Computational analysis of genome evolution

By

Aaron E. Darling

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

Doctor of Philosophy

(Computational Biology)

at the

UNIVERSITY OF WISCONSIN – MADISON

2006

Abstract

The explosive growth of genome sequencing has yielded complete genome sequences of several closely related bacterial species, and efforts to sequence entire populations are underway. Through genome comparison we expect to gain insight into the selective constraints shaping the evolution of these organisms. Genome comparison also provides a framework for characterizing the rates and patterns of large-scale evolutionary events such as genomic rearrangement and lateral gene transfer which to date are poorly understood.

This document describes the development of computational methods for the identification and classification of homologous genomic sequence among a set of sequenced genomes. The homology analysis consists of four basic procedures : (1) rapid identification of segmental homology from raw genomic sequence, (2) distinguishing orthologous and xenologous segments from paralogous segments, (3) global multiple alignment of orthologous and xenologous segments, and (4) discrimination between orthology and xenology.

The success of the analysis procedure rests on previously established models of sequence and genome evolution. Genome sequences typically comprise several million or billion nucleotides, thus the scale of the data analysis poses a challenge. Several heuristic approaches for coping with large datasets have been investigated and are reported herein.

Application of the analytic techniques to the sequenced genomes of Enteric bacteria reveals striking patterns of genome evolution. Rates of genomic rearrangement appear to be highly variable in the enteric bacteria and may be linked to adaptive evolution. The analysis reveals substantial evidence for widespread homologous recombination in populations of enteric bacteria, indicating that these microbes cannot be considered as clonal populations.

Acknowledgements

Dedicated to Loren. Keep your eyes trained on the firmament.

Thanks to Bob for being BOB.

Contents

Chapter 1

Introduction

Since Zuckerkandl and Pauling first described molecules as documents of evolutionary history (Zuckerkandl and Pauling, 1965), our ability to transcode DNA sequence into computer-readable information has undergone several dramatic revolutions. Current genome sequencing technology (Margulies et al., 2005, Shendure et al., 2005) provides low-cost sequencing for microbial genomes and populations. The vast quantity of genomic information available presents us with the tantalizing possibility of using molecular information to reconstruct the evolutionary history that has led to the current state of our biosphere.

Along the path to reconstructing evolutionary history we inevitably discover new facets of the biology of modern organisms. The indelible mark of evolution lies on every organism within and around us, and that mark can be exploited to draw inference on everything from population dynamics, to mating behavior, disease, the organism's biochemistry, and the organism's environment. Grounded in an understanding of modern biology and evolutionary history, we may begin to make similar inferences on the biology of organisms that lived many thousands or millions of years ago.

Given our newfound ability to read the documents of evolutionary history, we now face the challenge of comprehending the story unfolding before us. We must ask ourselves at what scale should we attempt to understand the process of evolution. Many previous studies have elucidated the evolutionary history of one or a few individual genes, which are taken as representative of the organism. When taken out of the context of the genomes in which they reside, the inferred evolutionary history of individual genes may show mysterious patterns that are difficult to interpret. For example, when interacting proteins co-evolve, distinct genes will have intertwined evolution but such an effect may not be observed by considering only one of the two genes. Thus to study organismal evolution it seems natural to study the evolution of genomes as a whole. Of course, organisms live in the context of an environment whose conditions often have a profound impact on the biology of the organism. Thus, one might also ask whether it makes sense to study genome evolution in isolation of a corresponding study on the evolution of the environment.

Recent comparative studies of bacterial genomes have demonstrated that members of the same microbial species may harbor as much as 10-20% unique genomic content not present in other isolates of the same species (Perna et al., 2001, Tettelin et al., 2005). In some cases the novel genomic content appears to be recently acquired and specific to the environment in which the particular bacterium lives (Sullivan et al., 2006). Furthermore, bacteriophage appear to play a fundamental role in introducing and maintaining genetic diversity within bacteria (Edwards and Rohwer, 2005). If genetic content is in fact frequently environment- and niche-specific, a study of individual microbial genomes in isolation would fail to reveal the fundamental role that environment-specific phage have played in evolution.

Inference of evolutionary history through DNA sequence is a startlingly complex task. Given two or more DNA sequences that presumably descended from a common ancestor, we would like to identify the most likely ancestral sequence, and a series of events that transformed the ancestor into the presently observed sequences. Our inferences are predicated on some model of molecular change, i.e. a set of allowable mutation operations that can be used to transform one sequence into another. Given a set of mutation operations, our model then must characterize the frequency with which each type of mutation might occur. Typically, we are uncertain what model best describes the molecular evolution of any given DNA sequence, thus we must further assume that our model is wrong. Even if the chosen model fails to capture the true nature of the evolutionary process, it may nevertheless prove to be a useful model if it can make reasonably accurate predictions when faced with data whose evolution violates model assumptions.

A model of genome evolution

As genomes evolve, they undergo large scale evolutionary processes not readily observed among short gene sequences. Recombination causes frequent genome rearrangements, horizontal transfer introduces new sequences into bacterial chromosomes, and deletions remove segments of the genome. Given a set of genomes to compare, conserved regions may exist among some or all taxa, and their ordering may be shuffled among taxa.

Traditional models of sequence evolution incorporate nucleotide substitution, and insertion and deletion of small subsequences (indels). To account for genome-scale evolution, we must extend the model to include rearrangement events such as inversion, translocation, and chromosomal fusion and fission. When combined with differential gene loss, segmental duplication can also create the effect of apparent genome rearrangement. Finally the model must incorporate some notion of gene acquisition.

Given our model of genome evolution and a data set of genome sequences, we would ideally be able to derive the most likely history of mutation events under that model. Unfortunately, the complex model structure and the scale of genomic datasets preclude direct analysis. In order to draw computationally tractable inference on genome evolution, we subdivide the analytic procedure into the separate steps of genome alignment and evolutionary analysis. Subsequent chapters of this document describe methods for genome alignment and evolutionary analysis that have been developed.

The genome alignment process identifies regions of sequence that are likely to be *orthologous*. That is, an alignment identifies nucleotides which are derived from the same nucleotide in the common ancestor of one or more extant genomes. When homologous genomic segments have been acquired via lateral gene transfer, such segments are said to be *xenologous* because the common ancestor of those segments is different than the common ancestor for the clonally reproduced portion of the genome.

The genome alignment techniques described herein do not distinguish between xenologous and orthologous segments. In order to distinguish such segments, we analyze the genome alignment to identify regions whose molecular evolution is best explained by a history that includes cross-species lateral gene transfer or intraspecific recombination.

We apply our genome alignment methods to a large group of enteric bacteria. The resulting genome alignments provide a foundation for investigations into the evolution of these bacteria. Specifically, we investigate rates of intraspecific recombination and gene acquisition both within species and across species.

1.1 An overview of the following chapters

The following chapters describe new methods we have developed to address the problem of genome alignment, and also document comparative analyses of enteric bacteria enabled by the computed genome alignments. Specifically, Chapter 2 discusses previous work related to genome alignment, statistical analysis of molecular evolution, and analysis of genome evolution. Chapter 3 describes an efficient technique for identifying local-multiple alignments which can subsequently be used as genome alignment anchors. The subsequent chapter describes an efficient approach to alignment of genomic DNA conserved among a group of closely-related organisms. Chapter 5 describes an extension of the genome alignment technique presented in Chapter 4 to handle organisms which have variable genomic mutation rates and have gained or lost substantial amounts of genetic material. We then scrutinize the accuracy of the described genome alignment methods in Chapter 6, drawing comparison to other state-of-the-art methods. Chapter 7 documents a technique for partitioning genome alignments into segments with consistent phylogenetic signal, i.e. distinguishing orthologous segments from xenologous segments. Chapter 8 describes an analysis of gene gain and loss patterns among a large group of enteric bacteria, based on genome alignments computed using our newly developed methods. Finally, Chapter 9 discusses problems with current approaches to genome alignment and proposes a Bayesian model of genome evolution for which alignments and evolutionary histories could be jointly estimated.

1.1.1 Specific contributions of this thesis

- A computational method for efficient match filtration and identification of local-multiple alignments, supporting rapid homology detection in large genome sequences

- A computational method for multiple genome alignment and comparison that identify orthologous and xenologous sequence more accurately than previous methods

- Simulation-based methods to characterize the accuracy of genome alignment algorithms

- An analysis of Enterobacteria to identify functional categories of genes that tend to be exceptionally well-conserved throughout evolution

- An analysis of *E. coli* populations to identify highly variable regions and discovery of an association among genomic variability and annotated functional non-coding RNA.

- A description of a Bayesian model of genome evolution that captures the major patterns of mutation in the *Enterobacteriacae.*

Chapter 2

Related work

Evolutionary models of nucleotide substitution describe rates and patterns of substitution between a pair of sequences. The simplest model, referred to as the Jukes-Cantor model, asserts that each nucleotide in the sequence has an equal probability of mutation per unit time, and that when it mutates, it becomes one of the other three nucleotides with equal probability (Jukes and Cantor, 1969). Similar models increase in flexibility and parameterization up to the general reversible model, which uses six parameters to specify the probability of mutation between any pair of nucleotides per unit time (Felsenstein, 2004). Such models are time-reversible, in the sense that if we have nucleotide i at one end of a branch and nucleotide j at the other, the probability of changing from i to j, $P(i \rightarrow j)$, is equal to that for changing from j to i, $P(j \rightarrow i)$, assuming uniform background nucleotide frequencies. When $P(i \rightarrow j)$ and $P(j \rightarrow i)$ are unequal, the model is not reversible and it becomes easier to calculate the position of the root on the tree. The most general non-reversible model specifies probabilities for all 12 possible nucleotide substitutions (Felsenstein, 2004).

2.0.2 Sequence alignment

The basic evolutionary models give rise to scoring schemes for the vast majority of sequence alignment methods. These sequence alignment methods combine a substitution matrix composed of log-likelihood estimates of nucleotide substitution probabilities with an empirically derived penalty for introducing gaps to ultimately arrive at a scoring scheme for alignments with gaps. Early sequence alignment algorithms such as Needleman-Wunsch calculate the highest scoring alignment between a pair of globally homologous sequences under the given scoring scheme (Needleman and Wunsch, 1970). Smith-Waterman local alignment extends the basic Needleman-Wunsch approach to the case where input sequences may not be globally homologous by identifying locally high-scoring subsequences (Smith and Waterman, 1981). Both methods utilize dynamic programming to find the highest scoring alignments. Although such methods could theoretically be applied to align several sequences of arbitrary length, their dynamic programming algorithms require $O(n^G)$ calculation where n is sequence length and G is the number of genomes. As either n or G grow the amount of computation required quickly becomes intractable.

The low-cost and ready availability of genome sequencing has driven development of scalable methods to align multiple sequences of arbitrary length. Many multiple sequence aligners extend Needleman-Wunsch to *progressive alignment* (Thompson et al., 1994, Lee et al., 2002, Notredame et al., 2000), which scales $O(Gn^2)$. In the progressive alignment model, a phylogenetic tree guides an alignment procedure where the most closely related sequences are aligned first and each additional sequence is aligned to the growing multiple alignment in an order specified by its distance in the phylogenetic guide tree. A further improvement to the progressive alignment strategy is the addition of an *iterative refinement* step performed after the initial progressive alignment (Do et al., 2005, Edgar, 2004). Iterative refinement repeatedly selects arbitrary sequence(s) to remove from the alignment and re-align. Empirical studies demonstrate that iterative refinement significantly improves alignments generated by progressive alignment approaches (Wallace et al., 2005). Surprisingly, iterative refinement produces better alignments when it considers guide trees other than the topology presumed to be the 'correct' phylogeny for the input sequences (Edgar, 2004).

Progressive multiple sequence alignment methods suffer the limitation that application to long (typically $n > 100$Kbp) sequences becomes prohibitively time-consuming. Several heuristic approaches to align long sequences have been developed under the assumption that highly similar subsequences can be found quickly and are likely to be part of the correct global alignment. These local alignments are used to *anchor* a global alignment, reducing the number of possible global alignments considered during a subsequent $O(n^2)$ dynamic programming step. Some spurious local alignments are typically found due to random sequence similarity, particularly when using a sensitive local alignment method. A method for selecting alignment anchors must be employed to filter out spurious matching regions. Alignment tools such as MUMmer (Delcher et al., 1999), GLASS (Batzoglou et al., 2000), and AVID (Bray et al., 2003) align pairs of long sequences, implementing various methods to discover local alignments. Similar *multiple* sequence alignment methods for long sequences have been developed and implemented in software packages such as MAVID (Bray and Pachter, 2003), Multi-LAGAN (Brudno et al., 2003a), TBA (Blanchette et al., 2004), MGA (Hohl et al., 2002), and Auber-Gene (Szklarczyk and Heringa, 2006). All of these pairwise and multiple sequence aligners assume the input sequences are free from significant rearrangements of sequence elements, selecting a single collinear set of alignment anchors.

Long genomic sequences typically contain significant rearrangements of orthologous sequence and methods have recently been developed to align genomic sequence in the presence of rearrangements (Brudno et al., 2003b, Darling et al., 2004a, Ovcharenko et al., 2005, Blanchette et al., 2004, Treangen and Messeguer, 2006, Raphael et al., 2004). Such methods relax the assumption that alignment anchors must occur in the same order and orientation, allowing inversions and other rearrangements of anchors. Once a set of anchors has been selected, these methods typically use progressive alignment to complete a multiple alignment.

Alignment anchor selection in the presence of rearrangements is closely related to the problem of *segmental homology detection*. The segmental homology detection task is simply to identify all homologous regions of sequence among a pair of genomes. One general approach identifies regions of sequence where local alignments tend to cluster together (Pevzner and Tesler, 2003a, Hampson et al., 2005, Calabrese et al., 2003, Kurtz et al., 2004b). Such methods consider the distance between local alignments on the chromosome as an indicator of segmental homology but do not usually consider quality (score) of such local alignments or their collinearity. A second set of approaches considers alignment scores and distances between alignments in a pairwise (Haas et al., 2004) or multiple sequence setting (Abouelhoda and Ohlebusch, 2004, Bourque et al., 2004). A third approach considers local alignment quality and collinearity, but not distance between local alignments in order to accommodate differential gene content due to deletion and horizontal transfer (Darling et al., 2004a, Mau et al., 2004). Other approaches combine chromosomal distance, local alignment score, and collinearity metrics (Darling et al., 2004b, Hampson et al., 2003). None of these methods consider the series of rearrangement events that would give rise to a given segmental homology structure.

All of the alignment methods described thus far use an ad-hoc scoring penalty to determine the placement of gaps in the alignment. A second body of work assumes a more rigorous evolutionary model that includes nucleotide birth and death rates in addition to substitution rates. Methods based on such a model are referred to as "statistical" alignment methods. When considering the probability of an alignment, these methods sum over the probability of all possible evolutionary histories that could give rise to that particular alignment given a fixed phylogenetic tree. The simplest evolutionary model that considers indels is the TKF91 model, which models single nucleotide insertions and deletions with equal birth and death rates for all sites in a sequence (Thorne et al., 1991). The TKF91 model has been studied extensively and extended from pairwise alignment to alignment on arbitrary phylogenetic trees (Nielsen, 2005). Because TKF91 only models single nucleotide indels, likelihood calculations for larger indels remain skewed. A slightly more realistic model was reported in TKF92, which models indels of arbitrary length, but which may not overlap each other in the evolutionary history (Thorne et al., 1992), i.e. an inserted sequence may not subsequently have a deletion. A further model improvement, referred to as the long-indel model, allows overlapping indels and was recently presented in conjunction with an algorithm to calculate alignment likelihoods under the model (Miklòs et al., 2004). The primary hindrance to widespread adoption of statistical alignment methods has been their prohibitive computational cost. The most efficient implementations of TKF91 require $O(2^G n^G)$ time to deterministically compute the most likely alignment, while the long indel model requires $O(n^4)$ time for an approximate pairwise alignment which allows up to two overlapping indels per site (Lunter et al., 2003, Nielsen, 2005, Metzler et al., 2001, Fleissner et al., 2005, Lunter et al., 2005, Holmes and Bruno, 2001). Recent progress in this area has yielded an implementation of long-indel model alignment called Bali-Phy (Redelings and Suchard, 2005, Suchard and Redelings, 2006). Bali-Phy simultaneously estimates the alignment and phylogenetic tree, using Markov-chain Monte-Carlo to sample the joint posterior distribution of alignments and phylogenies. The model of evolution assumes that indel rates are always proportional to substitution rates, thus variability in indel or substitution rates over time would constitute model violation.

A simple and obvious extension to the basic evolutionary models considers that nucleotide substitutions and indels do not occur with equal probability at all sites in a sequence. One example are coding regions where silent third base pair substitutions appear more frequently than substitutions at other sites and frameshift-inducing indels are usually selected against. Some score-based alignment methods can account for position-specific mutation rates (Kent and Zahler, 2000, Thompson et al., 1994, Edgar, 2004), but a more general approach has been implemented using *Profile Hidden Markov Models*, which model site-specific substitution, insertion, and deletion rates at all sites (Durbin et al., 1998). Profile-HMMs require $O(n^2)$ time and space to align a sequence to a profile. Construction of the initial profile can proceed from a manually-curated multiple alignment or *de novo* using Baum-Welch training. In order to accurately estimate site-specific mutation rates and produce reasonable alignments, such methods require much more sequence data than the previously described score based methods. Because large amounts of genome sequence data have not yet become available Profile-HMM methods have not yet been extended to large genomic sequences.

One criticism of Profile-HMM methods is their ignorance of the phylogenetic relationship among sequences contributing to the profile. To address this criticism several Tree-HMM models have been proposed (Qian and Goldstein, 2003, Mitchison, 1999, Mitchison and Durbin, 1995). Given a phylogeny, such models typically place a Profile-HMM at each node of the phylogeny, assigning probabilities for transitions between each pair of Match, Insert, and Delete states along each branch. Although Tree-HMMs can model site-specific variation along a phylogeny they remain difficult to construct in a statistically sound manner, usually requiring a pre-existing multiple sequence alignment and phylogeny. Furthermore, controversy exists over the issue of 'memory' whereby an ancestral state biased toward a particular type of insertion or deletion incorrectly biases descendant states toward the same insertion or deletion (Felsenstein, 2004).

2.0.3 Phylogenetic inference

Assuming that the sequences under study are related, phylogenetic inference attempts to reconstruct a likely history of their divergence and possibly the history of mutation events that gave rise to the observed sequences. Early methods used parsimony or some distance metric over nucleotide substitutions to inform tree inference. Although these methods can be efficiently applied to a large number of sequences, parsimony tends to underestimate true phylogenetic distance, while distance-based methods don't provide a history of mutation events (Holder and Lewis, 2003).

More recently, methods based on the previously described nucleotide substitution models have gained acceptance in the form of Maximum Likelihood (ML) or Bayesian estimates of phylogeny (Holder and Lewis, 2003). Bayesian methods provide a particularly appealing route for phylogenetic inference because not only can they provide the most likely consensus tree, but can also assess the uncertainty in various tree topologies and evolutionary scenarios. Bayesian phylogenetic inference over nucleotide substitution data was pioneered by Mau et al. (1999), and has since blossomed with several further refinements and widely used implementations (Larget and Simon, 1999, Huelsenbeck and Ronquist, 2001, Drummond et al., 2006).

