S. Clark, R. Dalal, A. Dewinter, J. Dixon, M. Foquet, A. Gaertner, P. Hardenbol, C. Heiner, K. Hester, D. Holden, G. Kearns, X. Kong, R. Kuse, Y. Lacroix, S. Lin, P. Lundquist, C. Ma, P. Marks, M. Maxham, D. Murphy, I. Park, T. Pham, M. Phillips, J. Roy, R. Sebra, G. Shen, J. Sorenson, A. Tomaney, K. Travers, M. Trulson, J. Vieceli, J. Wegener, D. Wu, A. Yang, D. Zaccarin, P. Zhao, F. Zhong, J. Korlach, and S. Turner. Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–138, Jan 2009.

[32] J. D. Evans, S. J. Brown, K. J. Hackett, G. Robinson, S. Richards, D. Lawson, C. Elsik, J. Coddington, O. Edwards, S. Emrich, T. Gabaldon, M. Goldsmith, G. Hanes, B. Misof, M. Munoz-Torres, O. Niehuis, A. Papanicolaou, M. Pfrender, M. Poelchau, M. Purcell-Miramontes, H. M. Robertson, O. Ryder, D. Tagu, T. Torres, E. Zdobnov, G. Zhang, and X. Zhou. The i5K Initiative: advancing arthropod genomics for knowledge, human health, agriculture, and the environment. *J. Hered.*, 104(5):595–600, 2013.

[33] S. Fang, L. Zhang, J. Guo, Y. Niu, Y. Wu, H. Li, L. Zhao, X. Li, X. Teng, X. Sun, L. Sun, M. Q. Zhang, R. Chen, and Y. Zhao. NONCODEV5: a comprehensive annotation database for long non-coding RNAs. *Nucleic Acids Res.*, 46(D1):D308–D314, Jan 2018.

[34] Joseph Felsenstein. Maximum Likelihood and Minimum-Steps Methods for Estimating Evolutionary Trees from Data on Discrete Characters. *Systematic Biology*, 22(3):240–249, 09 1973.

[35] Joseph Felsenstein. PHYLIP: Phylogeny Inference Package (Version 3.2). *Cladistics*, 5:164–166, 1989.

[36] D F Feng and R F Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of molecular evolution*, 25(4):351–360, 1987.

[37] Ian T Fiddes, Joel Armstrong, Mark Diekhans, Stefanie Nachtweide, Zev N Kronenberg, Jason G Underwood, David Gordon, Dent Earl, Thomas Keane, Evan E Eichler, et al. Comparative annotation toolkit (cat)-simultaneous clade and personal genome annotation. *bioRxiv*, page 231118, 2017.

[38] WM Fitch. Rate of change of concomitantly variable codons. *Journal of Molecular Evolution*, 1(1):84–96, 1971.

[39] Adam Frankish, Mark Diekhans, Anne-Maud Ferreira, Rory Johnson, Irwin Jungreis, Jane Loveland, Jonathan M Mudge, Cristina Sisu, James Wright, Joel Armstrong, et al. Gencode reference annotation for the human and mouse genomes. *Nucleic acids research*, 47(D1):D766–D773, 2018.

[40] Z. Fu, X. Chen, V. Vacic, P. Nan, Y. Zhong, and T. Jiang. MSOAR: a high-throughput ortholog assignment system based on genome rearrangement. *J. Comput. Biol.*, 14(9):1160–1175, Nov 2007.

[41] Manuel Garber, Mitchell Guttman, Michele Clamp, Michael C. Zody, Nir Friedman, and Xiaohui Xie. Identifying novel constrained elements by exploiting biased substitution patterns. *Bioinformatics*, 25(12):i54–i62, 05 2009.

[42] E. Garrison, J. Siren, A. M. Novak, G. Hickey, J. M. Eizenga, E. T. Dawson, W. Jones, S. Garg, C. Markello, M. F. Lin, B. Paten, and R. Durbin. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.*, 36(9):875–879, 10 2018.

[43] D. Genereux, J. Johnson, V. Marinescu, E. Murén, J. Armstrong, A. S. Armero, D. Juan, G. Bejerano, N. Casewell, L. Chemnick, J. Damas, F. de Palma, M. Diekhans, I. Fiddes, M. Garber, L. Goodman, W. Haerty, M. Houck, R. Hubley, T. Kivioja, L. Kuderna, E. Lander, Marques-Bonet T., J. Meadows, W. Murphy, W. Nash, H. J. Noh, M. Nweeia, B. Paten, A. Pfenning, K. Pollard, D. Ray, B. Shapiro, A. Smit, M. Springer, C. Steiner, R. Swofford, J. Taipale, E. Teeling, J. Turner-Maier, K. Lewin, J. Alfoldi, O. Ryder, B. Birren, and K. Lindblad-Toh. Genomics in an age of extinction. in submission.

[44] David Gordon, John Huddleston, Mark JP Chaisson, Christopher M Hill, Zev N Kronenberg, Katherine M Munson, Maika Malig, Archana Raja, Ian Fiddes, LaDeana W Hillier, et al. Long-read sequence assembly of the gorilla genome. *Science*, 352(6281):aae0344, 2016.

[45] R. E. Green, E. L. Braun, J. Armstrong, D. Earl, N. Nguyen, G. Hickey, M. W. Vandewege, J. A. St. John, S. Capella-Gutierrez, T. A. Castoe, C. Kern, M. K. Fujita, J. C. Opazo, J. Jurka, K. K. Kojima, J. Caballero, R. M. Hubley, A. F. Smit, R. N. Platt, C. A. Lavoie, M. P. Ramakodi, J. W. Finger, A. Suh, S. R. Isberg, L. Miles, A. Y. Chong, W. Jaratlerdsiri, J. Gongora, C. Moran, A. Iriarte, J. McCormack, S. C. Burgess, S. V. Edwards, E. Lyons, C. Williams, M. Breen, J. T. Howard, C. R. Gresham, D. G. Peterson, J. Schmitz, D. D. Pollock, D. Haussler, E. W. Triplett, G. Zhang, N. Irie, E. D. Jarvis, C. A. Brochu, C. J. Schmidt, F. M. McCarthy, B. C. Faircloth, F. G. Hoffmann, T. C. Glenn, T. Gabaldon, B. Paten, and D. A. Ray. Three crocodilian genomes reveal ancestral patterns of evolution among archosaurs. *Science*, 346(6215):1254449–1254449, 2014.

[46] M. Haeussler, A. S. Zweig, C. Tyner, M. L. Speir, K. R. Rosenbloom, B. J. Raney, C. M. Lee, B. T. Lee, A. S. Hinrichs, J. N. Gonzalez, D. Gibson, M. Diekhans, H. Clawson, J. Casper, G. P. Barber, D. Haussler, R. M. Kuhn, and W. J. Kent. The UCSC Genome Browser database: 2019 update. *Nucleic Acids Res.*, 47(D1):D853–D858, Jan 2019.

[47] M. V. Han and M. W. Hahn. Identifying parent-daughter relationships among duplicated genes. *Pac Symp Biocomput*, pages 114–125, 2009.

[48] R.S. Harris. *Improved pairwise alignment of genomic DNA*. PhD thesis, The Pennsylvania State University, 2007.

[49] G. Hickey, B. Paten, D. Earl, D. Zerbino, and D. Haussler. HAL: a hierarchical format for storing and analyzing multiple genome alignments. *Bioinformatics*, 29(10):1341–1342, 2013.

[50] Melissa J. Hubisz, Katherine S. Pollard, and Adam Siepel. PHAST and RPHAST: Phylogenetic analysis with space/time models. *Briefings in Bioinformatics*, 12(1):41–51, 2011.

[51] S. Huntley, D. M. Baggott, A. T. Hamilton, M. Tran-Gyamfi, S. Yang, J. Kim, L. Gordon, E. Branscomb, and L. Stubbs. A comprehensive catalog of human KRAB-associated zinc finger genes: insights into the evolutionary history of a large family of transcriptional repressors. *Genome Res.*, 16(5):669–677, May 2006.

[52] Chirag Jain, Sergey Koren, Alexander Dilthey, Adam M Phillippy, and Srinivas Aluru. A fast adaptive algorithm for computing whole-genome homology maps. *Bioinformatics*, 34(17):i748–i756, 09 2018.

[53] M. Jain, S. Koren, K. H. Miga, J. Quick, A. C. Rand, T. A. Sasani, J. R. Tyson, A. D. Beggs, A. T. Dilthey, I. T. Fiddes, S. Malla, H. Marriott, T. Nieto, J. O'Grady, H. E. Olsen, B. S. Pedersen, A. Rhie, H. Richardson, A. R. Quinlan, T. P. Snutch, L. Tee, B. Paten, A. M. Phillippy, J. T. Simpson, N. J. Loman, and M. Loose. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.*, 36(4):338–345, 04 2018.