With advances in genome sequencing, analyses of horizontal transfer and genome rearrangement have become feasible. Early methods to analyze genome rearrangements focused on determining parsimonious inversion and translocation scenarios among pairs of sequences (Hannenhalli and Pevzner, 1995). Parsimony models of inversion were later extended to phylogenetic inference among several rearranged genomes (Tang and Moret, 2003, Bourque and Pevzner, 2002). Larget et al. (2002) pioneered a Bayesian method to infer a series of inversion events and an associated phylogeny, and recently described extensions to their method that enable efficient and reliable analysis of large data sets (Larget et al., 2004). Recently Miklos (2003) described a Bayesian inference model for inversions and transpositions between a pair of genomes, however it has yet to be extended to phylogenetic inference among multiple genomes. Recent work has yielded new models for rearrangement that include the *block interchange* operation, whereby a segment of DNA may excise from the chromosome, form a circular-intermediate, and re-insert elsewhere in the chromosome, possibly linearizing with different endpoints than the original excised segment (Yancopoulos et al., 2005, Lu et al., 2005).

2.0.4 Integrated inference methods

As previously mentioned, the steps of model selection, alignment (inference of orthology), and phylogenetic inference are interrelated in that inferences made in one step can affect inferences made in another. Numerous attempts have been made to integrate these steps into a unified methodology. Many of these methods follow the Expectation-Maximization paradigm whereby they estimate the alignment given the tree, then re-estimate the tree given the alignment. One example is MAVID, which iteratively refines tree topology (but not branch lengths) and a genome alignment (Bray and Pachter, 2003). BADGER uses Bayesian MCMC to cosample inversion phylogeny and inversion history (Larget et al., 2004). Lunter et al. (2005) describe an efficient method for cosampling protein sequence alignments and phylogenetic trees using the TKF91 model, and the aforementioned Bali-Phy method extends the cosampling to a model that includes multi-residue indels. Sampling methods have the additional advantage of assessing confidence in a particular alignment or tree topology in the form of a posterior probability for the inference.

Chapter 3

Match filtration for local-multiple alignment

3.1 Introduction

Pairwise local sequence alignment has a long and fruitful history in computational biology and new approaches continue to be proposed (Ma et al., 2002a, Brudno and Morgenstern, 2002, Noé and Kucherov, 2004, Kent, 2002, Schwartz et al., 2003, Kahveci et al., 2004). Advanced filtration methods based on spaced-seeds have greatly improved the sensitivity, specificity, and efficiency of many local alignment methods (Choi et al., 2004, Li et al., 2006, Sun and Buhler, 2005, Xu et al., 2004, Flannick and Batzoglou, 2005). Common applications of local alignment can range from orthology mapping (Li et al., 2003) to genome assembly (Jaffe et al., 2003) to information engineering tasks such as data compression (Ane and Sanderson, 2005). Recent advances in sequence data acquisition technology (Margulies et al., 2005, Shendure et al., 2005) provide low-cost sequencing and will continue to fuel the growth of molecular sequence databases. To cope with advances in data volume, corresponding advances in computational methods are necessary; thus we present an efficient method for local multiple alignment of DNA sequence.

Unlike pairwise alignment, local multiple alignment constructs a single multiple alignment for all occurrences of a motif in one or more sequences. The motif occurrences may be identical or have degeneracy in the form of mismatches and indels. As such, local multiple alignments identify the basic repeating units in one or more sequences and can serve as a basis for downstream analysis tasks such as multiple genome alignment (Darling et al., 2004a, Hohl et al., 2002, Treangen and Messeguer, 2006, Dewey and Pachter, 2006), global alignment with repeats (Sammeth et al., 2005, Sammeth and Heringa, 2006, Raphael et al., 2004), or repeat classification and analysis (Edgar and Myers, 2005). Because it identifies multiple alignments, local multiple alignment differs from traditional pairwise methods for repeat analysis which either identify repeat families *de novo* (Kurtz et al., 2000) or using a database of known repeat motifs (Jurka et al., 2005).

Previous work on local multiple alignment includes an Eulerian path approach proposed by Zhang and Waterman (2005). Their method uses a *de Bruijn* graph based on exactly matching k-mers as a filtration heuristic. Our method can be seen as a generalization of the *de Bruijn* filtration to arbitrary spaced seeds or seed families. However, our method employs a different approach to seed extension that can identify long, low-copy number repeats.

The local multiple alignment filtration method we present has been designed to efficiently process large amounts of sequence data. It may be used to quickly find conserved repetitive motifs in a single sequence, or, may be used to identify putative homology in a group of concatenated sequences. The remainder of the chapter discusses our method in the context of finding repeats in a single sequence, although the method trivially generalizes to finding repeats and putative homology in a group of concatenated sequences. Our method is not designed to detect subtle motifs such as transcription

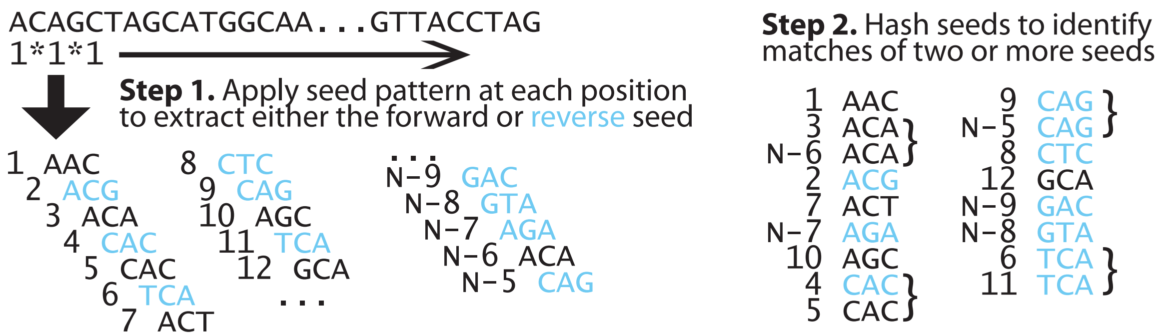

Figure 1: Application of the palindromic seed pattern `1*1*1` to identify degenerate matching subsequences in a nucleotide sequence of length N. The pattern `1*1*1` indicates a requirement for matching nucleotides at positions 1, 3, and 5 of a subsequence, while positions 2 and 4 may mismatch. The lexicographically-lesser of the forward and reverse complement subsequence induced by the seed pattern is used at each sequence position.

factor binding sites in small, targeted sequence regions—stochastic methods are better suited for such tasks (Bailey and Elkan, 1995, Siddharthan et al., 2005, Lawrence et al., 1993).

3.2 Overview of the method

Our local multiple alignment filtration method begins by generating a set of candidate multi-matches using *palindromic* spaced seed patterns (listed in Table 1). The seed pattern is evaluated at every position of the input sequence, and the lexicographically-lesser of the forward and reverse complement subsequence induced by the seed pattern is hashed to identify seed matches (Figure 1). The use of *palindromic* seed patterns offers computational savings by allowing both strands of DNA to be processed simultaneously.

Given an initial set of matching sequence regions, our algorithm then maximally extends each match to cover the entire surrounding region of sequence identity. A visual

Weight	Pattern	Seed Rank by Sequence Identity					
		65%	70%	75%	80%	85%	90%
5	11*1*11	1	1	1	1	1	1
6	1*11***11*1	1	1	1	1	1	1
7	11**1*1*1**11	1	1	1	1	1	1
8	111**1**1**111	1	1	1	1	1	1
9	111*1**1**1*111	1	1	1	1	1	1
10	111*1**1*1**1*111	1	1	1	1	1	1
11	1111**1*1*1**1111	1	1	1	1	1	2
12	1111**1*1*1*1**1111	5	3	1	1	1	1
13	1111**1**1*1*1**1**1111	> 10	5	1	1	1	1
14	1111**11*1*1*11**1111	2	2	1	1	1	1
15	1111*1*11**1**11*1*1111	1	1	1	1	1	1
16	1111*1*11**11**11*1*1111	2	1	1	1	1	1
18	11111**11*1*11*1*11**11111	1	1	1	1	1	1
19	1111*111**1*111*1**111*1111	5	2	1	1	1	1
20	11111*1*11**11*11**11*1*11111	> 10	> 10	3	1	1	1
21	11111*111*11*1*11*111*11111	1	1	1	3	3	2

Table 1: Palindromic spaced seeds used by `procrastAligner`. The sensitivity ranking of a seed at various levels of sequence identity is given in the columns at right. A seed with rank 1 is the most sensitive seed pattern for a given weight and percent sequence identity. The default seeds used by `procrastAligner` are listed here, while the additional optional seeds appear in Tables 17 and 18 of Appendix A.

example of maximal extension is given by the black match in Figure 2. In order to extend over each region of sequence $\mathcal{O}(1)$ times, our method extends matches in order of decreasing multiplicity–we extend the highest multiplicity matches first. When a match can no longer be extended without including a gap larger than w characters, our method identifies the neighboring *subset* matches within w characters, i.e. the light gray seed in Figure 2. We then *link* each neighboring subset match to the extended match. We refer to the extended match as a *superset* match. Rather than immediately extend the subset match(es), we *procrastinate* and extend the subset match later when it has the highest multiplicity of any match waiting to be extended. When extending a match

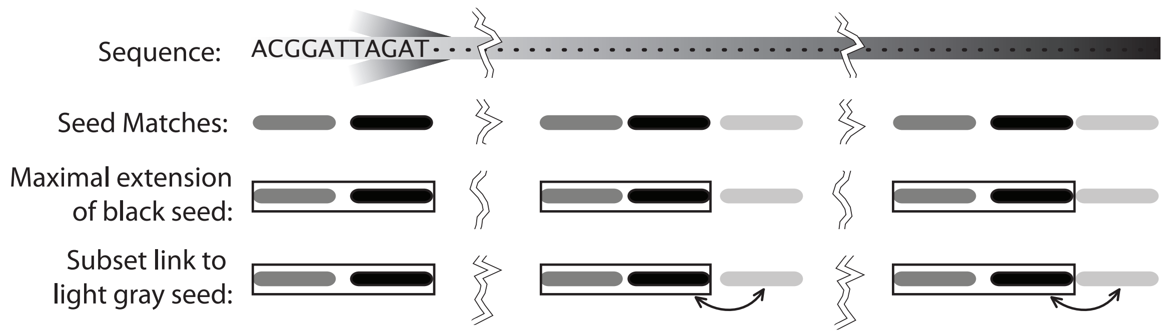

Figure 2: Seed match extension. Three seed matches are depicted as black, gray, and light gray regions of the sequence. Black and gray have multiplicity 3, while light gray has multiplicity 2. We maximally extend the black seed to the left and right and in doing so, the black seed chains with the gray seed to the left. The light gray seed is adjacent to only two out of three components in the extended black seed, thus we refer to the light gray seed as a *subset* relative to the extended black seed. We *procrastinate* and extend the light gray seed later. We create a link between light gray and the extended black seed match.

with a linked superset (light gray in Figure 2), we immediately include the entire region covered by the linked superset match–obviating the need to re-examine sequence already covered by a previous match extension.

We score alignments generated by our method using the entropy equation and exact p-value method in Nagarajan et al. (2005). Our method may produce many hundreds or thousands of local multiple alignments for a given genome sequence, thus it is important to rank them by significance. When computing column entropy, we treat gap characters as missing data.

3.3 Algorithm

3.3.1 Notation and assumptions

Given a sequence $\mathcal{S} = s_1, s_2, \ldots, s_N$ of length N defined over an alphabet $\{A, C, G, T\}$, our goal is to identify local multiple alignments on subsequences of $\mathcal{S}$. Our filtration method first generates candidate chains of ungapped alignments, which are later scored and possibly re-aligned. Denote an ungapped alignment, or match, among subsequences in $\mathcal{S}$ as an object M. We assume as input a set of ungapped alignments $\mathbf{M}$. We refer the number of regions in $\mathcal{S}$ matched by a given match $M_i \in \mathbf{M}$ as the *multiplicity* of M_i, denoted as $|M_i|$. We refer to each matching region of M_i as a *component* of M_i. Note that $|M_i| \geq 2 \; \forall \; M \in \mathbf{M}$. We denote the left-end coordinates in $\mathcal{S}$ of each component of M_i as $M_i.L_1, M_i.L_2, \ldots, M_i.L_{|M_i|}$, and similarly we denote the right-end coordinates as $M_i.R_x$. When aligning DNA sequences, matches may occur on the forward or reverse complement strands. To account for this phenomenon we add an orientation value to each matching region: $M_i.O_x \in \{1, -1\}$, where 1 indicates a forward strand match and -1 for reverse.

Our algorithm has an important limitation on the matches in $\mathbf{M}$: no two matches M_i and M_j may have the same left-end coordinate, e.g. $M_i.L_x \neq M_j.L_y \; \forall \; i, j, x, y$ except for the identity case when $i = j$ and $x = y$. This constraint has been referred to by others as *consistency* and *transitivity* (Szklarczyk and Heringa, 2004) of matches. In the present work we only require consistency and transitivity of matches longer than the seed length, e.g. seed matches may overlap.

3.3.2 Data structures

Our algorithm begins with an initialization phase that creates three data structures. The first data structure is a set of *Match Records* for each match $M \in \mathbf{M}$. The *Match Record* stores M, a unique identifier for M, and two items which will be described later in Section 3.3.3: a set of linked match records, and a *subsuming match pointer*. The linked match records are further subdivided into four classes: a left and right *superset link*, and left and right *subset links*. The *subsuming match pointer* is initially set to a *NULL* value. Figure 3 shows a schematic of the match record.

We refer to the second data structure as a *Match Position Lookup Table*, or $\mathbf{P}$. The table has N entries $p_1, p_2, \ldots, p_N$, one per character of $\mathcal{S}$. The entry for p_t stores the unique identifier of the match M_i and x for which $M_i.L_x = t$ or the *NULL* identifier if no match has t as a left-end coordinate. We call the third data structure a *Match extension procrastination queue*, or simply the *procrastination queue*. Again, we denote the multiplicity of a match M by $|M|$. The *procrastination queue* is a binary heap of matches ordered on $|M|$ with higher values of $|M|$ appearing near the top of the heap. The heap is initially populated with all $M \in \mathbf{M}$. This queue dictates the order in which matches will be considered for extension.

3.3.3 Extending matches

Armed with the three aforementioned data structures, our algorithm begins the chaining process with the match at the front of the *procrastination queue*. For a match M_i that has not been subsumed, the algorithm first attempts extension to the left, then to the right. Extension in each direction is done separately in an identical manner and we

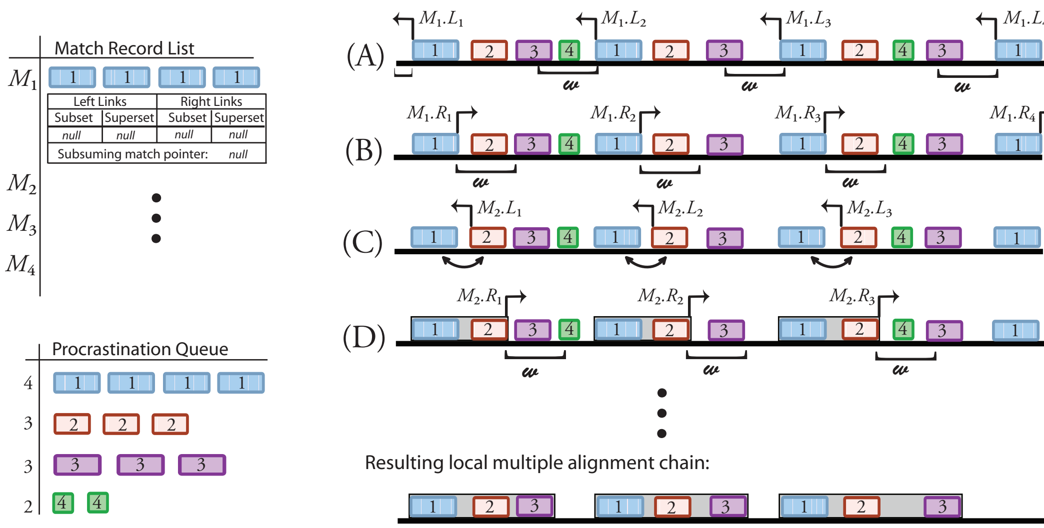

Figure 3: The match extension process and associated data structures. **(A)** First we pop the match at the front of the procrastination queue: M_1 and begin its leftward extension. Starting with the leftmost position of M_1, we use the *Match Position Lookup Table* to enumerate every match with a left-end within some distance w. Only $M_4.L_1$ is within w of M_1, so it forms a singleton *neighborhood group* which we discard. **(B)** M_1 has no *neighborhood groups* to the left, so we begin extending M_1 to the right. We enumerate all matches within w to the right of M_1. M_2 lies to the right of 3 of 4 components of M_1 and so is not subsumed, but instead gets linked as a right-subset of M_1. We add a left-superset link from M_2 to M_1. **(C)** Once finished with M_1 we pop M_2 from the front of the procrastination queue and begin leftward extension. We find the left-superset link from M_2 to M_1, so we extend the left-end coordinates of M_2 to cover M_1 accordingly. No further leftward extension of M_2 is possible because M_1 has no left-subset links. **(D)** Beginning rightward extension on M_2 we construct a neighborhood list and find a chainable match M_3, and a subset M_4. We extend M_2 to include M_3 and mark M_4 as inconsistent and hence not extendable. Upon completion of the chaining process we have generated a list of local multiple alignments.

arbitrarily choose to describe leftward extension first. The first step in leftward match extension for M_i is to check whether it has a left superset link. If so, we perform a *link extension* as described later. For extension of M_i without a superset link, we use the *Match Position Lookup Table* $\mathbf{P}$ to enumerate all matches within a fixed distance w of M_i. For each component $x = 1, 2, \ldots, |M_i|$ and distance $d = 1, 2, \ldots, w$ we evaluate first whether $p_{M_i.L_x-(d \cdot M_i.O_x)}$ is not *NULL*. If not then $p_{M_i.L_x-(d \cdot M_i.O_x)}$ stores an entry $\langle M_j, y\rangle$ which is a pointer to neighboring match M_j and the matching component y of M_j.

In order to consider matches on both forward and reverse strands, we must evaluate whether $M_i.O_x$ and $M_j.O_y$ are consistent with each other. We define the relative orientation of $M_i.O_x$ and $M_j.O_y$ as $o_{i,j,x,y} = M_i.O_x \cdot M_j.O_y$ which causes $o_{i,j,x,y} = 1$ if both $M_i.O_x$ and $M_j.O_y$ match the same strand and -1 otherwise. We create a tuple of the form $\langle M_j, o_{i,j,x,y}, x, d, y\rangle$ and add it to a list called the *neighborhood list*. In other words, the tuple stores (1) the unique match ID of the match with a left-end at sequence coordinate $M_i.L_x - (d \cdot M_i.O_x)$, (2) the relative orientation of $M_i.O_x$ and $M_j.O_y$, (3) the matching component x of M_i, (4) the distance d between M_i and M_j, and (5) the matching component y of M_j. If $M_j = M_i$ for a given value of d, we stop adding *neighborhood list* entries after processing that one. The *neighborhood list* is then scanned to identify groups of entries with the same match ID M_j and relative orientation $o_{i,j,x,y}$. We refer to such groups as *neighborhood groups*. Entries in the same *neighborhood group* that have identical x or y values are considered "ties" and need to be broken. Ties are resolved by discarding the entry with the larger value of d in the fourth tuple element: we prefer to chain over shorter distances. After tiebraking, each *neighborhood group* falls into one of several categories:

- **Superset**: The *neighborhood group* contains $|M_i|$ separate entries. M_j has higher multiplicity than M_i, e.g. $|M_j| > |M_i|$. We refer to M_j as a superset of M_i.