[54] M. Jain, H. E. Olsen, B. Paten, and M. Akeson. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biol.*, 17(1):239, 11 2016.

[55] Erich D Jarvis, Siavash Mirarab, Andre J Aberer, Bo Li, Peter Houde, Cai Li, Simon YW Ho, Brant C Faircloth, Benoit Nabholz, Jason T Howard, et al. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

[56] T Jiang and L Wang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1:337–348, 1994.

[57] Toby Johnson. Reciprocal best hits are not a logically sufficient condition for orthology. *arXiv*, 5(2):1–8, 2007.

[58] Thomas H Jukes and Charles R Cantor. Evolution of protein molecules. *Mammalian protein metabolism*, 3(21):132, 1969.

[59] K. M. Kapheim, H. Pan, C. Li, S. L. Salzberg, D. Puiu, T. Magoc, H. M. Robertson, M. E. Hudson, A. Venkat, B. J. Fischman, A. Hernandez, M. Yandell, D. Ence, C. Holt, G. D. Yocum, W. P. Kemp, J. Bosch, R. M. Waterhouse, E. M. Zdobnov, E. Stolle, F. B. Kraus, S. Helbing, R. F. Moritz, K. M. Glastad, B. G. Hunt, M. A. Goodisman, F. Hauser, C. J. Grimmelikhuijzen, D. G. Pinheiro, F. M. Nunes, M. P. Soares, E. D. Tanaka, Z. L. Simoes, K. Hartfelder, J. D. Evans, S. M. Barribeau, R. M. Johnson, J. H. Massey, B. R. Southey, M. Hasselmann, D. Hamacher, M. Biewer, C. F. Kent, A. Zayed, C. Blatti, S. Sinha, J. S. Johnston, S. J. Hanrahan, S. D. Kocher, J. Wang, G. E. Robinson, and G. Zhang. Social evolution. Genomic signatures of evolutionary transitions from solitary to group living. *Science*, 348(6239):1139–1143, Jun 2015.

[60] Kazutaka Katoh and Daron M. Standley. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution*, 30(4):772–780, 01 2013.

[61] Birte Kehr, Kathrin Trappe, Manuel Holtgrewe, and Knut Reinert. Genome alignment with graph data structures: a comparison. *BMC bioinformatics*, 15(1):99, 2014.

[62] W James Kent. Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[63] W James Kent, Robert Baertsch, Angie Hinrichs, Webb Miller, and David Haussler. Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11484–11489, 2003.

[64] Szymon M Kiełbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin C Frith. Adaptive seeds tame genomic sequence comparison. *Genome research*, 21(3):487–493, 2011.

[65] P. A. Kitts, D. M. Church, F. Thibaud-Nissen, J. Choi, V. Hem, V. Sapojnikov, R. G. Smith, T. Tatusova, C. Xiang, A. Zherikov, M. DiCuccio, T. D. Murphy, K. D. Pruitt, and A. Kimchi. Assembly: a resource for assembled genomes at NCBI. *Nucleic Acids Res.*, 44(D1):73–80, Jan 2016.

[66] Klaus-Peter Koepfli, Benedict Paten, the Genome 10K Community of Scientists, and Stephen J. O'Brien. The genome 10k project: A way forward. *Annual Review of Animal Biosciences*, 3(1):57–111, 2015. PMID: 25689317.

[67] Stefanie König, Lars Romoth, Lizzy Gerischer, and Mario Stanke. Simultaneous gene finding in multiple genomes. *Bioinformatics*, 32(21), 2016.

[68] Eugene V. Koonin. Orthologs, Paralogs, and Evolutionary Genomics. *Annual Review of Genetics*, 39(1):309–338, 2005.

[69] Jonas Korlach, Gregory Gedman, Sarah B. Kingan, Chen-Shan Chin, Jason T. Howard, Jean-Nicolas Audet, Lindsey Cantin, and Erich D. Jarvis. De novo PacBio long-read and phased avian genome assemblies correct and add to reference genes generated with intermediate and short reads. *GigaScience*, 6(10), 08 2017.

[70] Z. N. Kronenberg, I. T. Fiddes, D. Gordon, S. Murali, S. Cantsilieris, O. S. Meyerson, J. G. Underwood, B. J. Nelson, M. J. P. Chaisson, M. L. Dougherty, K. M. Munson, A. R. Hastie, M. Diekhans, F. Hormozdiari, N. Lorusso, K. Hoekzema, R. Qiu, K. Clark, A. Raja, A. E. Welch, M. Sorensen, C. Baker, R. S. Fulton, J. Armstrong, T. A. Graves-Lindsay, A. M. Denli, E. R. Hoppe, P. Hsieh, C. M. Hill, A. W. C. Pang, J. Lee, E. T. Lam, S. K. Dutcher, F. H. Gage, W. C. Warren, J. Shendure, D. Haussler, V. A. Schneider, H. Cao, M. Ventura, R. K. Wilson, B. Paten, A. Pollen, and E. E. Eichler. High-resolution comparative analysis of great ape genomes. *Science*, 360(6393), 06 2018.

[71] Sudhir Kumar, Glen Stecher, Michael Suleski, and S. Blair Hedges. TimeTree: A Resource for Timelines, Timetrees, and Divergence Times. *Molecular Biology and Evolution*, 34(7):1812–1819, 04 2017.

[72] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. Singularity: Scientific containers for mobility of compute. *PLoS ONE*, 12(5):1–20, 2017.

[73] Harris A. Lewin, Gene E. Robinson, W. John Kress, William J. Baker, Jonathan Coddington, Keith A. Crandall, Richard Durbin, Scott V. Edwards, Félix Forest, M. Thomas P. Gilbert, Melissa M. Goldstein, Igor V. Grigoriev, Kevin J. Hackett, David Haussler, Erich D. Jarvis, Warren E. Johnson, Aristides Patrinos, Stephen Richards, Juan Carlos Castilla-Rubio, Marie-Anne van Sluys, Pamela S. Soltis, Xun Xu, Huanming Yang, and Guojie Zhang. Earth biogenome project: Sequencing life for the future of life. *Proceedings of the National Academy of Sciences*, 115(17):4325–4333, 2018.

[74] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 05 2018.

[75] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[76] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, S. Li, H. Yang, J. Wang, and J. Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.*, 20(2):265–272, Feb 2010.

[77] J. Lilue, A. G. Doran, I. T. Fiddes, M. Abrudan, J. Armstrong, R. Bennett, W. Chow, J. Collins, S. Collins, A. Czechanski, P. Danecek, M. Diekhans, D. D. Dolle, M. Dunn, R. Durbin, D. Earl, A. Ferguson-Smith, P. Flicek, J. Flint, A. Frankish, B. Fu, M. Gerstein, J. Gilbert, L. Goodstadt, J. Harrow, K. Howe, X. Ibarra-Soria, M. Kolmogorov, C. J. Lelliott, D. W. Logan, J. Loveland, C. E. Mathews, R. Mott, P. Muir, S. Nachtweide, F. C. P. Navarro, D. T. Odom, N. Park, S. Pelan, S. K. Pham, M. Quail, L. Reinholdt, L. Romoth, L. Shirley, C. Sisu, M. Sjoberg-Herrera, M. Stanke, C. Steward, M. Thomas, G. Threadgold, D. Thybert, J. Torrance, K. Wong, J. Wood, B. Yalcin, F. Yang, D. J. Adams, B. Paten, and T. M. Keane. Sixteen diverse laboratory mouse reference genomes define strain-specific haplotypes and novel functional loci. *Nat. Genet.*, 50(11):1574–1583, Nov 2018.

[78] Kerstin Lindblad-Toh, Manuel Garber, Or Zuk, Michael F. Lin, Brian J. Parker, Stefan Washietl, Pouya Kheradpour, Jason Ernst, Gregory Jordan, Evan Mauceli, Lucas D. Ward, Craig B. Lowe, Alisha K. Holloway, Michele Clamp, Sante Gnerre, Jessica Alföldi, Kathryn Beal, Jean Chang, Hiram Clawson, James Cuff, Federica Di Palma, Stephen Fitzgerald, Paul Flicek, Mitchell Guttman, Melissa J. Hubisz, David B. Jaffe, Irwin Jungreis, W. James Kent, Dennis Kostka, Marcia Lara, Andre L. Martins, Tim Massingham, Ida Moltke, Brian J. Raney, Matthew D. Rasmussen, Jim Robinson, Alexander Stark, Albert J. Vilella, Jiayu Wen, Xiaohui Xie, Michael C. Zody, Jen Baldwin, Toby Bloom, Chee Whye Chin, Dave Heiman, Robert Nicol, Chad Nusbaum, Sarah Young,