- **Chainable**: The *neighborhood group* contains $|M_i|$ separate entries. M_j and M_i have equal multiplicity, e.g. $|M_j| = |M_i|$. We can chain M_j and M_i.
- **Subset**: The *neighborhood group* contains $|M_j|$ separate entries such that $|M_j| < |M_i|$. We refer to M_j as a subset of M_i.
- **Novel Subset**: The *neighborhood group* contains r separate entries such that $r < |M_i| \wedge r < |M_j|$. We refer to the portion of M_j in the list as a *novel subset* of M_i and M_j because this combination of matching positions does not exist as a match in the initial set of matches $\mathbf{M}$.

The algorithm considers each *neighborhood group* for chaining in the order given above: chainable, subset, and finally, novel subset. Superset groups are ignored, as any superset links would have already been created when processing the superset match.

Chainable matches

To chain match M_i with *chainable* match M_j we first update the left-end coordinates of M_i by assigning $M_i.L_x \leftarrow \min(M_i.L_x, M_j.L_y)$ for each $\langle i, j, x, y\rangle$ in the *neighborhood group* entries. Similarly, we update the right-end coordinates: $M_i.R_x \leftarrow \max(M_i.R_x, M_j.R_y)$ for each $\langle i, j, x, y\rangle$ in the group. If any of the coordinates in M_i change we make note that a *chainable* match has been chained. We then update the *Match Record* for M_j by setting its *subsuming match pointer* to M_i, indicating that M_j is now invalid and is subsumed by M_i. Any references to M_j in the *Match Position Lookup Table* and elsewhere may be lazily updated to point to M_i as they are encountered. If M_j has a left superset link, the link is inherited by M_i and any remaining neighborhood groups with *chainable* matches are ignored. *Chainable* groups are processed in order of increasing d value so that the nearest *chainable* match with a superset link will be encountered first. A special case exists when $M_i = M_j$. This occurs when M_i represents an inverted repeat within w nucleotides. We never allow M_i to chain with itself.

Subset matches

We defer subset match processing until no more chainable matches exist in the neighborhood of M_i. A subset match M_j is considered to be completely contained by M_i when for all x, y pairs in the *neighborhood group*, $M_i.L_x \leq M_j.L_y \wedge M_j.R_y \leq M_i.R_x$. When subset match M_j is completely contained by M_i, we set the *subsuming match pointer* of M_j to M_i. If the subset match is not contained we create a *link* from M_i to M_j. The subset link is a tuple of the form $\langle M_i, M_j, x_1, x_2, \ldots, x_{|M_j|}\rangle$ where the variables $x_1 \ldots x_{|M_j|}$ are the x values associated with the $y = 1 \ldots |M_j|$ from the *neighborhood list* group entries. The link is added to the left subset links of M_i and we remove any pre-existing right superset link in M_j and replace it with the new link.

Novel subset matches

A novel subset may only be formed when both M_i and M_j have already been maximally extended, otherwise we discard any novel subset matches. When a novel subset exists matches we create a new match record M_{novel} with left- and right-ends equal to the outward boundaries of M_i and M_j. Rather than extend the novel subset match immediately, we *procrastinate* and place the novel subset in the *procrastination queue*. Recall that the novel subset match contains r matching components of M_i and M_j. In constructing M_{novel}, we create links between M_{novel} and each of M_i and M_j such that

Figure 4: Interplay between tandem repeats and novel subset matches. There are two initial seed matches, one black, one gray. The black match has components labelled 1-7, and the neighborhood size w is shown with respect to component 7. As we attempt leftward extension of the black match we discover the gray match in the neighborhood of components 2 and 5 of black. A subset link is created. We also discover that some components of the black match are within each others' neighborhood. We classify the black match as a tandem repeat and construct a novel subset match with one component for each of the four tandem repeat units: $\{1\}, \{2, 3, 4\}, \{5, 6\}, \{7\}$.

M_{novel} is a left and a right subset of M_i and M_j, respectively. The links are tuples of the form outlined in the previous section on subset matches.

Occasionally a *neighborhood group* representing a novel subset match may have $M_i = M_j$. This can occur when M_i has two or more components that form a tandem or overlapping repeat. If $M_i.L_x$ has $M_i.L_y$ in its neighborhood, and $M_i.L_y$ has $M_i.L_z$ in its neighborhood, then we refer to $\{x, y, z\}$ as a tandem unit of M_i. A given tandem unit contains between one and $|M_i|$ components of M_i, and the set of tandem units forms a partition on the components of M_i. In this situation we construct a novel subset match record with one component for each tandem unit of M_i. If M_i has only a single tandem unit then we continue without creating a novel subset match record. Figure 4 illustrates how we process tandem repeats.

After the first round of chaining

If the *neighborhood list* contained one or more chainable groups we enter another round of extending M_i. The extension process repeats starting with either *link extension* or by construction of a new *neighborhood list*. When the boundaries of M_i no longer change, we classify any subset matches as either subsumed or outside of M_i and treat them accordingly. We process novel subsets. Finally, we may begin extension in the opposite (rightward) direction. The rightward extension is accomplished in a similar manner, except that the neighborhood is constructed from $M_i.R_x$ instead of $M_i.L_x$ and d ranges from $-1, -2, \ldots, -w$ and ties are broken in favor of the largest d value. Where left links were previously used, right links are now used and vice-versa.

Chaining the next match

When the first match popped from the *procrastination queue* has been maximally extended, we pop the next match from the *procrastination queue* and consider it for extension. The process repeats until the *procrastination queue* is empty. Prior to extending any match removed from the *procrastination queue*, we check the match's *subsuming match pointer*. If the match has been subsumed extension is unnecessary.

3.3.4 Link extension

To be considered for leftward link extension, M_i must have a left superset link to another match, M_j. We first extend the boundaries of M_i to include the region covered by M_j and unlink M_i from M_j. Then each of the left subset links in M_j are examined in turn to identify links that M_i may use for further extension. Recall that the link from M_i to M_j is of the form $\langle M_j, M_i, x_1, \ldots, x_{|M_i|}\rangle$. Likewise, a left subset link from M_j to another match M_k is of the form $\langle M_j, M_k, z_1, \ldots, z_{|M_k|}\rangle$. To evaluate whether M_i may follow a given link in the left subsets of M_j, we take the set intersection of the x and z values for each M_k that is a left subset of M_j. We can classify the results of the set intersection as:

- **Superset**: $\{x_1, \ldots, x_{|M_i|}\} \subset \{z_1, \ldots, z_{|M_k|}\}$ Here M_k links to every component of M_j that is linked by M_i, in addition to others.

- **Chainable**: $\{x_1, \ldots, x_{|M_i|}\} = \{z_1, \ldots, z_{|M_k|}\}$ Here M_k links to the same set of components of M_j that M_i links.

- **Subset**: $\{x_1, \ldots, x_{|M_i|}\} \supset \{z_1, \ldots, z_{|M_k|}\}$ Here M_i links to every component of M_j that is linked by M_k, in addition to others.

- **Novel Subset**: $\{x_1, \ldots, x_{|M_i|}\} \cap \{z_1, \ldots, z_{|M_k|}\} \neq \emptyset$ Here M_k is neither a superset, chainable, nor subset relative to M_i, but the intersection of their components in M_j is non-empty. M_k and M_i form a novel subset.

Left subset links in M_j are processed in the order given above. Supersets are never observed, because M_k would have already unlinked itself from M_j when it was processed (as described momentarily). When M_k is a chainable match, we extend M_i to include the region covered by M_k and set the subsuming match pointer in M_k to point to M_i. We unlink M_k from M_j, and M_i inherits any left superset link that M_k may have. When M_k is a subset of M_i we unlink M_k from M_j and add it to the *deferred subset list* to be processed once M_i has been fully extended. Finally, we never create novel subset matches during link extension because M_k will never be a fully extended match.

If a chainable match was found during leftward link extension, we continue for another round of leftward extension. If not, we switch directions and begin rightward extension.

3.3.5 Time complexity

A *neighborhood list* may be constructed at most w times per character of $\mathcal{S}$, and construction uses sorting by key comparison, giving $\mathcal{O}(wN \log wN)$ time and space. Similarly, we spend $\mathcal{O}(wN \log wN)$ time performing link extension. The upper bound on the total number of components in the final set of matches is $\mathcal{O}(wN)$. Thus, the overall time complexity for our filtration algorithm is $\mathcal{O}(wN \log wN)$.

3.4 Results

We have created a program called `procrastAligner` for Linux, Windows, and Mac OS X that implements the described algorithm. Our open-source implementation is available as C++ source code licensed under the GPL.

We compare the performance of our method in finding Alu repeats in the human genome to an Eulerian path method for local multiple alignment (Zhang and Waterman, 2005). The focus of our algorithm is efficient filtration, thus we use a scoring metric that evaluates the filtration sensitivity and specificity of the ungapped alignment chains produced by our method. We compute sensitivity as the number of Alu elements hit by a match, out of the total number of Alu elements. We compute specificity as the ratio of match components that hit an Alu to the sum of match multiplicity for all matches that hit an Alu. Thus, we do not penalize our method for finding legitimate repeats that are not in the Alu family.

The comparison between `procrastAligner` and the Eulerian method is necessarily indirect, as each method was designed to solve different (but related) problems. The Eulerian method uses a *de Bruijn* graph for filtration, but goes beyond filtration to compute gapped alignments using banded dynamic programming. We report scores for a version of the Eulerian method that computes alignments only on regions identified by its *de Bruijn* filter. The results suggest that by using our filtration method, the sensitivity of the Eulerian path local multiple aligner could be significantly improved. A second important distinction is that our method reports *all* local multiple alignment chains in its allotted runtime, whereas the Eulerian method identifies only a single alignment.

We also test the ability of our method to provide accurate anchors for genome alignment. Using a manually curated alignment of 144 Hepatitis C virus genome sequences (Kuiken et al., 2005), we measure the anchoring sensitivity of our method as the fraction of pairwise positions aligned in the correct alignment that are also present in `procrastAligner` chains. We measure positive predictive value as the number of match component pairs that contain correctly aligned positions out of the total number of match component pairs. `procrastAligner` may generate legitimate matches in the repeat regions of a single genome. The PPV score penalizes `procrastAligner` for identifying such legitimate repeats, which subsequent genome alignment would have to disambiguate. Using a seed size of 9 and $w = 27$, `procrastAligner` has a sensitivity of 63% and PPV of 67%.

3.5 Discussion

We have described an efficient method for local multiple alignment filtration. The chains of ungapped alignments that our filter outputs may serve as direct input to multiple genome alignment algorithms. The test results of our prototype implementation on

Accession	Length	Rep	Fm	Alu (bp)	Div, %	Met	Sn %	Sp %	T (s)	Sw	w
AF435921	22 Kb	28	10	261 (69)	15.0 (6.4)	Eul	96.3	99.4	1	-	-
						pro	100	95.9	1	9	27
Z15025	38 Kb	52	13	245 (85)	15.7 (5.7)	Eul	98.6	96.7	4	-	-
						pro	100	82.5	2	9	27
AC034110	167 Kb	87	18	261 (72)	12.2 (5.9)	Eul	93.5	95.2	14	-	-
						pro	100	97.9	3	15	45
AC010145	199 Kb	118	13	277 (55)	15.0 (5.6)	Eul	85.2	93.7	32	-	-
						pro	99.1	99.2	3	15	45
Hs Chr 22	1 Mbp	404	32	252 (79)	15.2 (6.1)	Eul	72.4	99.4	85	-	-
						pro	98.3	97.3	20	15	45

Table 2: Performance of `procrastAlign` and the Eulerian path approach on Alu repeats. Rep: total number of Alu elements; Fm: number of Alu families; Alu: average Alu length in bp (S.D.); Div: average Alu divergence (S.D.); Met: alignment method, Eul = Eulerian, pro = `procrastAligner`; Sn: sensitivity; Sp: specificity; T: compute time; Sw: palindromic seed weight; w: max gap size. Alus were identified by RepeatMasker (Jurka et al., 2005). We report data for the fast version of the Eulerian path method as given by Table 1 of (Zhang and Waterman, 2005). Sensitivity and specificity of `procrastAlign` was computed as described in the text.

Alu sequences demonstrate improved sensitivity over *de Bruijn* filtration. A promising avenue of further research will be to couple our filtration method with subsequent refinement using banded dynamic programming.

The alignment scoring scheme we use can rank alignments by information content, however a biological interpretation of the score remains difficult. If a phylogeny and model of evolution for the sequences were known *a priori* then a biologically relevant scoring scheme could be used (Prakash and Tompa, 2005). Unfortunately, the phylogenetic relationship for arbitrary local alignments is rarely known, especially among repetitive elements or gene families within a single genome and across genomes. It may be possible to use simulation and MCMC methods to score alignments where the phylogeny and model of evolution is unknown *a priori*, but doing so would be computationally prohibitive for our application.

3.6 Acknowledgments

An abridged version of this chapter appeared as Darling, Treangen, Zhang, Kuiken, Messeguer, and Perna (2006). AED designed the research and implemented `procrastAligner`. TJT and AED designed the filtration and scoring algorithms and coauthored the manuscript. LZ computed optimal palindromic seed patterns.

Chapter 4

Alignment of closely-related genomes

Genome alignment is a fundamentally different task than sequence alignment. The nature of genome evolution violates basic assumptions made by traditional alignment methods, such as complete collinearity and consistency in the phylogenetic signal. To compensate, a genome alignment method must include not just sequence alignment, but a method for detecting segmental homology as well, and it must be robust to variance in the phylogenetic signal.

A second distinguishing feature of genome alignment stems from the fact that genome sequences are typically much larger than the sequences for which dynamic-programming based alignment methods were originally designed. The well-known Needleman-Wunsch algorithm to find the best global alignment of a pair of sequences requires $\mathcal{O}(N^2)$ compute time (Needleman and Wunsch, 1970). For sequences as large as 10Kbp-100Kbp modern computational hardware can compute the full score matrix and trace back the optimal alignment path. However, bacterial genome sequences typically range in size from 1Mbp to 10Mbp, while eukaryotic genomes can be anywhere between 1Mbp and several hundred gigabases in size. Computation of a full alignment score matrix using dynamic programming for such sequences is too time-consuming on modern compute hardware. Although dynamic programming approaches that exploit parallel hardware have been used with some success (Martins et al., 2001), an approach that is tractable on commodity compute hardware is strongly preferable.

To effectively trim the overall alignment search space without sacrificing alignment quality, a heuristic commonly referred to as anchored alignment (Delcher et al., 1999) or banded dynamic programming (Zhang et al., 2000) was devised. Anchored alignment typically begins by using a fast string-matching method to find high-scoring local alignments. It then restricts the computation of scores in the dynamic programming matrix to the regions around the high-scoring local alignments. Anchored alignment methods operate under the assumption that the optimal global alignment is very likely to include the high-scoring local alignments. In general, anchoring heuristics yield quality alignments in a fraction of the compute time otherwise necessary to compute an optimal alignment (Ureta-Vidal et al., 2003). As such, all modern genome alignment approaches use anchoring heuristics.

4.0.1 The Mauve algorithm

Our development of a multiple genome alignment algorithm was motivated by the recent sequencing of a group of nine enterobacteria. At the time, existing anchored alignment methods were unable to cope with the substantial amount of genomic rearrangement and lateral gene transfer that these microbes have experienced. Other aspects of the genomic biology of these microbes such as the presence of a small number of large-repetitive regions figured prominently into our algorithm design. We refer to the presently described alignment algorithm as “Mauve.”

When searching for alignment anchors across multiple genomes, problems arise if a particular repetitive motif occurs numerous times in each sequence because it becomes unclear which combination of regions to align. Our target data set of enteric genomes are known to have significant repetitive regions such as ribosomal RNA operons and prophages. For a repetitive element existing r times in each of G genomes, there will be r^G possible alignment anchors, of which at most r represent truly orthologous anchors. As more genomes are aligned, the number of possible anchors grows exponentially while the number of anchors that can be included in an alignment of orthologous sequences remains constant. Mauve avoids this problem by using approximate Multiple Maximal Unique Matches (multi-MUMs) of some minimum length k as alignment anchors. Approximate multi-MUMs are subsequences shared by two or more genomes that match according to a seed pattern. As described in the previous chapter, a seed pattern specifies a pattern of nucleotides that must match. For example `11*11*11*` would specify a seed of length 9 and weight 6 where every nucleotide except the third, sixth, and ninth must match (Ma et al., 2002b). Furthermore, at least one realization of the matching seed pattern contained in the matching subsequence must occur only once in those genomes to satisfy the uniqueness property. We refer to matches which satisfy these properties as approximate multi-MUMs because they represent unique subsequences which match each other approximately, tolerating a small amount of degeneracy. Finally, the approximate multi-MUMs must be bounded on either side by a region without any seed matches. For the sake of brevity, we will simply use multi-MUMs to refer to approximate multi-MUMs unless otherwise noted. Because using unique seeds reduces anchoring sensitivity in conserved repetitive regions and regions that have undergone numerous nucleotide substitutions or indels, Mauve employs a recursive anchoring strategy that progressively reduces k, searching for smaller anchors in the remaining unmatched regions.

The enterobacterial genomes are known to have undergone significant genome rearrangements as described in their genome papers. Algorithms used by other global multiple alignment systems anchor their alignments by selecting the highest scoring collinear chain of local alignments (Hohl et al., 2002, Bray and Pachter, 2003, Brudno et al., 2003a). Such methods preclude identification of the rearrangements known to exist in our data set and many others. To successfully align our target genomes, the anchor selection method should identify consistent (collinear) subsets of local alignments to use as anchors while filtering out unlikely local alignments. Ideally, an algorithm would identify a maximum-weight set of anchors such that each collinear subset of anchors meets some minimum-weight criteria. This problem can be cast as the graph-theoretic Maximum Subgraph with Large Girth problem and thus an exact solution is computationally intractable (Raphael et al., 2004, Pevzner et al., 2004). Mauve uses a greedy breakpoint elimination algorithm to generate an approximate solution to the maximum-weight non-collinear anchoring problem.

To align the intervening regions of sequence between anchors our method employs the progressive dynamic programming approach of Clustal-W (Thompson et al., 1994). In progressive alignment, a phylogenetic guide tree specifies the optimal progression of sequences to align when building the multiple alignment. Rather than recalculating a guide tree during each alignment of intervening regions, Mauve infers a single global phylogenetic tree. Using a single average genome phylogeny not only saves compute time but recent results show it may yield a more robust phylogeny (Rokas et al., 2003).

The alignment algorithm can be summarized as follows:

1. Find local alignments (multi-MUMs)

2. Use the local alignments to calculate a phylogenetic guide tree

3. Select a subset of the local alignments to use as anchors–these anchors are partitioned into locally collinear blocks (LCBs)

4. Perform recursive anchoring to identify additional alignment anchors within and outside each LCB

5. Perform a progressive alignment of each LCB using the guide tree

The following sections give an overview of each step in the alignment process.

Finding local alignments

Mauve finds multi-MUMs using a simple seed-and-extend hashing method similar to that used by `GRIL` (Darling et al., 2004b). In addition to finding matching regions that exist in all genomes, the algorithm identifies matches that exist in only a subset of the genomes being aligned. While the seed-and-extend algorithm has time complexity $O(G^2n + Gn\log Gn)$ where G is again the number of genomes and n average genome length, it is very fast in practice. Finding multi-MUMs typically consumes less than a minute per bacterial size genome, and 3-4 hours per mammalian genome on a standard workstation computer. Appendix B contains a detailed description of the matching algorithm, which has been extended to approximate string matching with gapped seed patterns. The resulting local-multiple alignments are similar in nature to the alignments produced by `procrastAligner`, except that internal gaps are not permitted.

Formally we define each multi-MUM as a tuple $\langle L, S_1, ...S_G\rangle$ where L is the length of the multi-MUM, and S_j is the left-end position of the multi-MUM in the j^{th} genome sequence. We denote the resulting set of multi-MUMs as $\mathbf{M} = \{M_1...M_N\}$. The i^{th} multi-MUM in $\mathbf{M}$ is referred to as M_i. To refer to the length of M_i we use the notation

$M_i.L$ and similarly, we refer to the left end of M_i in the j^{th} genome sequence using the notation $M_i.S_j$. If multi-MUM M_i includes a region in the reverse complement orientation in sequence j, we define the sign of $M_i.S_j$ to be negative. Finally, if multi-MUM M_i does not exist in sequence j, we define $M_i.S_j$ to be 0 – the left-most position in any genome is 1 (or -1).

Calculating a guide tree

The method described to find local alignments differs from that used by `GRIL` in that it can identify local alignments in subsets of the genomes under study. Mauve exploits the information provided by subset multi-MUMs as a distance metric to construct a phylogenetic guide tree using Neighbor Joining (Saitou and Nei, 1987).

Specifically, the ratio of base pairs shared between two genomes to their genome length provides an estimate of sequence similarity. A log transformation converts the similarity estimate to a distance value for the Neighbor Joining distance matrix. Because multi-MUMs can overlap each other, calculating the similarity metric requires that overlaps among multi-MUMs are resolved such that each matching residue counts only once. To resolve an overlap, one match remains unchanged while the overlapping portion of the other match gets trimmed off and its remaining portion can still be counted. Mauve resolves overlaps in favor of the higher multiplicity match, where $multiplicity(M_i)$ is defined as the number of genomes for which $M_i.S_j \neq 0$. If the multiplicity of two overlapping matches is identical, the overlap is resolved in favor of the longer match.

After eliminating overlaps in M, we can count the number of matching residues $Match_{x,y}$ between two genomes G_x and G_y as $Match_{x,y} = \sum_{i=1}^{|M|}(M_i.S_x)^0(M_i.S_y)^0 M_i.L$. The distance between genomes can then be calculated as $d_{match}(G_x, G_y) = -\log \frac{Match_{x,y}}{2\min(|G_x|,|G_y|)}$.

This definition of distance is similar to that used by others for whole-genome phylogeny reconstruction (Henz et al., 2005).

Because the anchor selection method described below operates only on multi-MUMs with $multiplicity(M_i) = G$, the guide tree is calculated prior to anchor selection so that it can take advantage of multi-MUMs with $multiplicity(M_i) < G$.

Selecting a set of anchors

In addition to local alignments that are part of truly homologous regions, the set of multi-MUMs $\mathbf{M}$ may contain spurious matches arising due to random sequence similarity. This step attempts to filter out such spurious matches while determining the boundaries of locally collinear blocks (LCB). An LCB can be considered a consistent subset of the local alignments in $\mathbf{M}$. Formally, an LCB is a sequence of local alignments $lcb \subseteq \mathbf{M}$, $lcb = \{M_1, M_2, ..., M_{|lcb|}\}$ that satisfies a total ordering property such that $M_i.S_j \leq M_{i+1}.S_j$ holds for all i, $1 \leq i \leq |lcb|$ and all j, $1 \leq j \leq G$. For a given set of multi-MUMs, the minimum partitioning of $\mathbf{M}$ into collinear blocks can be found through breakpoint analysis (Blanchette et al., 1997). Breakpoint analysis requires that matching regions exist in *all* genomes under study, so multi-MUMs with multiplicity less than G are removed from $\mathbf{M}$ before performing this step of the algorithm.

Given a minimum weight criteria $MinimumWeight \geq 0$, Mauve uses a greedy breakpoint elimination algorithm to remove low-weight collinear blocks of $\mathbf{M}$. As part of step 3 above, Mauve performs the following substeps repeatedly until all collinear blocks in $\mathbf{M}$ meet the minimum weight requirement:

3.1 Determine a partitioning of $\mathbf{M}$ into collinear blocks $\mathbf{CB}$

3.2 Calculate the weight, $w(cb_i)$ of each collinear block $cb_i \in \mathbf{CB}$

3.3 Identify the minimum weight collinear block: let $z = \min_{cb \in \mathbf{CB}} w(cb)$

3.4 Stop if $w(z) \geq MinimumWeight$

3.5 Delete the minimum weight collinear block: remove each multi-MUM $M \in z$ from $\mathbf{M}$

3.6 Where breakpoints have been eliminated by removing z merge surrounding collinear blocks and update their weights

3.7 Go to step 3.3

Here $w(cb)$ is defined as $\Sigma_{M_i \in cb} M_i.L$. Step 3.1 is identical to the method used by `GRIL` for partitioning $\mathbf{M}$ into collinear subsets and is described in Appendix C.

In order to provide a fair measure of weight, each nucleotide in an LCB should count only once toward its weight. For this reason, breakpoint determination uses the set of non-overlapping multi-MUMs that remain after guide tree calculation. By default the $MinimumWeight$ parameter is set to $3k$, where k is the seed length used during the initial search for multi-MUMs. We chose $3k$ as a default minimum weight because it appears to filter the majority of spurious matches in data sets we have evaluated. Figure 5 illustrates the process of identifying collinear blocks of multi-MUMs and how removing a low-weight collinear region can eliminate a breakpoint. The resulting collinear sets of anchors delineate the LCBs that are used to guide the remainder of the alignment process.

Recursive anchoring and gapped alignment

The initial anchoring step may not be sensitive enough to detect the full region of homology within and surrounding the LCBs. In particular, repetitive regions and regions with frequent nucleotide substitutions are likely to lack sufficient anchors for complete

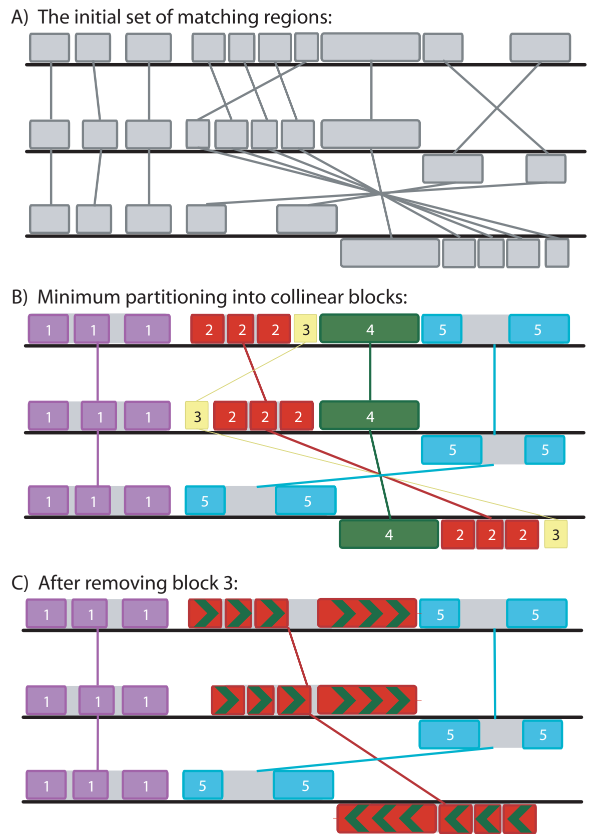

Figure 5: A pictorial representation of greedy breakpoint elimination in 3 genomes. **A**) The algorithm begins with the initial set of local alignments (multi-MUMs) represented as connected blocks. Blocks below a genome's center line are inverted relative to the reference sequence. **B**) the matches are partitioned into a minimum set of collinear blocks. Each sequence of identically-colored blocks represents a collinear set of matching regions. One connecting line is drawn per collinear block. Block 3 (yellow) has a low weight relative to other collinear blocks. **C**) As low weight collinear blocks are removed, adjacent collinear blocks coalesce into a single block, potentially eliminating one or more breakpoints. Gray regions within collinear blocks are targeted by recursive anchoring.

alignment. Using the existing anchors as a guide, two types of recursive anchoring are performed repeatedly. First, regions outside of LCBs are searched to extend the boundaries of existing LCBs and identify new LCBs. In figure 1C, this corresponds to searching the white regions outside LCBs. Second, unanchored regions within LCBs are searched for additional alignment anchors. This corresponds to searching the grey regions within LCBs in Figure 1C.

When searching for additional anchors outside existing LCB boundaries, two factors contribute to Mauve finding additional anchors. First, Mauve uses a smaller value of the match seed size k. Second, only the regions outside existing LCB boundaries are searched, so regions not unique in the entire genome may be unique within regions outside LCBs. Not only can the range of existing LCBs be extended by searching regions outside LCB boundaries, but new LCBs that meet the minimum weight requirement can be identified as well. To perform the search, the outside sequences in each genome are concatenated into a single sequence per genome. We refer to the set of concatenated sequences as $\mathbf{S}$ and the concatenated sequence from the j^{th} genome as S_j. Multi-MUMs of minimum length k are found, where $k = seed\_size(\mathbf{S}) - 2$, and $seed\_size(\mathbf{S}) = \log_2(\Sigma_{j=1}^{G} \frac{length(S_j)}{G})$. Because the left-end coordinates of each new multi-MUM are defined in terms of the concatenated sequence they must be transposed back into the original coordinate system. Also, any matches spanning two concatenated subsequences must be split. The transposed multi-MUMs are added to $\mathbf{M}$ and iterative removal of low-weight collinear subsets is performed as above. The process of searching regions outside LCBs is repeated until $\Sigma_{cs \in \mathbf{CS}} w(cs)$ remains the same during two successive iterations of the search.

In addition to missing anchors outside the boundaries of LCBs, the initial anchoring pass may have lacked the sensitivity to find anchors in large regions within each LCB. Because progressive alignment requires relatively dense anchors (at least one anchor per 10Kbp of sequence), Mauve performs recursive anchoring on the intervening regions between each pair of existing anchors. Not only does this step anchor more divergent regions of sequence, it also locates anchors in conserved repeats because many k-mers that are not unique in the whole genome are likely to be unique within the intervening regions between existing anchors.

Unlike other genome aligners which perform a fixed number of recursive passes with a pre-determined sequence of anchor sizes, Mauve calculates a minimum anchor size based on the length of the intervening sequence and stops recursive anchoring when either no additional anchors are found or when the intervening region is shorter than a fixed length, defaulting to 200bp. During each recursive anchor search new LCBs may be found, for example in the case of local rearrangements or in-place inversion, and these new LCBs must also meet the $MinimumWeight$ requirement. For each recursive search, k is calculated as above: $k = seed\_size(\mathbf{S})$ where $\mathbf{S}$ is the set of intervening sequences, one per genome. By dynamically calculating the value of k, Mauve ensures that k is sized appropriately for the intervening region. Selecting k too large prevents discovery of multi-MUMs in polymorphic regions, whereas selecting k too small increases the likelihood that k-mers will not be unique in the intervening region.

Armed with a complete set of alignment anchors, Mauve performs a Clustal-W progressive alignment using the genome guide tree calculated previously. The progressive alignment algorithm is executed once for each pair of adjacent anchors in every LCB, calculating a global alignment over each LCB. Tandem repeats less than 10Kbp in total length are aligned during this phase. Regions larger than 10Kbp without an anchor are ignored.

4.1 Alignment results

The Mauve genome alignment procedure results in a global alignment of each locally collinear block that has sequence elements conserved among *all* the genomes under study. Nucleotides in any given genome are aligned only once to other genomes suggesting orthology among aligned residues—Mauve makes no attempt to align paralagous regions. The remaining unaligned regions may be lineage-specific sequence, or conserved or paralagous repetitive regions and can be identified as such during subsequent processing with other tools. Large ($>$ 10Kbp) regions introduced to a subset of the genomes by horizontal transfer are not aligned by Mauve because they do not have alignment anchors conserved among all sequences. Both large and small regions existing in only a subset of the genomes and that also underwent local rearrangement remain unaligned.

Alignment of 9 enterobacteria

We applied Mauve to align the the 9 target enterobacterial genomes shown in Figure 6. Previous studies of these genomes indicates they underwent significant genome rearrangement, horizontal transfer, and other recombination (Perna et al., 2001, Deng et al., 2003). Mauve consumed 3 hours to align the 9 taxa on a 2.4GHz computer with 1GB of RAM. The alignment of the 9 taxa reveals 45 LCBs with a minimum weight of 69. Figure 6 shows the guide tree generated for these species. The visualization of the genome rearrangement structure generated by the Mauve viewer is shown in Figure 7. We can quickly visually confirm several known inversions such as the O157:H7 EDL933

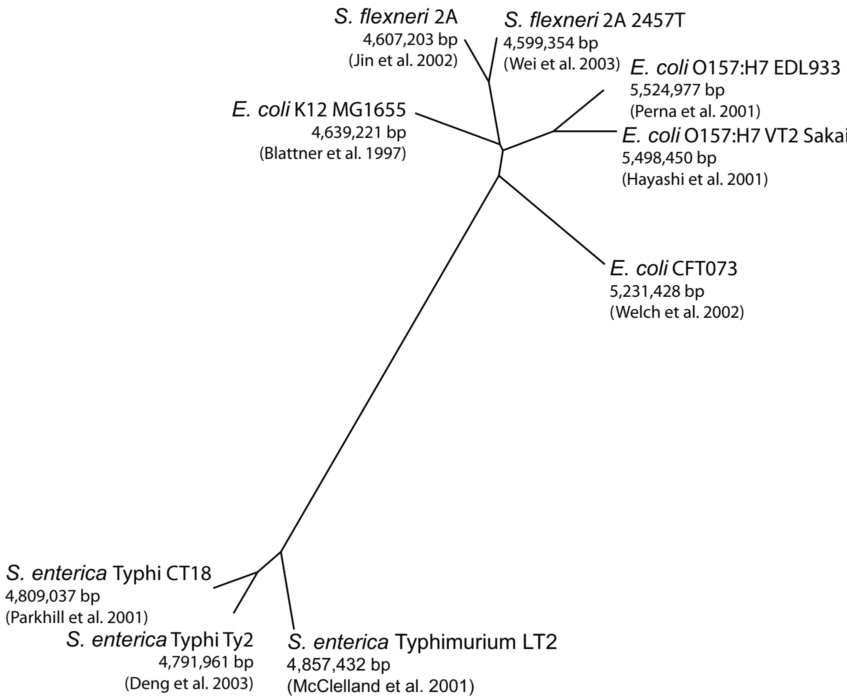

Figure 6: An unrooted phylogenetic tree relating nine enteric genomes. The tree is a phylogenetic guide tree calculated using Neighbor-Joining on a genome-content distance metric.

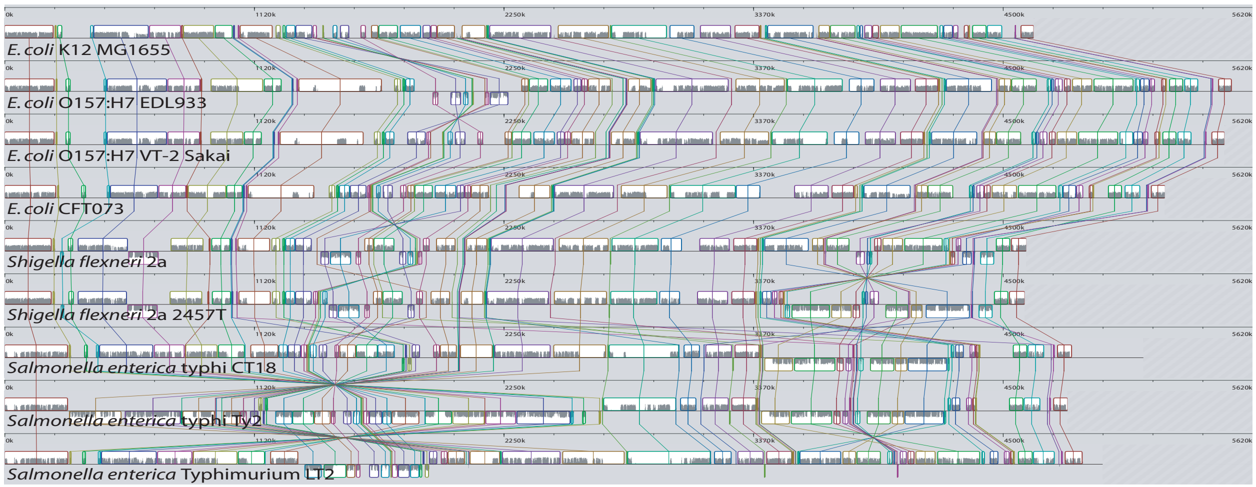

Figure 7: Mauve visualization of an alignment of the 9 target enterobacteria shown in Figure 6. Each genome sequence is plotted along a horizontal track. Locally collinear blocks in each genome (regions without rearrangements) are surrounded by a colored box and connected to the homologous region in each of the other genome sequences. Blocks below a genome's center line are in the reverse complement orientation relative to the reference genome. Within each locally collinear block, a similarity plot shows the average sequence conservation in that region. The *Shigella* and *Salmonella* genomes have undergone more genome rearrangements than that of *E. coli*, likely due to the presence of specific mobile genetic elements.

inversion relative to K-12 (Perna et al., 2001) and the large inversion about the origin of replication among the *S. enterica* serovars Typhi CT18 and Ty2 (Deng et al., 2003).

We proceeded to extract conserved backbone sequence from the alignment. Again, backbone is defined as regions of the alignment containing more than 50 gap-free columns without stretches of 50 or more consecutive gaps in any single genome sequence. Under this definition, the 9 enterobacteria have 2.86Mbp of conserved backbone sequence broken into 1252 backbone segments. Across the backbone the level of nucleotide identity is high, above 95% within each *Escherichia* and *Salmonella* genus, and about 70% across the two genuses (data not shown).

4.1.1 Alignment of mammalian genomes

We applied the Mauve genome alignment system to perform a whole-genome alignment of the mouse, rat, and human genomes. The RepeatMasked assemblies of human (NCBI 35), mouse (NCBI 33), and rat (RGSC 3.4) were searched for unique 3-way seed matches on the forward and reverse strands using the palindromic seed pattern: `11111*111*11*1*11*111*11111`. This seed pattern is the most sensitive pattern at weight 21 for sequences with 65%-75% identity, as described in Chapter 3. Initial seed matches were maximally extended in each direction until the seed pattern no longer matched at any overlapping position. A total of 922,081 ungapped 3-way alignments containing unique sequence resulted. The initial set of 3-way matches gave rise to 567,782 LCBs, to which we applied greedy breakpoint elimination to remove all LCBs up to a minimum weight of 55, yielding a baseline set of 520,423 3-way matches that compose 6483 LCBs. The complete analysis consumed approximately 24 hours on a 1.6GHz Linux PC with 2.5GB memory and two hard disks used for an external-sort of the string

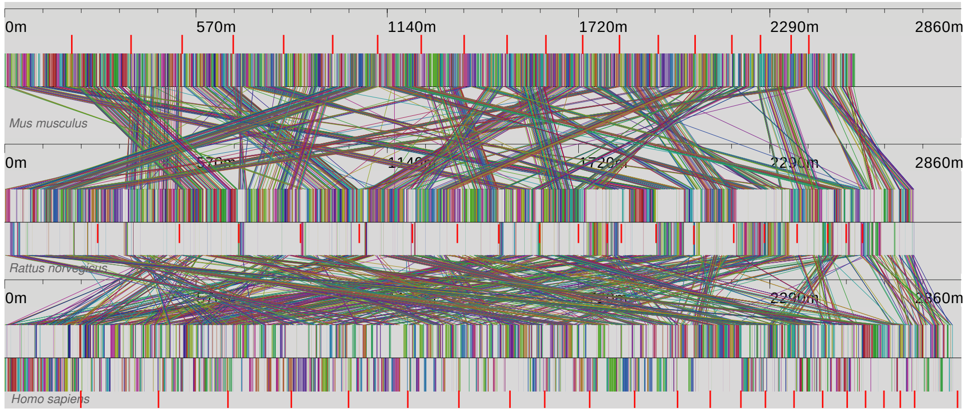

Figure 8: Mauve visualization of locally collinear blocks identified between concatenated chromosomes of the mouse, rat, and human genomes. Each of the 1,251 blocks shown here have a minimum weight of 90. Red vertical bars demarcate interchromosomal boundaries. The Mauve rearrangement viewer enables users to interactively zoom in on regions of interest and examine the local rearrangement structure. The computation consumed approximately 24 hours on a 1.6GHz workstation with 2.5GB memory.

matching data structures.