Jane Wilkinson, Kim C. Worley, Christie L. Kovar, Donna M. Muzny, Richard A. Gibbs, Andrew Cree, Huyen H. Dihn, Gerald Fowler, Shalili Jhangiani, Vandita Joshi, Sandra Lee, Lora R. Lewis, Lynne V. Nazareth, Geoffrey Okwuonu, Jireh Santibanez, Wesley C. Warren, Elaine R. Mardis, George M. Weinstock, Richard K. Wilson, Kim Delehaunty, David Dooling, Catrina Fronik, Lucinda Fulton, Bob Fulton, Tina Graves, Patrick Minx, Erica Sodergren, Ewan Birney, Elliott H. Margulies, Javier Herrero, Eric D. Green, David Haussler, Adam Siepel, Nick Goldman, Katherine S. Pollard, Jakob S. Pedersen, Eric S. Lander, and Manolis Kellis. A high-resolution map of human evolutionary constraint using 29 mammals. *Nature*, 478(7370):476–482, 2011.

[79] Philippe Lopez, Didier Casane, and Hervé Philippe. Heterotachy, an important process of protein evolution. *Molecular biology and evolution*, 19(1):1–7, 2002.

[80] Bin Ma, John Tromp, and Ming Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics (Oxford, England)*, 18(3):440–445, 2002.

[81] J. E. Mank, E. Axelsson, and H. Ellegren. Fast-X on the Z: rapid evolution of sex-linked genes in birds. *Genome Res.*, 17(5):618–624, May 2007.

[82] Guillaume Marçais, Arthur L. Delcher, Adam M. Phillippy, Rachel Coston, Steven L. Salzberg, and Aleksey Zimin. MUMmer4: A fast and versatile genome alignment system. *PLoS Computational Biology*, 14(1):1–14, 2018.

[83] Tobias Marschall, Manja Marz, Thomas Abeel, Louis Dijkstra, Bas E. Dutilh, Ali Ghaffaari, Paul Kersey, Wigard P. Kloosterman, Veli Mäkinen, Adam M. Novak, Benedict Paten, David Porubsky, Eric Rivals, Can Alkan, Jasmijn A. Baaijens, Paul I.W. De Bakker, Valentina Boeva, Raoul J.P. Bonnal, Francesca Chiaromonte, Rayan Chikhi, Francesca D. Ciccarelli, Robin Cijvat, Erwin Datema, Cornelia M. Van Duijn, Evan E. Eichler, Corinna Ernst, Eleazar Eskin, Erik Garrison, Mohammed El-Kebir, Gunnar W. Klau, Jan O. Korbel, Eric Wubbo Lameijer, Benjamin Langmead, Marcel Martin, Paul Medvedev, John C. Mu, Pieter Neerincx, Klaasjan Ouwens, Pierre Peterlongo, Nadia Pisanti, Sven Rahmann, Ben Raphael, Knut Reinert, Dick de Ridder, Jeroen de Ridder, Matthias Schlesner, Ole Schulz-Trieglaff, Ashley D. Sanders, Siavash Sheikhizadeh, Carl Shneider, Sandra Smit, Daniel Valenzuela, Jiayin Wang, Lodewyk Wessels, Ying Zhang, Victor Guryev, Fabio Vandin, Kai Ye, and Alexander Schönhuth. Computational pan-genomics: Status, promises and challenges. *Briefings in Bioinformatics*, 19(1):118–135, 2018.

[84] C. Y. McLean, P. L. Reno, A. A. Pollen, A. I. Bassan, T. D. Capellini, C. Guenther, V. B. Indjeian, X. Lim, D. B. Menke, B. T. Schaar, A. M. Wenger, G. Bejerano, and D. M. Kingsley. Human-specific loss of regulatory DNA and the evolution of human-specific traits. *Nature*, 471(7337):216–219, Mar 2011.

[85] E. Mendoza and C. Scharff. Protein-Protein Interaction Among the FoxP Family Members and their Regulation of Two Target Genes, VLDLR and CNTNAP2 in the Zebra Finch Song System. *Front Mol Neurosci*, 10:112, 2017.