We further applied greedy breakpoint elimination to the baseline set of 6,483 LCBs, recording the observed genomic permutation at each successively higher LCB weight up to a minimum weight of 100,000. At minimum weight 97,673 (the last weight before 100,000), there are 75 3-way LCBs among the mouse, rat, and human genomes. At weights larger than 500, the LCB weight roughly corresponds to the overall chromosomal length of an LCB, with the average LCB chromosomal length being 100-1000x the LCB weight. A visualization of the overall mammalian LCB structure is shown in Figure 8. Complete 3-way mammalian genome alignments based on the initial set of 6,483 LCBs were computed using 24 hours of parallel compute time on a 96-CPU Orion Deskside cluster. The results are available from `http://gel.ahabs.wisc.edu/~koadman/orion_results`

4.2 Discussion

With the advent of genome sequencing a new type of sequence alignment problem, that of whole genome comparison, has emerged. Early approaches to genome alignment were designed to tackle dramatically increased sequence lengths, but did not consider the additional types of evolutionary events observed on the genome scale. Genome rearrangements, horizontal transfer, and duplication obfuscate orthology. As genomes continue to be sequenced, automatic and accurate identification of genome rearrangements becomes increasingly important, especially as high levels of rearrangement have been observed among both eukaryotes and prokaryotes (Pevzner and Tesler, 2003b, Lefebvre et al., 2003, Pevzner and Tesler, 2003a).

The Mauve genome alignment method represents a first step toward multiple genome comparison in the presence of large-scale evolutionary events. It is capable of aligning conserved regions in the presence of genome rearrangement, and appears to scale efficiently to long genomes. However, our experience with Mauve clearly indicates that many challenges remain in genome alignment. A more sensitive local alignment technique would permit our method to be applied to more distantly related organisms. A method for determining breakpoints with local alignments existing in a subset of the genomes would facilitate anchored alignment of the large lineage-specific regions currently missed.

Some organisms are known to have small, local sequence rearrangements such as reordering of protein domains in coding regions. In such cases, the proximity of the rearrangement to neighboring homologous sequence should clearly be considered. Other types of rearrangement do not exhibit locality bias: symmetric inversions about the origin and terminus of replication and rearrangements mediated by mobile elements are common in prokaryotes and can move sequence to distant parts of the genome. A more sophisticated rearrangement scoring method may attempt to score a particular pattern of anchors based on the sequence of rearrangement events and recombination mechanisms suggested by that pattern of anchors.

4.3 Acknowledgments

Portions of this chapter appeared as Darling, Mau, Blattner, and Perna (2004a).

Chapter 5

Alignment of genomes with lineage-specific content

5.1 Introduction

Advances in genome sequencing technology have made large-scale sequencing of microbial genomes not only possible, but relatively affordable (Margulies et al., 2005, Shendure et al., 2005). It has been estimated that current genome sequences represent less than 1% of global microbial species diversity (Tettelin et al., 2005). Studies aiming to catalog environmental sequence diversity have already produced initial data (Venter et al., 2004, Tringe et al., 2005), and more are expected to follow. Genomic sequence comparison stands to provide a framework for understanding the biology of newly sequenced organisms through comparison to model organisms.

In the context of comparative genomics, whole genome alignments solve an important problem. While it may be possible to assess the gene content of an organism using gene-based reciprocal-best-hit BLAST methods, such approaches are error-prone (Koski and Golding, 2001), neglect important non-genic content and perhaps more importantly, frequently neglect comparison of overall genome structure. Genome alignment, on the other hand, provides a framework for simultaneous comparison of genic and non-genic

Organism	Genome size w/Plasmids	Accession
E. coli K12 MG1655	4654221	U00096
E. coli O157:H7 EDL933	5623806	AE005174
E. coli O157:H7 Sakai	5594477	BA000007
E. coli HS	4643538	AAJY00000000
E. coli E24377A	4980187	AAJZ00000000
E. coli CFT073	5231428	AE014075
E. coli UTI89	5179971	CP000243
Shigella boydii Sb227	4646520	CP000036
Shigella flexneri 2457T	4988914	AE014073
Shigella flexneri 301	4828821	AE005674
Shigella dysenteriae Sd197	4551958	CP000034
Shigella sonnei Ss046	5039661	CP000038
Salmonella enterica Choleraesius B67	4944000	AE017220
Salmonella enterica Typhi Ty2	4791961	AE014613
Salmonella enterica Typhi CT18	5133713	AL513382
Salmonella typhimurium LT2	4951371	AE006468
Salmonella paratyphi A ATCC9150	4585229	CP000026
Yersinia pestis Antiqua	4879836	CP000308
Yersinia pestis Nepal 516	4646286	CP000305
Yersinia pestis 91001	4803217	AE017042
Yersinia pestis CO92	4829855	AL590842
Yersinia pestis KIM	4781914	AE009952
Yersinia pseudotuberculosis IP31758	4721828	AAKT00000000
Yersinia pseudotuberculosis IP32953	4840899	BX936398
Erwinia chrysanthemi 3937	4922802	-
Erwinia caratovora SCRI1043	5064019	-

Table 3: Twenty-five publicly-available, finished enteric genomes sequences form our target set for multiple genome alignment.

content and genome structure. Genome alignment faces a challenge, however, as most current methods do not account for large-scale mutational forces that disrupt gene order, create paralogs, and incorporate novel content into genomic sequences. Furthermore, of the genome alignment methods that do exist, few have been integrated into a single coherent analysis methodology, limiting their widespread use.

In the present study, we focus on a large set of enteric bacteria (listed in Table 3) whose genomes have proven unalignable using previous techniques. This group includes microbes whose rates and patterns of mutation exhibit substantial variability, as shown in Figure 9. Specifically, the closely related members of the *Yersinia* genus appear to have unstable chromosome structure (Deng et al., 2002), showing evidence for numerous rearrangements since their divergence 1,500–20,000 years ago (Achtman et al., 1999). At the opposite extreme, estimates place the speciation of *E. coli* and *Salmonella* at 120–160 million years ago (Ochman and Wilson, 1987), but cross-species comparisons show little or no change in genome organization among *E. coli* and *Salmonella*. Thus, rates of rearrangement in enteric bacteria are lineage-specific and can vary substantially.

In addition to genome rearrangement, the genomes of enteric bacteria also undergo substantial gain and loss of genetic material, which we collectively refer to as *gene flux*. Within the species *E. coli*, pairwise comparisons of individual isolates indicate that each isolate may contain as much as 20% novel gene content relative to the other (Perna et al., 2001). The large amount of novel content in *E. coli* isolates implies that either the cenancestor of *E. coli* had a relatively large genome which has undergone lineage-specific reductions, or that *E. coli* rapidly acquires novel content from the environment.

When designing a system for multiple genome alignment, the observed heterotachy in rates of genomic rearrangement and gene flux becomes an important consideration. An alignment scoring scheme that scales a rearrangement penalty based on nucleotide divergence among taxa would not accurately capture the patterns observed in our data.

We describe a new genome alignment method that directly addresses heterotachy in the rates of genomic rearrangement and gene flux. The new method extends previous methods for progressive genome alignment (Brudno et al., 2003a, Bray and Pachter, 2003) by using an anchor selection scheme that applies a breakpoint penalty to account for rearrangement. The scoring method adjusts the breakpoint penalty based

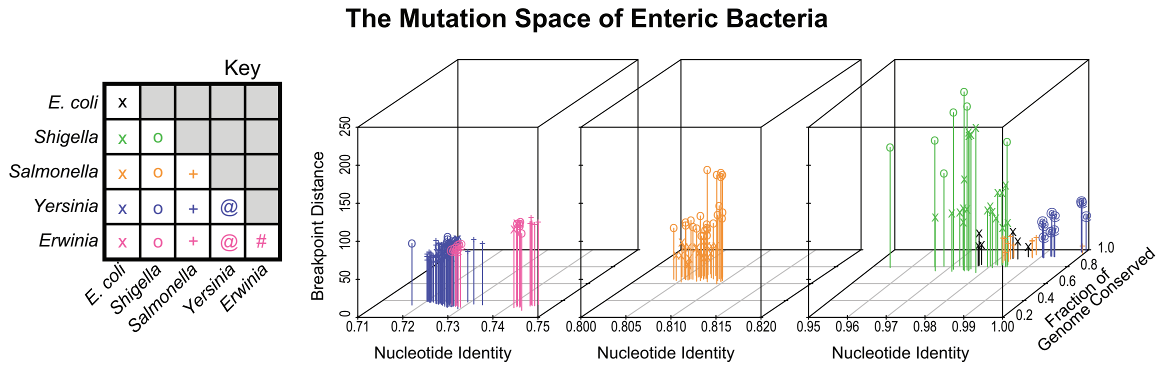

Figure 9: Pairwise genome alignments of enteric bacteria reveal the level of nucleotide identity in conserved segments, average fraction of the genome contained in conserved segments, and number of gene-order breakpoints among each pair. Within the genus *Yersinia* little nucleotide-level divergence exists, but a substantial amount of genomic rearrangement has occurred (rightmost blue points). For easier visualization, the mutation space has been split into three focused regions of nucleotide identity which contain all pairwise comparisons.

on pairwise estimates of breakpoint distance and genome conservation distance. We apply a random-walk statistic to the resulting multiple genome alignments to distinguish segments conserved among subsets of the taxa from segments conserved among all taxa and from novel sequence. We implement the new alignment method in a freely available, open-source software package called Progressive Mauve, available from `http://gel.ahabs.wisc.edu/mauve`

5.2 Methods

The Progressive Mauve alignment method consists of five basic steps: (1) local-multiple alignment of highly similar unique subsequences, (2) construction of breakpoint and conservation distance matrices and a conservation-based guide tree, (3) progressive anchored alignment, (4) iterative refinement within collinear segments, and (5) identification of segments conserved among two or more genomes using random-walk statistics and transitive homology relationships. We describe each of these steps in turn below.

Notation and assumptions

Our genome alignment algorithm takes as input a set of G genome sequences $g_1, g_2, \cdots \in \mathbf{G}$. We denote the length of genome i as $|g_i|$. Our method computes alignments along a guide tree Ψ, and we use n to denote an arbitrary node in Ψ. As Ψ is a rooted bifurcating tree, an internal node n always has two children, which we refer to as $n.c_1$ and $n.c_2$ or simply c_1 and c_2 when n is implied by context. Furthermore, we define the set of leaf nodes at or below n as $Leaf(n)$ and similarly, the leaf nodes at or below the children of n as $Leaf(c_1)$ and $Leaf(c_2)$. The two sets of leaf nodes on c_1 and c_2 are disjoint, and each leaf node represents a genome from the set of input genomes $\mathbf{G}$. Finally, we use the function $Des(n)$ to refer to all descendant nodes at or below n.

Various default parameter settings in our software implementation depend on the average length of input genome sequences. We define a function to compute average genome length as:

$$AvgSize(\mathbf{G}) = \sum_{g \in \mathbf{G}} \frac{|g|}{G}$$

5.2.1 Local multiple alignment

We perform local-multiple alignment using a variation of the technique described in Appendix B. The new seed-and-extend string matching method seeds local multiple alignments in unique regions of sequence that match in two or more genomes, just like the previous method. If a seed matches in three or more genomes but is unique in only a subset of those genomes, the new method extends the seed among the subset in which it is unique. The previous approach would have ignored such seed matches. We further improve the new method to use palindromic spaced seeds Darling et al. (2006), allowing for some degeneracy in the matching regions. Thus, the resulting local multiple alignments can no longer be considered multi-MUMs, as they may contain mismatches (but no indels). By default, we use a seed with weight equal to $\log_2(AvgSize(\mathbf{G})/1.5$. For enteric genomes, the default seed weight is 15, with length 23. We refer to the initial set of local multiple alignments generated in this step as $\mathbf{M}_{initial}$.

5.2.2 Pairwise distance matrix and guide tree construction

We construct two distance matrices, one which estimates the breakpoint distance among each pair of genomes, and a second which estimates the amount of non-homologous sequence among any pair of genomes (conservation distance). We refer to the breakpoint distance matrix as $\mathbf{B}$ and the conservation distance matrix as $\mathbf{C}$. Both are $G \times G$ matrices with values in the range $[0, 1]$. We compute the conservation distance in the same manner as previously reported Darling et al. (2004a). Briefly, the conservation distance for a pair of genomes is the average fraction of each genome covered by pairwise local alignments, subtracted from one to form a distance. The precomputed local multiple alignments are projected to pairwise alignments for the purpose of computing conservation distance.

The breakpoint distance between a pair of genomes G_i, G_j is simply the number of breakpoints in homologous gene order between that pair of genomes. Since we do not know *a priori* which segments of G_i and G_j are homologous we must estimate the breakpoint distance through genome alignment. Without already knowing the relative amounts of nucleotide divergence, gene flux, and genomic rearrangement among G_i and G_j, it is difficult to pick a single breakpoint penalty for greedy breakpoint elimination

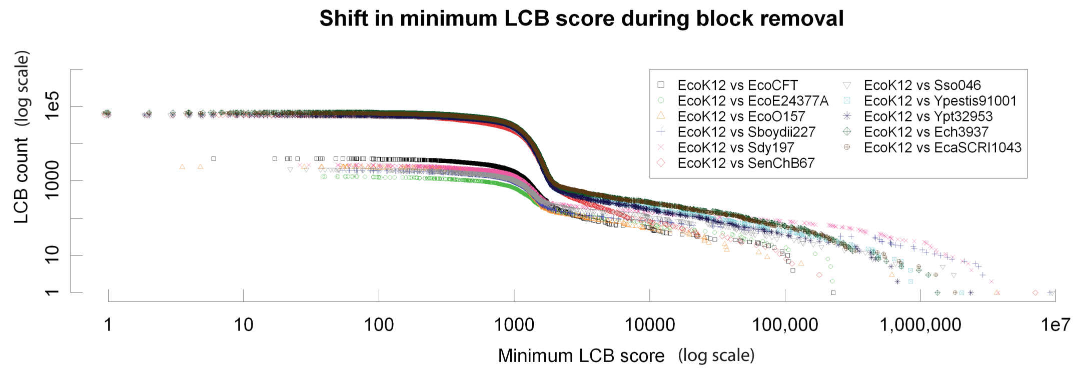

Figure 10: The change in the number of LCBs as minimum scoring LCB are successively removed. Pairwise comparisons of *E. coli* K12 MG1655 with several other enterobacteria are shown. A pronounced downward shift in the number of LCBs occurs as the minimum score surpasses 2000. For this data set we use a minimum LCB score of 100,000 to provide a conservative estimate of breakpoint distance.

(described in Chapter 4) that provides precise estimates of the breakpoint distance for any G_i and G_j. We use the anchor scoring metrics described below to compute LCB anchor scores on pairwise matches among each pair of G_i and G_j. However, we observe that small, usually spurious, matches constitute the large number of low-scoring LCBs present in most pairwise comparisons, whereas most of the genome (and matches) usually reside in a small number of high scoring LCBs.

Figure 10 illustrates the number of LCBs as a function of the minimum LCB score remaining during application of greedy breakpoint elimination to enteric genome sequences. Manual validation of genome alignments indicates that only correct pairwise LCBs remain at minimum LCB scores ranging between 30,000-50,000. Furthermore, it appears that a conservative scoring threshold of 100,000 still captures the relative number breakpoints among pairwise comparison. Since we use estimated breakpoint distances as scaling factors for subsequent alignment scoring, we do not need to know the absolute breakpoint distance; a relative estimate of rearrangement rate suffices. The software implementation of our method takes the default minimum LCB score for distance estimation to be: $4500 \log_2(\sum_{g \in \mathbf{G}} \frac{|g|}{G})$ which equates to roughly 100,000 for genomes averaging 5Mbp in size.

The breakpoint distance is the total number of pairwise LCBs among G_i and G_j, minus 1, however $\mathbf{B}_{i,j}$ must be a value between 0 and 1. Referring to the estimated breakpoint distance between G_i and G_j as $d_{i,j}$, we arrive at values for $\mathbf{B}$ through the following normalization:

$$\begin{aligned} MaxDist(\mathbf{G}) &= \max(\frac{AvgSize(\mathbf{G})}{50000}, \max_{G_a, G_b \in \mathbf{G}} d_{a,b}) \\ \mathbf{B}_{i,j} &= \frac{d_{i,j}}{2MaxDist(\mathbf{G})} \end{aligned}$$

Here, $AvgSize(\mathbf{G})$ computes the average genome size, while $MaxDist(\mathbf{G})$ computes the maximum breakpoint distance. Rather than strictly using the maximum observed breakpoint distance, we estimate a "high" rate of rearrangement to be 20 breakpoints per megabase of sequence and use the maximum of the "high" estimate and the observed estimates as our normalizing distance. Without this adjustment, the values of $\mathbf{B}$ would vary considerably when analyzing only stable genomes versus a combination of rearranged and stable genomes. Finally, we multiply $MaxDist(\mathbf{G})$ by two to ensure that distances never exceed 0.5, a value which provides substantial scaling of the scoring functions described below.

We compute the topology and branch lengths of the guide tree Ψ using neighbor-joining (Saitou and Nei, 1987) on the pairwise conservation distance matrix. Our conservation distance measure is not an additive distance, thus the guide tree may have negative branch lengths. In general, negative lengths are inconsequential to the alignment procedure.

5.2.3 Objective scores

Like many sequence alignment methods, Progressive Mauve seeks to optimize a well-defined objective score which has been designed to assign higher values to better alignments. For performing gapped alignments of collinear segments, we apply the sum-of-pairs score with affine gap penalties (Thompson et al., 1994, Feng and Doolittle, 1987). For selecting the collinear chains of local alignments that serve as genome alignment anchors we apply a different objective score which we refer to as the the sum-of-pairs anchoring score. We also describe a variation on the sum-of-pairs anchoring score which can account for the genome arrangement inferred at internal nodes of the guide tree.

Local alignment scoring

During the course of genome alignment, our method attempts to discriminate between local alignments that suggest orthology (or xenology) and alignments of regions with random similarity or paralogy. Local alignments believed to be in orthologous (or xenologous) regions ultimately become anchors for the whole-genome alignment. We score local alignments using an anchor scoring scheme designed to assign high scores to well-conserved regions that are unique in each genome.

Prior to beginning genome alignment, we compute a uniqueness value for each position of every input genome. For a given position in G_i, the uniqueness is calculated as 1 over the number of genome-wide matches to the spaced seed pattern at that site. The uniqueness of each site always ranges between 1 and 0, with highly repetitive sites having uniqueness values close to 0. We refer to the uniqueness value of site x in G_i as $\mathbf{U}_{i,x}$.

For a pairwise local alignment M among genomes G_i and G_j, we compute the average uniqueness of M using only sites in G_i and G_j that are aligned to each other in M. Skipping unaligned sites prevents large internal gaps from influencing the uniqueness of M. Define an aligned column of M as a tuple $col = \langle a, b \rangle$ containing the aligned sequence coordinates in G_i and G_j, and refer to coordinates as $col.a$ and $col.b$, respectively. If we define the set of all aligned columns in M as $cols(M)$, then the average uniqueness score of M can be written as

$$Uniqueness(M) = \sum_{col \in cols(M)} \frac{\mathbf{U}_{i,col.a} + \mathbf{U}_{j,col.b}}{2|cols(M)|}$$

We score the quality of a given pairwise local alignment M using the HOXD nucleotide substitution matrix (Chiaromonte et al., 2002). The HOXD matrix has been demonstrated to provide good discrimination between homologous and non-homologous sequence in a variety of organisms, even at high levels of sequence divergence. We use previously derived affine gap penalties, -400 for a gap open and -35 for a gap extension (Schwartz et al., 2003). We refer to the pairwise affine gap and substitution score as $PairScore(M)$.

The total anchor score of M is computed as

$$AnchorScore(M) = PairScore(M) \cdot Uniqueness(M).$$

LCB scoring

Although in general an LCB may refer to a collinear segment of two or more genomes, the LCBs considered during our progressive alignment procedure are always pairwise.

We calculate the anchor score of an LCB as the sum of its constituent pairwise local alignment scores:

$$LcbAnchorScore(L) = \sum_{M \in L} AnchorScore(M)$$

The weighted breakpoint penalty

As genomes diverge they may undergo genomic rearrangement. As a result, we must identify alignment anchors that occur in a different order and orientation in each genome. To complicate matters, spurious matches and matches among paralogs also frequently occur in a different order and orientation in each genome. To ensure accurate alignment anchoring we would like to filter out any local alignments that arise due to paralogous segmental homology, in addition to any low-scoring spurious matches.

When computing LCB structure among a pair of extant genomes, we apply a breakpoint penalty designed to account for the expected amount of genomic rearrangement and gene flux that has occurred since their divergence. We define a matrix of breakpoint penalties among each pair of genomes as

$$\mathbf{W}_{i,j} = w\mathbf{B}_{i,j}\mathbf{C}_{i,j}$$

where w is a user-defined minimum LCB score. Empirical evidence indicates that a value of 30,000 gives high-quality estimates of LCB structure for our target data set (see Figure 10, full data not shown). The software implementation sets $w = 1500AvgSize(\mathbf{G})$ by default, the value of which is approximately 30,000 for our enteric genomes.

The sum-of-pairs anchoring score

Given a node n and set of pairwise LCBs among each cross-pair of the genomes at or below nodes $n.c_1$ and $n.c_2$, we compute the sum-of-pairs anchoring score as

$$SPAnchorScore(n, \mathbf{L}) = \sum_{G_i \in Leaf(c_1)} \sum_{G_j \in Leaf(c_2)} (|\mathbf{L}_{i,j}| - 1)\mathbf{W}_{i,j} \sum_{l \in \mathbf{L}_{i,j}} LcbAnchorScore(l)$$

The sum-of-pairs + ancestral anchoring score

A second, optional LCB scoring scheme used by our method is the SP extant+ancestral score. This scoring scheme has been designed to also score pairwise LCB structure between extant genomes and the sequence arrangement inferred at internal nodes of the alignment tree.

When computing LCB structure for a node n in the alignment tree we apply a weighted breakpoint penalty $\mathbf{A}_n$ which is an average penalty among cross-pairs of descendant genomes. Specifically, the values of $\mathbf{A}$ for each internal node n are defined as

$$\mathbf{A}_n = \sum_{G_i \in Leaf(c_1)} \sum_{G_j \in Leaf(c_2)} \frac{\mathbf{W}_{i,j}}{|Leaf(c_1)||Leaf(c_2)|}$$

When n has only two leaf-node descendants, representing genomes G_i and G_j, $\mathbf{A}_n$ is identical to $\mathbf{W}_{i,j}$. To arrive at the SP extant+ancestral anchor score, we then modify the original SP anchoring score to include score terms for internal nodes below n:

$$SPAncestralAnchorScore(n) = \sum_{G_i \in Des(c_1)} \sum_{G_j \in Des(c_2)} (|\mathbf{L}_{i,j}| - 1)\mathbf{A}_{i,j} \sum_{l \in \mathbf{L}_{i,j}} LcbAnchorScore(l)$$

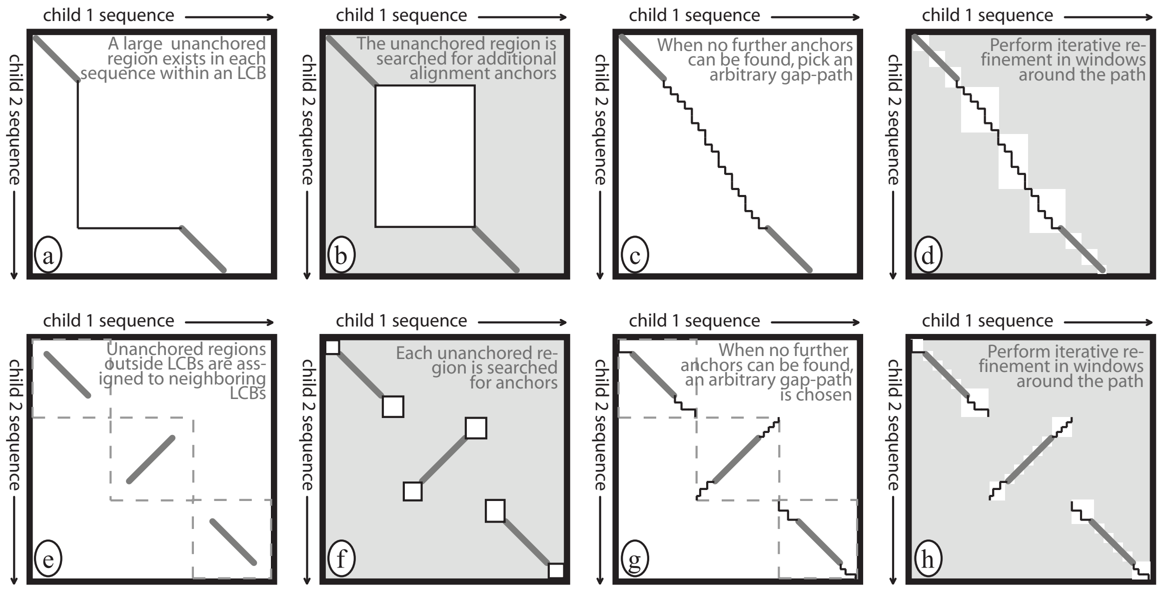

Figure 11: Treatment of regions without alignment anchors. Pairwise alignments among the two children nodes of an internal node are shown in dotplot format. Panels a-d demonstrate processing of a large gap inside a single LCB. Panels e-h demonstrate processing gaps between LCBs.

5.2.4 Progressive anchored multiple genome alignment

Staring with the guide tree Ψ, a set of local-multiple alignments $\mathbf{M}_{initial}$, and a weighted breakpoint penalty matrix $\mathbf{W}$, the following algorithm computes a multiple genome alignment among sequences in $\mathbf{G}$:

1. Select the closest pair of unaligned nodes that have the same parent in Ψ. We refer to the unaligned nodes as c_1, c_2 and their parent as n.

2. Extract all precomputed pairwise matches between cross-pairs of genomes in $Leaf(c_1)$ and $Leaf(c_2)$ from $\mathbf{M}_{initial}$. Local multiple alignments may be projected to pairwise alignments.

3. Translate pairwise matches among extant genomes into coordinates of c_1 and c_2, call the resulting set of pairwise matches $\mathbf{M}_n$. When c is a leaf node, the translation

is trivial and match coordinates remain unchanged.

4. Eliminate overlaps and resolve inconsistent alignments among matches in $\mathbf{M}_n$ as described in Darling et al. (2004a).

5. Translate matches in $\mathbf{M}_n$ back down the tree to construct a set of local-multiple alignments among $Des(n)$, which we refer to as $\mathbf{M}_t$. For every pairwise match in $\mathbf{M}_n$ a corresponding local-multiple alignment among $Des(n)$ exists in $\mathbf{M}_t$.

6. For each cross-pair of genomes $G_i,\ G_j$ in $Leaf(c_1)$ and $Leaf(c_2)$, project the local-multiple alignments in $\mathbf{M}_t$ to their pairwise coordinates. Refer to the resulting set of projected matches as $\mathbf{M}_{i,j}$. Each projected match retains a pointer to the original ancestral match in $\mathbf{M}_n$ from which it came.

7. Partition each set of pairwise projected matches $\mathbf{M}_{i,j}$ into a set of pairwise Locally Collinear Blocks $\mathbf{L}_{i,j}$

8. Compute the current SP anchor score for n as $SPAnchorScore(n, \mathbf{L})$

9. Perform sum-of-pairs greedy breakpoint elimination:

   9.1. Remove the pairwise LCB that results in the largest improvement in $SPAnchorScore(n, \mathbf{L})$. When removing the LCB, remove all pairwise projected matches in the LCB, and remove the corresponding matches in $\mathbf{M}_n$ and any other associated projections in $\mathbf{M}_{i,j}$.

   9.2. Removing the LCB may allow neighboring LCBs to coalesce. Recompute scores for all neighboring LCBs.

   9.3. Compute the new SP anchoring score $SPAnchorScore(n, \mathbf{L})$. If the new score is larger than the previous score, return to step 9.1, otherwise continue to step 10.

10. Pick arbitrary endpoints for LCBs in the breakpoint regions between LCBs (Figure 11 panel e).
11. Check whether the final SP anchoring score has improved. If not, go to step 14.
12. Recursive anchor search. Search for additional anchors in large gaps between existing anchors and outside LCBs. Figure 11, panels a, b, and e, f illustrate the recursive anchor search inside and outside LCBs, respectively.
13. Return to step 3. Use any matches identified by the recursive anchor search, in addition to the matches that remained after greedy breakpoint elimination as input to Step 3.
14. Pick an arbitrary gap path in unanchored regions (Figure 11 panels c and g).
15. Perform an anchored profile-profile alignment using MUSCLE (Edgar, 2004) The MUSCLE source code was modified to support anchored profile-profile alignment. To limit compute time, we enforce a maximum distance between anchors of 20,000nt. When we encounter a gap larger than 20,000nt between anchors, we add an anchor point on the gap-path midway between the nearest existing anchor points.
16. If nodes remain to be aligned then return to Step 1, otherwise end progressive alignment.

An example of sum-of-pairs greedy breakpoint elimination

Iterative refinement

We subject each aligned locally collinear block to an iterative refinement process using the MUSCLE sequence alignment tool. To reduce overall execution time, we use window-based iterative refinement to restrict the total search space. In window-based

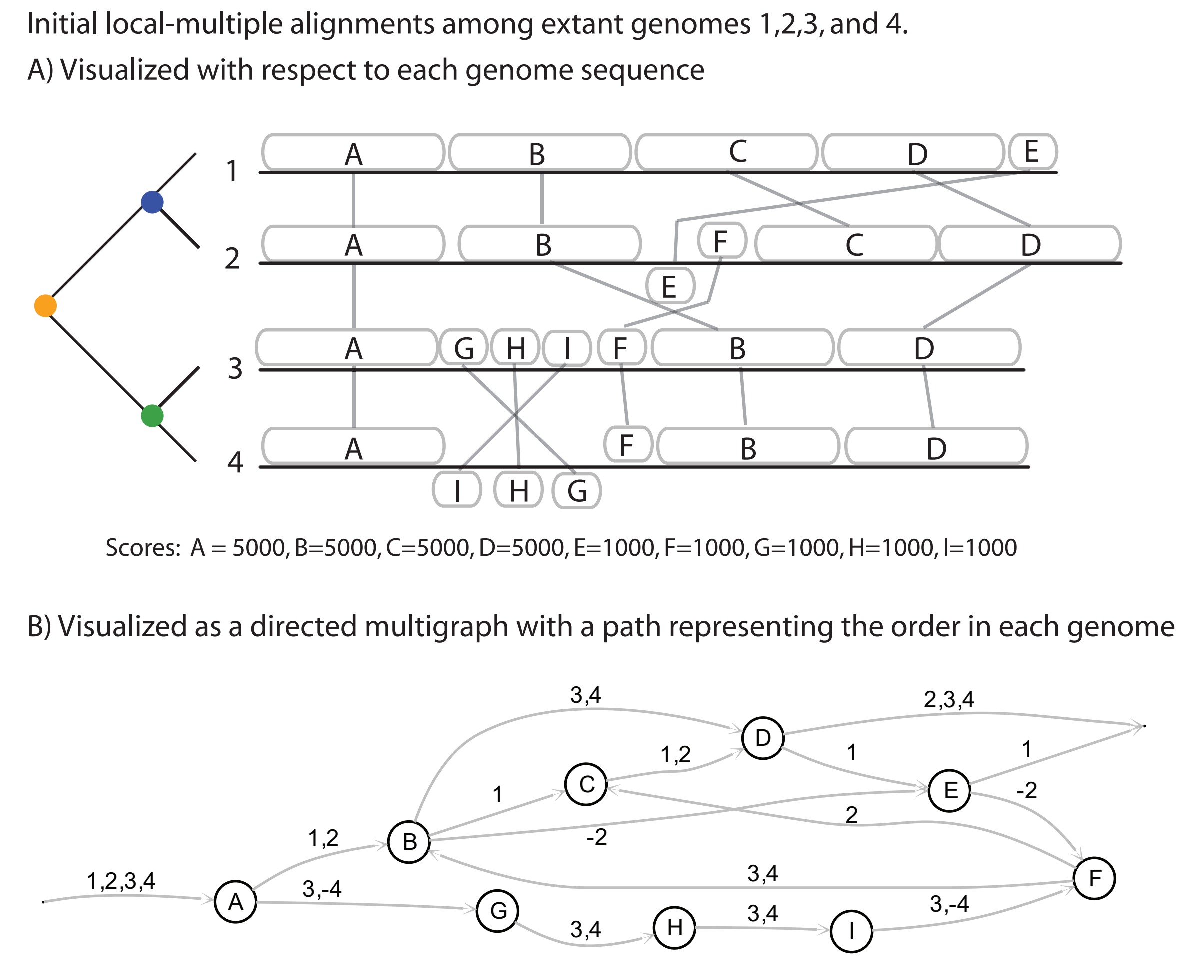

Figure 12: Panel **A**: An initial set of local multiple alignments has been calculated among four genomes, labeled 1–4. The chosen alignment guide tree is shown at left. Each genome sequence is laid out horizontally and segments contained in local-multiple alignments are depicted as blocks linked between genomes. Blocks below a genome's center line match the reverse complement strand in that genome. For simplicity we assume that pairwise alignment scores are equal for all pairs of genomes and assume the scores given above. Panel **B**: The local multiple alignments in **A** induce a directed multigraph where each local multiple alignment is a node and edges connect alignments that are adjacent in each genome. A path from source to sink vertex exists for each of genomes 1–4, with edges labeled accordingly. Traversal of a given genome's path visits nodes in the order of the corresponding local-multiple alignments in that genome. Negative edge labels indicate a switch in the strand matched by adjacent alignments.

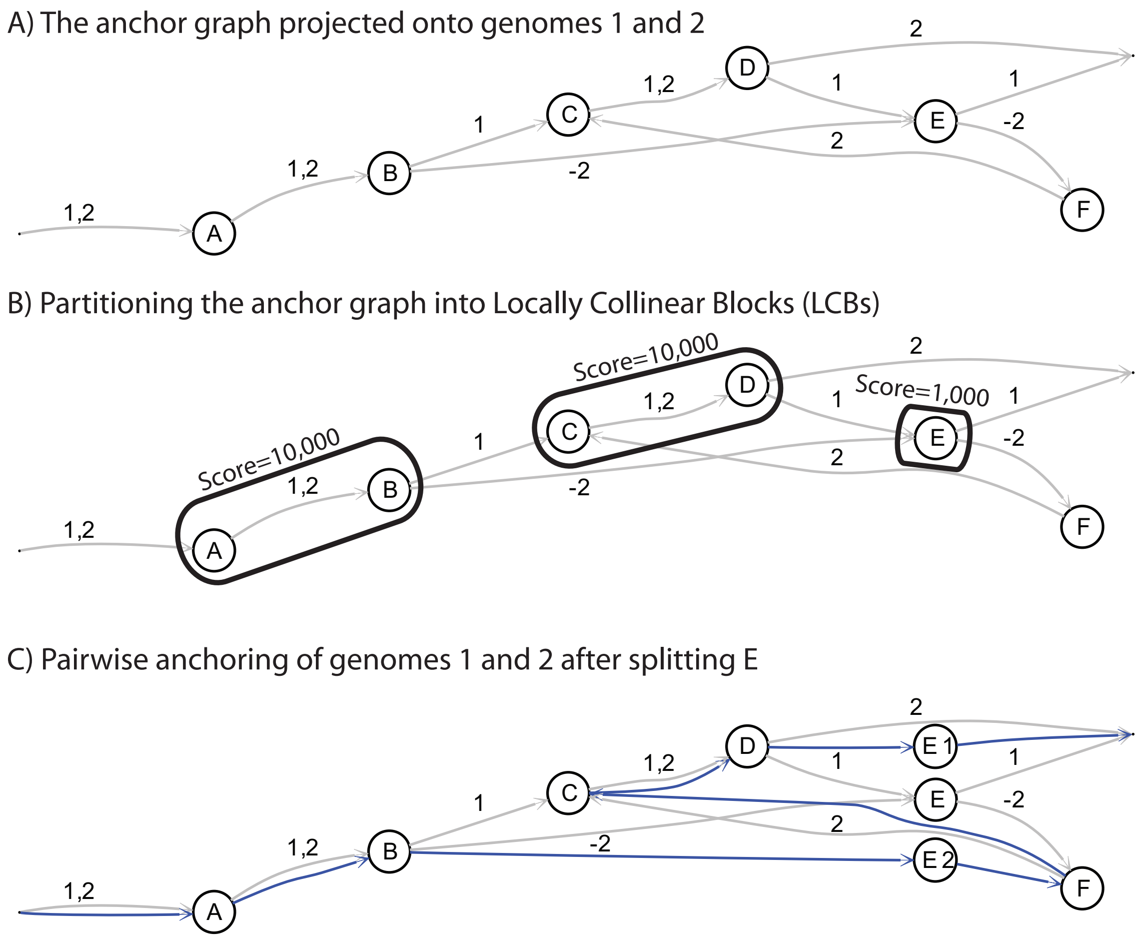

Figure 13: Panel **A**: The full graph shown in Figure 12 is projected to the subgraph containing only edges labeled 1 and 2. We perform greedy breakpoint elimination on this pairwise projection. Panel **B**: Pairwise LCBs among genomes 1 and 2 are identified as nodes connected by "simple" paths, i.e. paths with edge labels 1,2, or singleton nodes which have edges labeled with both 1 and 2 but are not part of any simple paths. A cycle exists in the subgraph among nodes C, D, E, and F, and corresponds to a putative genome rearrangement between genomes 1 and 2. The cycle partitions the local multiple alignments into three LCBs: {AB}, {CD}, and {E} with scores 10,000, 10,000, and 1,000, respectively. F does not contribute to any LCB since it doesn't match in both 1 and 2. Each LCB score is equal to the sum of its constituent alignment scores. Panel **C**: Pairwise anchoring of genomes 1 and 2. The anchoring score penalizes the initial anchor configuration for two breakpoints, worth 1,500 each, for a total anchor score of 10,000+10,000+1,000-2x1,500 = 18,000. We then consider the effect of removing each LCB on the anchoring score. Removal of {AB} would eliminate a single breakpoint and result in a total anchor score of 9,500 because A and B no longer contribute 5,000 each to the score. Removal of {CD} would eliminate a single breakpoint, also giving a total anchor score of 9,500. Removal of {E} would eliminate two breakpoints and give a total anchor score of 18,500. We remove {E} because it improves the anchor score from 18,000 to 18,500. We create the consensus alignment path shown in blue which corresponds to the ancestor of 1 and 2 in the guide tree. The removal of E corresponds to splitting the node into separate nodes per-genome (labeled E1 and E2) in the blue consensus path.