[86] G. Meng, Y. Li, C. Yang, and S. Liu. MitoZ: a toolkit for animal mitochondrial genome assembly, annotation and visualization. *Nucleic Acids Res.*, 47(11):e63, Jun 2019.

[87] Siavash Mirarab, Md. Shamsuzzoha Bayzid, Bastien Boussau, and Tandy Warnow. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215), 2014.

[88] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[89] Ngan Nguyen, Glenn Hickey, Brian J. Raney, Joel Armstrong, Hiram Clawson, Ann Zweig, Donna Karolchik, William James Kent, David Haussler, and Benedict Paten. Comparative assembly hubs: Web-accessible browsers for comparative genomics. *Bioinformatics*, 30(23):3293–3301, 2014.

[90] Ngan Nguyen, Glenn Hickey, Daniel R Zerbino, Brian Raney, Dent Earl, Joel Armstrong, W James Kent, David Haussler, and Benedict Paten. Building a pan-genome reference for a population. *Journal of computational biology : a journal of computational molecular cell biology*, 22(5):387–401, 2015.

[91] R. A. Notebaart, M. A. Huynen, B. Teusink, R. J. Siezen, and B. Snel. Correlation between sequence conservation and the genomic context after gene duplication. *Nucleic Acids Res.*, 33(19):6164–6171, 2005.

[92] C Notredame, DG Higgins, and Jaap Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–217, 2000.

[93] B. D. Ondov, T. J. Treangen, P. Melsted, A. B. Mallonee, N. H. Bergman, S. Koren, and A. M. Phillippy. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.*, 17(1):132, 06 2016.

[94] Emmanuel Paradis, Julien Claude, and Korbinian Strimmer. APE: Analyses of Phylogenetics and Evolution in R language. *Bioinformatics*, 20(2):289–290, 01 2004.

[95] B. Paten, M. Diekhans, D. Earl, J. S. John, J. Ma, B. Suh, and D. Haussler. Cactus graphs for genome comparisons. *J. Comput. Biol.*, 18(3):469–481, Mar 2011.

[96] B. Paten, D. Earl, N. Nguyen, M. Diekhans, D. Zerbino, and D. Haussler. Cactus: Algorithms for genome multiple sequence alignment. *Genome Res.*, 21(9):1512–1528, Sep 2011.

[97] Benedict Paten, Javier Herrero, Kathryn Beal, Stephen Fitzgerald, and Ewan Birney. Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome research*, 18(11):1814–28, 2008.

[98] Benedict Paten, Javier Herrero, Stephen Fitzgerald, Kathryn Beal, Paul Flicek, Ian Holmes, and Ewan Birney. Genome-wide nucleotide-level mammalian ancestor reconstruction. *Genome Research*, 18(11):1829–1843, 2008.

[99] Benedict Paten, Adam M. Novak, Jordan M. Eizenga, and Erik Garrison. Genome graphs and the evolution of genome inference. *Genome Research*, 27(5):665–676, 2017.

[100] Paul A. Pevzner, Haixu Tang, and Glenn Tesler. De novo repeat classification and fragment assembly. *Genome Research*, 14(9):1786–1796, 2004.

[101] K. S. Pollard, M. J. Hubisz, K. R. Rosenbloom, and A. Siepel. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.*, 20(1):110–121, Jan 2010.

[102] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2–approximately maximum-likelihood trees for large alignments. *PLoS ONE*, 5(3):e9490, Mar 2010.

[103] R. O. Prum, J. S. Berv, A. Dornburg, D. J. Field, J. P. Townsend, E. M. Lemmon, and A. R. Lemmon. A comprehensive phylogeny of birds (Aves) using targeted next-generation DNA sequencing. *Nature*, 526(7574):569–573, Oct 2015.

[104] Benjamin Raphael, Degui Zhi, Haixu Tang, and Pavel Pevzner. A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Research*, 14(11):2336–2346, 2004.

[105] S. Reddy, R. T. Kimball, A. Pandey, P. A. Hosner, M. J. Braun, S. J. Hackett, K. L. Han, J. Harshman, C. J. Huddleston, S. Kingston, B. D. Marks, K. J. Miglia, W. S. Moore, F. H. Sheldon, C. C. Witt, T. Yuri, and E. L. Braun. Why Do Phylogenomic Data Sets Yield Conflicting Trees? Data Type Influences the Avian Tree of Life more than Taxon Sampling. *Syst. Biol.*, 66(5):857–879, Sep 2017.