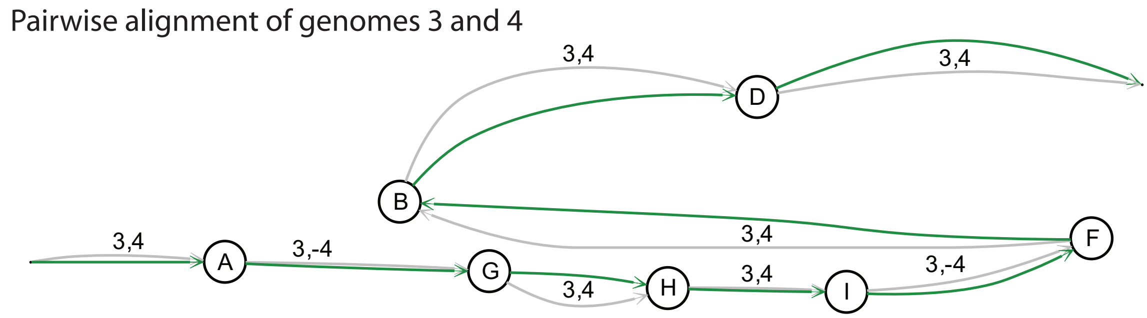

Figure 14: Panel **A**: Pairwise anchoring of genomes 3 and 4. The full graph shown in Figure 12 is projected to the subgraph containing only edges labeled 4 and 5. The inversion of matches G, H, and I in genome 4 induces three pairwise LCBs: {A}, {GHI}, and {FBD}, scoring 5,000, 3,000, and 11,000, respectively. Each of the two breakpoints come with a penalty of 1,500, for a total anchoring score of 16,000. Removing any of the three LCBs fails to increase the anchoring score, so the anchors remain identical to the initial set of local alignments between genomes 3 and 4.

refinement, the alignment is divided into non-overlapping windows, each of which is refined separately. Figure 11 panels d and h show window-based iterative refinement for a given alignment tree node n. Regions aligned with few gaps may be refined in windows of 500 or 200 alignment columns. When a region of the existing alignment is ambiguous, containing many gaps, we select a window size of 20,000 alignment columns. The relatively large window size gives MUSCLE greater latitude in shifting gaps to identify optimal the alignment. These window sizes were chosen empirically to provide a reasonable trade-off between speed and accuracy (data not shown).

Identification of segments conserved among two or more genomes

The MUSCLE global alignment program dutifully finds the highest-scoring alignment between alignment anchors, regardless of whether the intervening region contains homologous sequence. Occasionally non-homologous regions become aligned as a side effect of forced global alignment in regions between anchors. In order to identify and remove

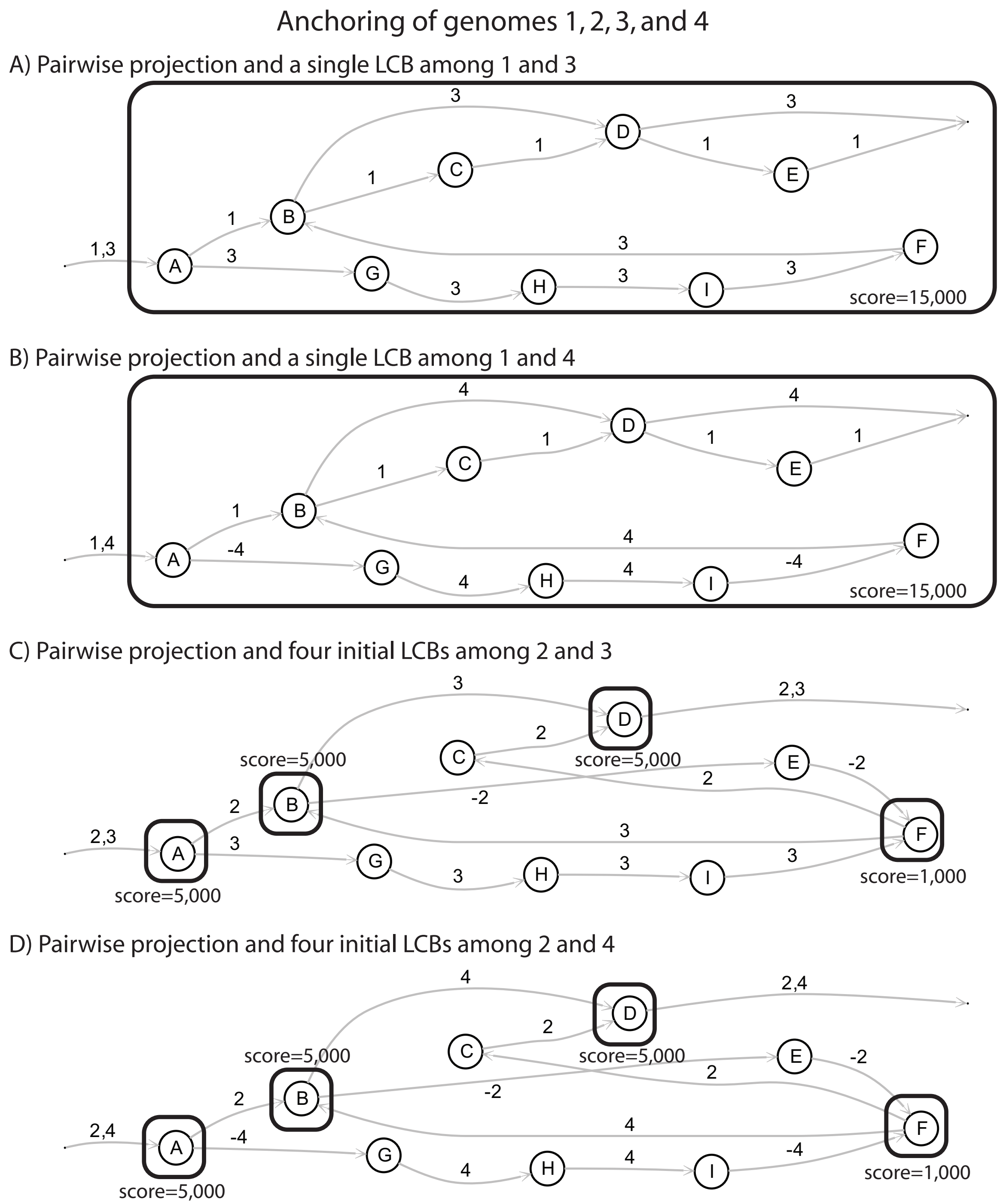

Figure 15: Panel **A**: The pairwise projection of the graph in Figure 12 to genomes 1 and 3 has no cycles or inverted segments, yielding a single LCB. The LCB has score 15,000 and since no breakpoints exist, the pairwise anchoring score for 1,3 is 15,000. Panel **B**: The pairwise projection to genomes 1 and 4 also has a single LCB with score 15,000. Panel **C**: The pairwise projection to genomes 2 and 3 has a cycle among nodes B, E, and F. The cycle induces four pairwise LCBs: {A},{B},{D}, and {F} with scores 5,000, 5,000, 5,000, and 1,000 respectively. The initial anchor configuration is penalized for three breakpoints, giving a total pairwise anchor score of 11,500. Panel **D**: The pairwise projection to 2 and 4 also has a cycle inducing four LCBs. The pairwise anchoring score for 2 and 4 is also 11,500.

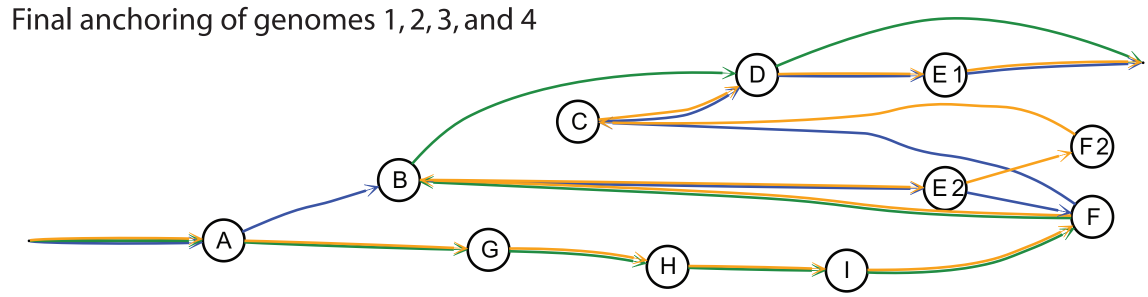

Figure 16: To arrive at a final anchoring for genomes 1, 2, 3, and 4, we apply sum-of-pairs greedy breakpoint elimination to the pairwise projections shown in Figure 15. The projections 1,3 and 1,4 have no breakpoints, and thus no breakpoint elimination can be applied. Projections 2,3 and 3,4 each have four LCBs. We compute the total SP anchoring score as the sum of each pairwise anchoring score: 15,000 + 15,000 + 11,500 + 11,500 = 53,000. We then evaluate whether removal of any pairwise LCB would improve the total SP anchoring score. We arbitrarily choose to consider LCBs from the 2,3 projection first. Removing {A} would result in a reduction from four to three breakpoints, and a loss of the 5,000 points contributed by {A} to the projection of 2,3. Because we impose transitive homology, we must also remove {A} from the pairwise projections 1,3 and 1,4 and 2,4 if we remove it from 2,3. Thus the total SP anchoring score with {A} removed becomes 10,000+10,000+8,000+8,000=36,000. We do not remove {A} because the SP anchoring score would decrease. Removing {D} has the same effect on the SP anchoring score as removal of {A}. Next, we evaluate removal of the LCB {F}. Removal of {F} would eliminate the cycle in the projection of 2,3, resulting in a single pairwise LCB with score 15,000. Again, if {F} is removed from 2,3 it must also be removed from all other pairwise projections, namely 2,4 (but not 3,4). The total SP anchoring score after removing {F} would be 15,000+15,000+15,000+15,000=60,000. Finally, we consider removal of {B} from projection 2,3. Removal of {B} also eliminates the cycle in 2,3 and would give a total SP anchoring score of 15,000+15,000+11,000+11,000=52,000. Because projections 2,3 and 2,4 have identical LCBs, we need not consider the score impact of removing LCBs from 2,4. At this point, we remove the LCB which offers the largest increase in the SP anchoring score: {F}. After removal of {F}, the SP anchoring score can no longer be improved and we arrive at the final anchoring depicted above as a gold-colored path. Notice that F does not form an anchor among genomes 2,3 and 2,4, but it remains a valid pairwise anchor among 3,4 and is included in the golden path.

aligned, non-homologous regions we apply random-walk statistics to the HOXD substitution and affine gap score (Chiaromonte et al., 2002, Schwartz et al., 2003). Nucleotide substitution scoring matrices are log-ratio estimates of the probability that a pair of nucleotides are homologous, versus the probability they are non-homologous. The substitution and affine gap score are designed to assign high scores to homologous regions and low scores to non-homologous regions. Random walk statistics require a score function that will be negative on average, however, aligned LCBs typically contain high sequence identity, so the substitution score is a very large positive number on average. Thus, we invert the log ratios and multiply the affine gap penalties by -1, which causes homologous LCBs to have a negative score on average. We can then apply random walk statistics to identify high-scoring segments indicative of a non-homologous region.

We performed simulation studies to select an appropriate significance threshold for random-walk excursions. Specifically, we simulated molecular evolution among a pair of sequences under the HKY85 model with 0.75 substitutions per site, Ts/Tv ratio=4, gamma-distributed rate heterogeneity (shape=1), and 0.05 indels per site with lengths sampled from a Poisson with intensity 3. These parameters were selected to be at or beyond the outer limits of sequence alignable by our method. We performed 200 simulations of sequences with average length 1,000,000 nt. Scoring the simulations yields 42,429,635 excursions which indicate a 99.9% threshold score of 2727 in the extreme value distribution, and a 99.99% threshold of 4076.

We identify boundaries of non-homologous sequence as regions where the score of a random-walk excursion exceeds our score threshold. Given the boundaries of pairwise segments likely to be non-homologous, we compute the complementary boundaries of pairwise segments likely to be homologous. We then apply the notion of transitive homology (Szklarczyk and Heringa, 2004, 2006) by finding the union of all overlapping pairwise homologous segments. We refer to the resulting segments as "backbone." The regions complementary to the "backbone" are genome-specific "islands" of sequence content. We unalign any aligned regions that lie outside a backbone segment.

5.3 Results

The Progressive Mauve alignment algorithm results in a multiple genome alignment where any nucleotide is aligned to at most one other nucleotide. After filtration of non-homologous segments, the remaining aligned regions are typically either mono-topoorthologous (Dewey and Pachter, 2006) or xenologous (Fitch, 2000), and rarely paralogous or non-homologous. In addition to predictions of homologous nucleotides, Progressive Mauve predicts the endpoints of segmental homology among each pair of genomes. Finally, the algorithm also predicts the boundaries of genome-specific sequence and sequence conserved in two or more of the genomes under study, which we refer to as *backbone* sequence.

5.3.1 An alignment of enterobacteria

We apply the progressive genome alignment method to two groups of enteric bacteria: a set of 12 *E. coli* and *Shigella* genomes (described presently), and a set of 9 genomes of Enterobacteriacae (described in Chapter 8. The alignment of 12 *E. coli* genomes consumes approximately 12 hours of computation and 6GB memory on an AMD Opteron workstation. A visualization of the resulting alignment is shown in Figure 17. The final alignment consists of 355 LCBs of minimum length 28, which constitute a total of 12.0

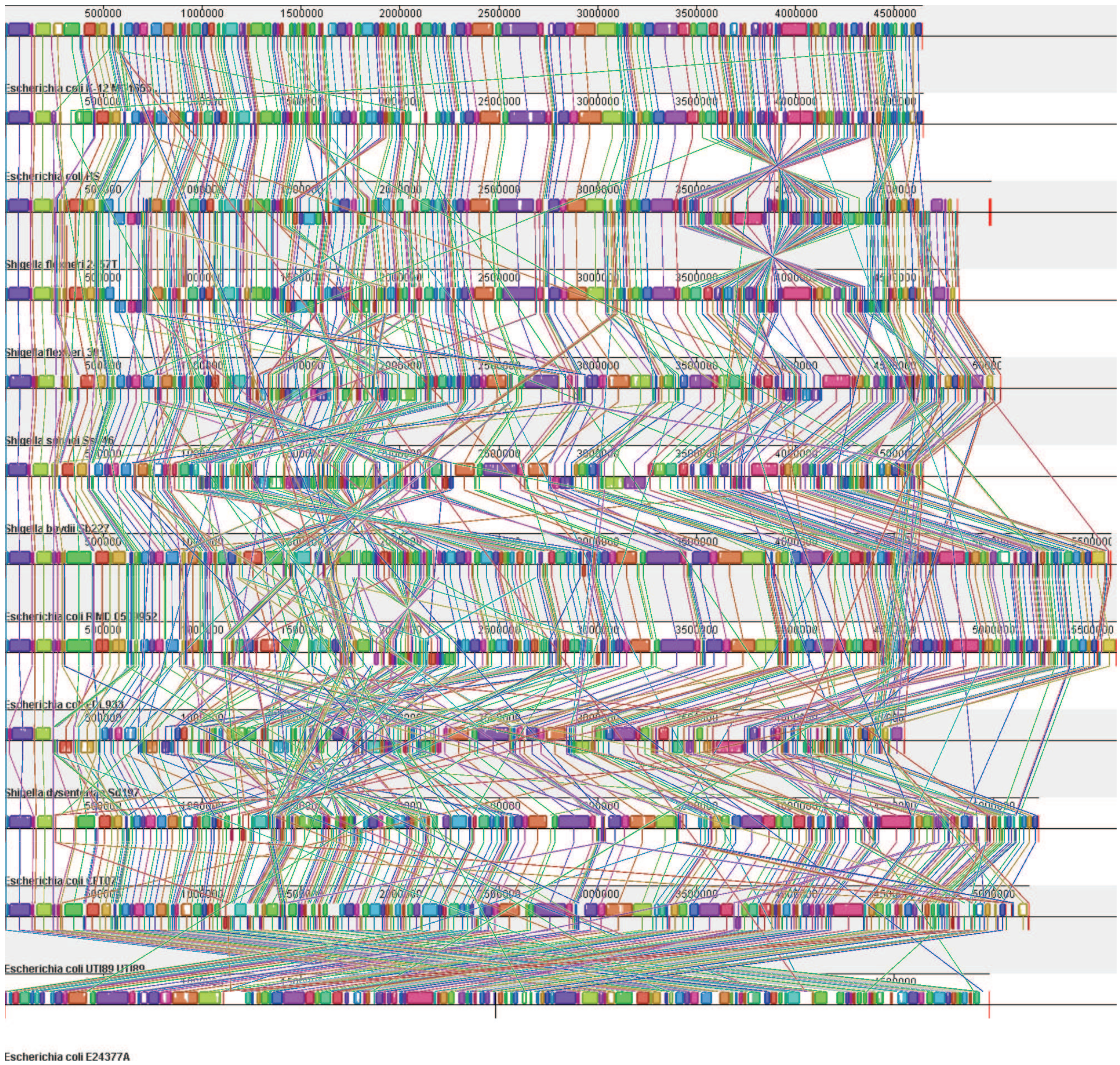

Figure 17: An alignment of 12 *E. coli* genomes reveals 355 well-supported locally collinear blocks and substantial amounts of lineage-specific sequence. Each genome is laid out on a horizontal track. Colored blocks indicate segmental homology, with lines connecting orthologous LCBs across genomes. Blocks shifted below a genome's center axis are in the reverse complement orientation relative to the reference genome. Crossing LCB-connecting lines indicate that a rearrangement has taken place. The circular genome of *E. coli* E24377A, shown at bottom, appears to have been linearized at a different point than the other genomes, resulting in a large number of crossing LCB-connecting lines.

Mbp of unique sequence. The *E. coli* appear to have undergone substantial amounts of gene flux, and some isolates, particularly *Shigella* isolates, appear to be undergoing rapid genome rearrangement.

5.3.2 Interactive visualization

We have developed an interactive visualization tool to assist exploration and interpretation of the alignments generated by our method. The Mauve visualization environment enables inspection of multiple genome alignments at all scales, from a global display of comparative genome architecture to detailed inspection of nucleotide substitution. As shown in Figure 18, each aligned genome is displayed on a horizontal track composed of a sequence similarity plot and annotated sequence features. The viewer reads and displays annotated sequence features from GenBank format flat files using the BioJava library. The sequence similarity plot shows segmental homology as round rectangles (blocks), with an average sequence identity plot inside the rounded rectangle.