[106] Liam J. Revell. phytools: an r package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution*, 3(2):217–223, 2012.

[107] E. S. Rice, S. Kohno, J. S. John, S. Pham, J. Howard, L. F. Lareau, B. L. O'Connell, G. Hickey, J. Armstrong, A. Deran, I. Fiddes, R. N. Platt, C. Gresham, F. McCarthy, C. Kern, D. Haan, T. Phan, C. Schmidt, J. R. Sanford, D. A. Ray, B. Paten, L. J. Guillette, and R. E. Green. Improved genome assembly of American alligator genome reveals conserved architecture of estrogen signaling. *Genome Res.*, 27(5):686–696, 05 2017.

[108] Gene E Robinson, Kevin J Hackett, Mary Purcell-Miramontes, Susan J Brown, Jay D Evans, Marian R Goldsmith, Daniel Lawson, Jack Okamuro, Hugh M Robertson, David J Schneider, et al. Creating a buzz about insect genomes. *Science*, 331(6023):1386–1386, 2011.

[109] C. Schusdziarra, M. Blamowska, A. Azem, and K. Hell. Methylation-controlled J-protein MCJ acts in the import of proteins into human mitochondria. *Hum. Mol. Genet.*, 22(7):1348–1357, Apr 2013.

[110] Scott Schwartz, W James Kent, Arian Smit, Zheng Zhang, Robert Baertsch, Ross C Hardison, David Haussler, and Webb Miller. Human–Mouse Alignments with BLASTZ. *Genome Research*, 13(1):103–107, 2003.

[111] Adam Siepel, Katherine S Pollard, and David Haussler. New methods for detecting lineage-specific selection. In *Annual International Conference on Research in Computational Molecular Biology*, pages 190–205. Springer, 2006.

[112] Smit, A. F. A. and Hubley, R. and Green, P. RepeatMasker Open-4.0. `http://www.repeatmasker.org`, 2013-2015.

[113] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[114] Mark S. Springer and John Gatesy. On the importance of homology in the age of phylogenomics. *Systematics and Biodiversity*, 16(3):210–228, 2018.

[115] Alexandros Stamatakis. Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

[116] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[117] John Vivian, Arjun Arkal Rao, Frank Austin Nothaft, Christopher Ketchum, Joel Armstrong, Adam Novak, Jacob Pfeil, Jake Narkizian, Alden D Deran, Audrey Musselman-Brown, et al. Toil enables reproducible, open source, big biomedical data analyses. *Nature biotechnology*, 35(4):314, 2017.

[118] Robert M Waterhouse, Mathieu Seppey, Felipe A Simão, Mosè Manni, Panagiotis Ioannidis, Guennadi Klioutchnikov, Evgenia V Kriventseva, and Evgeny M Zdobnov. BUSCO Applications from Quality Assessments to Gene Prediction and Phylogenomics. *Molecular Biology and Evolution*, 35(3):543–548, 12 2017.

[119] Neil I Weisenfeld, Vijay Kumar, Preyas Shah, Deanna Church, and David B Jaffe. Direct determination of diploid genome sequences. *bioRxiv*, 2016.

[120] D. R. Zerbino, N. Johnson, T. Juettemann, S. P. Wilder, and P. Flicek. WiggleTools: parallel processing of large collections of genome-wide datasets for visualization and statistical analysis. *Bioinformatics*, 30(7):1008–1009, Apr 2014.

[121] B. Zhang, F. Penagaricano, A. Driver, H. Chen, and H. Khatib. Differential expression of heat shock protein genes and their splice variants in bovine preimplantation embryos. *J. Dairy Sci.*, 94(8):4174–4182, Aug 2011.

[122] G. Zhang, C. Rahbek, G. R. Graves, F. Lei, E. D. Jarvis, and M. T. Gilbert. Genomics: Bird sequencing project takes off. *Nature*, 522(7554):34, Jun 2015.

[123] Guojie Zhang, Cai Li, Qiye Li, Bo Li, Denis M Larkin, Chul Lee, Jay F Storz, Agostinho Antunes, Matthew J Greenwold, Robert W Meredith, et al. Comparative genomics reveals insights into avian genome evolution and adaptation. *Science*, 346(6215):1311–1320, 2014.

[124] Christian M Zmasek and Sean R Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–828, 2001.