The height of the sequence identity plot reflects the average column entropy for the region of the alignment covered by a column of display pixels. Specifically, the similarity plot height is directly proportional to a similarity value $s(\mathcal{A}, g, i)$ which we define as follows. Consider the alignment $\mathcal{A}$ as a $G \times C$ matrix, where each of the G rows corresponds to a genome and there are C columns. Each matrix entry is an element in the alphabet $\{A, C, G, T, -\}$. To calculate the similarity for a given genome $g \in \mathbf{G}$, we project $\mathcal{A}$ to the submatrix $\mathcal{A} : g$, which is the submatrix formed by removing all columns where the entry for genome g is a gap $(-)$. The similarity value for position i of g can then be calculated as:

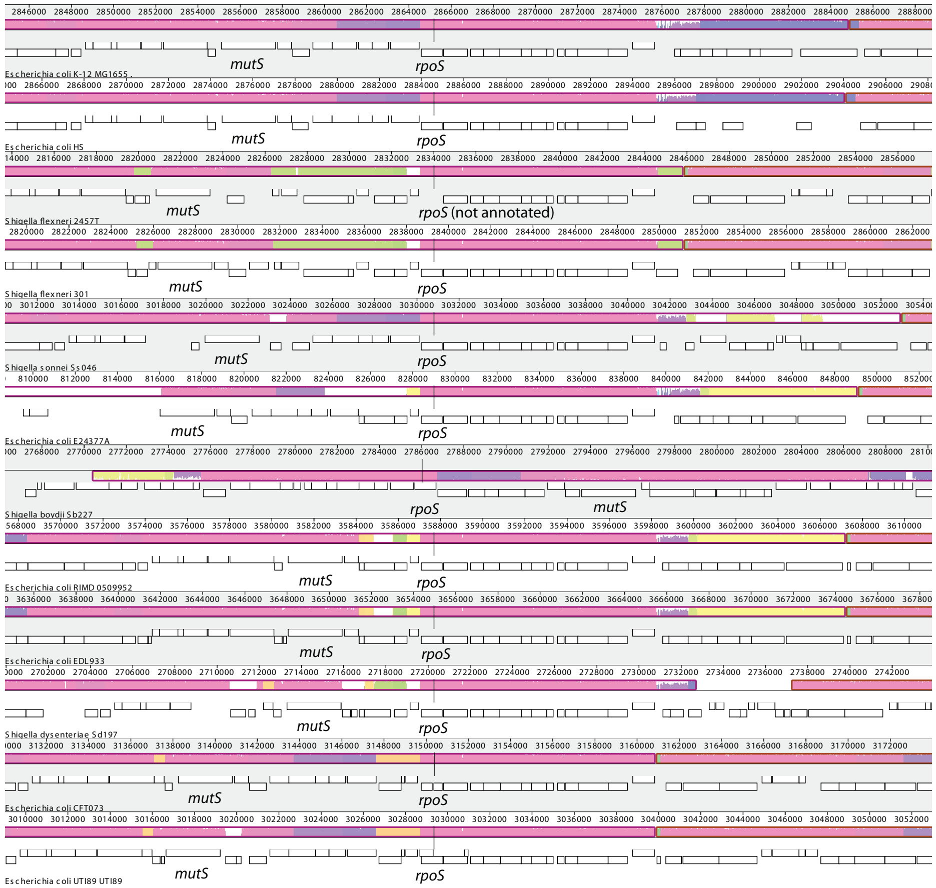

Figure 18: A detailed view of the hypervariable region between the genes *mutS* and *rpoS* in *E. coli* K12. In the region between *mutS* and *rpoS*, several each taxa have acquired an alternative set of non-homologous genes. We refer to such non-homologous genes surrounded by conserved orthologous genes as *alternalogs*. A black rectangle outlines the region containing alternalogs in the figure, and colors on the similarity plot indicate the taxon groupings of segments that are conserved among two or more genomes. Most importantly, mauve-colored segments are conserved among all taxa. The blue segments are conserved among K12, HS, CFT073, UTI89, and E24377A. Goldenrod segments are specific to the uropathogenic CFT073 and UTI89 isolates. Bright yellow segments are conserved between EDL933 and RIMD, and alternatively *S. sonnei* and *S. boydii*. Light green segments are conserved among the two *S. flexneri*, while medium green segments are conserved between *S. flexneri*, *S. dysenteriae*, EDL933, and RIMD. The observed pattern of segmental homology appears to result from a combination of intraspecific recombination and differential gene loss.

$$
\begin{aligned}
s(\mathcal{A}, g, i) &= 1 - \sum_{j=i-(\omega/2)}^{i+\omega/2} \frac{H(\mathcal{A}:g, j)}{\omega} \\
H(\mathcal{A}, g, j) &= - \sum_{a \in \{A,C,G,T\}} \frac{count(a, \mathcal{A}, g, j)}{|\mathbf{G}|} \log_2 \frac{count(a, \mathcal{A}, g, j)}{|\mathbf{G}|} \\
&\quad - \frac{count('-', \mathcal{A}, g, j)}{|\mathbf{G}|} \log_2 \frac{1}{|\mathbf{G}|} \\
count(a, \mathcal{A}, g, j) &= \sum_{k=1\ldots|\mathbf{G}|} 1_{(\mathcal{A}:g)_{k,j}=a}
\end{aligned}
$$

where ω is a constant sliding window size, defaulting to 5nt. The function $count(a, \mathcal{A}, g, j)$ counts the number of times character a occurs in column j of $\mathcal{A} : g$. The function $H(\mathcal{A}, g, j)$ effectively computes the Shannon entropy of alignment column j in the submatrix $\mathcal{A} : g$, with slight modification to consider each gap $'-'$ as a different character. This modification causes a column of all gaps or nearly all gaps to have high entropy, implying poor sequence conservation. Without the modification, heavily gapped alignment columns would appear to be well conserved. When information about the location of conserved "backbone" segments is available, we further modify the equations above to compute similarity only on the subset of genomes in which the segment is considered to be conserved. Finally, when a single display pixel covers a range of sequence coordinates $x \ldots y$, we display the average similarity plot height for that pixel, computed as:

$$
\sum_{i=x}^{y} \frac{sim(\mathcal{A}, g, i)}{y - x}
$$

It is worth noting that ω may be set to 0 so that the display of average similarity does not use sliding windows to smooth the similarity peaks. Numerous problems exist with analyses based on sliding window methods, although for the type of exploratory data analysis presented by the Mauve viewer, use of a sliding window should not pose a problem.

5.4 Discussion

Multiple alignments of genomes with rearrangement and lineage-specific sequence may provide evidence for ancestral rearrangement events that are undetectable with pairwise comparisons of extant sequences. The simplest scenario for which an ancestral rearrangement can be detected in a multiple alignment, but not among pairwise alignments is shown for three genomes in Figure 19.

The alignments produced by our method serve as a foundation for further study into all aspects of genome evolution. Both deterministic (Bourque and Pevzner, 2002, Tang and Moret, 2003) and Bayesian (Miklos, 2003, Larget et al., 2002) methods for inference of genome rearrangement histories may directly use the LCB predictions as input. A challenge exists, however, because such methods typically assume that orthologous segments are present in all genomes under study. Alignments produced by Progressive Mauve frequently contain segments conserved in only a subset of the organisms under study, presumably due to differential gene loss or acquisition via lateral transfer. Bayesian methods for inference of gene content evolution via loss and lateral transfer have recently been proposed (Csuros and Miklos, 2006), but work remains to integrate such models with a model of genome rearrangement.

In addition to supporting studies of genomic rearrangement, our multiple genome alignments enable genome-wide study of recombination patterns and selective forces.

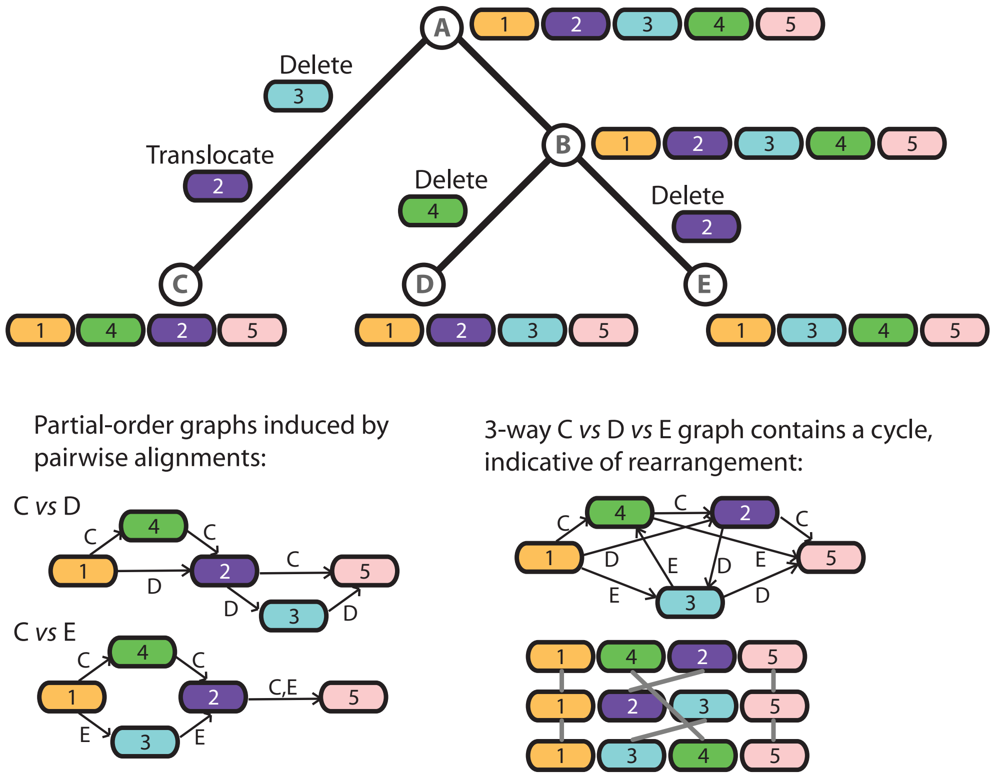

Figure 19: Some genome rearrangement events may be undetectable using pairwise comparisons, but revealed through multiple genome comparison. The common ancestor (A) of extant genomes C, D, and E has five genes, numbered 1 through 5. A transposition occurs on the branch from A to C, but the transposition is not observable in pairwise comparisons between C, D, and E due to differential gene loss. A simultaneous comparison of C, D, and E reveals the rearrangement as a cycle in the alignment graph.

Several studies of recombination and selection among microbial genomes have been published to date, however the majority have focused only on genic regions, ignoring important non-coding sequence (Chen et al., 2006).

Acknowledgments

Progressive Mauve was conceived on a couch in Barcelona.

Chapter 6

Evaluating alignment accuracy

Without a 'correct' alignment of the enteric genomes, the alignments calculated by the previously described methods can not be evaluated for accuracy. Although several benchmark data sets exist for protein sequence alignment (Thompson et al., 1999, Edgar, 2004), no such benchmark data sets exist for the genome alignment task. Construction of an alignment accuracy benchmark would require manual curation of a whole-genome multiple alignment that includes rearrangement and lateral gene transfer, a task that to date has proven too time-consuming and difficult. Despite the lack of a manually curated correct alignment, we can estimate the alignment accuracy by modeling evolution and aligning simulated data sets.

The inferential power yielded by using simulated evolution to evaluate alignment accuracy is only as strong as the degree to which the simulation faithfully represents the evolutionary processes that produce naturally occurring genomes of interest. Keeping that fact in mind, we constructed a simplistic model of genome evolution that we believe captures the major types, patterns, and frequencies of events in the history of the enteric genomes. Given a rooted phylogenetic tree and an ancestral sequence we would like to generate evolved sequences for each internal and leaf node of the tree, along with a multiple sequence alignment of regions conserved throughout the simulated evolution. To effectively represent genome evolution, the simulation must include nucleotide substitutions and indels in addition to genome scale events such as horizontal transfer, inversion, and rearrangement.

Nucleotide substitutions are ostensibly the best studied and most ubiquitous mutation process. We use the HKY85 (Hasegawa et al., 1985) model of nucleotide substitution implemented in the Monte-Carlo simulation package called Seq-gen (Rambaut and Grassly, 1997). We apply a Transition/Transversion ratio of 4 and gamma-distributed rate heterogeneity with shape parameter $\alpha = 1$. Small insertions and deletions (indels) are modeled as occurring with uniform frequency and distribution throughout the genomes, with a size sampled from a Poisson distribution with mean value 3bp. When studying the differences between *E. coli* O157:H7 EDL933 and K-12 MG1655 (Perna et al., 2001), it became clear that a small number of horizontal transfers introducing large regions of sequence have occurred, while the majority of transfers introduced small sequence regions. Our model includes large horizontal transfer events uniformly distributed in length between 10Kbp and 60Kbp. The size of small horizontal transfer events is sampled from a geometric distribution with mean value 200bp. Horizontal transfer is implemented by simultaneously evolving a set of 'donor' genomes from which horizontally transferred sequence can be sampled.

Using the observation that two overlapping inversion events can result in a translocation, our model does not explicitly implement translocation events. The length of inversions are sampled from a geometric distribution with mean value 50Kbp. Locations for inversion and horizontal transfer events are sampled uniformly throughout the genome, and all events are simulated to have taken place at a point in time given by a marked Poisson process over the phylogenetic tree. Finally, genome size is expected to stay relatively constant over time, so deletion events are sampled with the same size and frequency as events that introduce new sequence. Our implementation of the evolutionary model described above is referred to as the simple genome evolver, or just `sgEvolver`.

6.1 Alignment scoring

We score the calculated alignments against the correct alignments generated during the evolution process. Previous studies of alignment accuracy have used a sum-of-pairs scoring scheme to characterize the nucleotide level accuracy of the aligner (Thompson et al., 1999, Darling et al., 2004a). The experiments presented here use sum-of-pairs scoring, but we also define several new accuracy measures intended to quantify each alignment system's ability to detect segmental homology and predict breakpoints of genomic rearrangement. We treat nucleotide alignment accuracy more precisely by defining criteria for True Positive, False Positive, and False Negative alignments, allowing us to characterize both sensitivity (recall) and positive predictive value (precision) of each method. A summary of the scoring metrics appears in Table 4 and full definitions follow.

For nucleotide-level alignment accuracy metrics, we classify each pair of nucleotides aligned in a calculated alignment as either True Positive (TP), False Positive (FP), or False Negative (FN). A True Positive is a pair of nucleotides aligned in the calculated alignment that also appear in the correct alignment. A False Positive is a pair of nucleotides aligned in the calculated alignment that is not found in the correct alignment. A False Negative is a pair of nucleotides aligned in the correct alignment which were not aligned in the calculated alignment. We do not quantify True Negative (TN) alignments,

Nucleotide Sensitivity	TP / (TP + FN)	The fraction of correctly aligned nucleotide pairs in the calculated alignment.
Nucleotide PPV	TP / (TP + FP)	The fraction of nucleotide pairs correctly aligned in the calculated alignment, out of the total nucleotide pairs aligned in the calculated alignment.
LCB Sensitivity	TP / (TP + FN)	The fraction of LCBs in the correct alignment that had at least one correctly aligned pair of nucleotides in the calculated alignment.
LCB PPV	TP / (TP + FP)	The fraction of LCBs in the calculated alignment that had at least one correctly aligned pair of nucleotides.
Breakpoint localization	-	The distance between the predicted breakpoint of rearrangement and the true breakpoint of rearrangement.

Table 4: A summary of the scoring metrics used to evaluate accuracy of genome alignments

as there are exponentially many TN possibilities.

We also quantify the ability of each aligner to correctly identify orthologous segmental homology in the form of Locally Collinear Blocks (LCBs). For each possible pair of genomes we measure whether the aligner finds LCBs among that pair, yielding a sum-of-pairs LCB accuracy metric. When an aligner correctly aligns at least one pair of nucleotides in an LCB, we consider the LCB as correctly found in the corresponding pair of genomes (True Positive). Pairwise LCBs in the correct alignment which have no correctly aligned pairs in the calculated alignments are considered not found (False Negative). Any pairwise LCB in the calculated alignment that contains no correctly aligned positions is considered to be a False Positive. As with the nucleotide accuracy metric, there are exponentially many True Negative LCB predictions which we do not report.

Finally, we quantify how well each aligner localizes the exact breakpoint of rearrangement. When an LCB is correctly predicted in the calculated alignment, we record the difference between the boundary coordinates of the correct LCB and those of the calculated LCB. When the difference is negative, the calculated alignment has underpredicted the boundary, i.e. the calculated LCB does not extend to cover the full region of homology. A positive difference indicates an overprediction, where the calculated LCB includes additional sequence beyond the end of the segmental homology. We report mean, standard deviation, and quantile statistics for LCB boundary predictions.

6.2 Experiments

Using the simple genome evolver, we designed and executed experiments to compare the performance of several genome alignment systems under a variety of mutational regimes. Multiple alignment experiments used a phylogenetic guide tree estimated for a group of nine *E. coli*, *Shigella*, and *Salmonella*, midpoint rooted to provide an entry point for the ancestral sequence. Figure 20 shows the topology and branch lengths of the tree used for our simulation studies. Rather than generate a random ancestral sequence, we used DNA randomly sampled from an enteric genome in order to preserve the distribution of sequence motifs and repetitive subsequences found in naturally occurring genomes. Additional enteric DNA was sampled for use as a donor sequence pool for insertion and horizontal transfer events. Both samplings are without replacement, i.e. the ancestral target sequence and the ancestral donor sequences are never identical to each other.

We processed all evolution simulations and genome alignments using the Condor high throughput computing environment at the University of Wisconsin. The Wisconsin Condor cluster contains over 1000 compute nodes and allowed us to rapidly align thousands of simulated data sets.

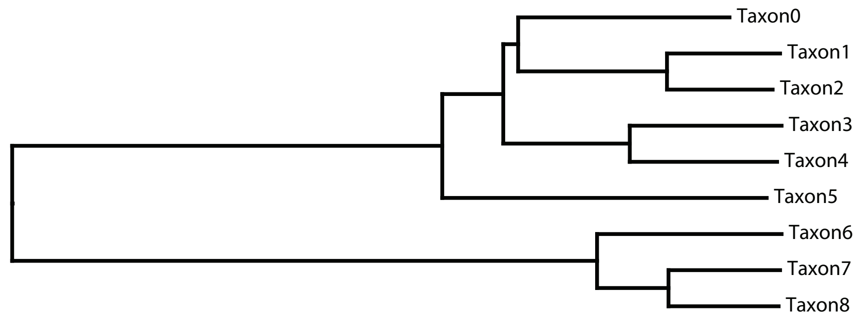

Figure 20: A phylogenetic tree relating the nine enteric genomes studied in Chapter 4. The tree was calculated using Neighbor-Joining on a genome-content distance metric. The unrooted tree has been midpoint-rooted for simulation studies.

6.2.1 Experiment: genomes without rearrangement

Our first experiment compared the ability of the original Mauve, Multi-LAGAN version 1.2, Mavid version 0.9, Mauve 1.3.0, and Progressive Mauve to align collinear sequences that had undergone increasing amounts of nucleotide substitution and indels. This experiment is designed to test the sensitivity of the anchoring methods employed by each aligner. We simulated evolution of nine genomes at 20 increasing nucleotide substitution rates and 20 increasing indel rates, performing 3 replicate experiments of each combination of substitution rate and indel rate.

Each aligner's average sensitivity for each simulation is displayed in Figure 21. From the figure, it is obvious that the original Mauve implemention's alignment score drops precipitously in the presence of an increasing substitution rate. The improved version of Mauve which uses approximate multi-MUM anchors (versions 1.0 and later) performs substantially better than the original Mauve, but still falls short of Mavid and Multi-LAGAN at high mutation rates. We attribute this behavior to Mauve's requirement that the multi-MUM anchors be present in all genomes under study. Multi-LAGAN's alignment anchors can contain substitutions and indels, and must only align pairs of

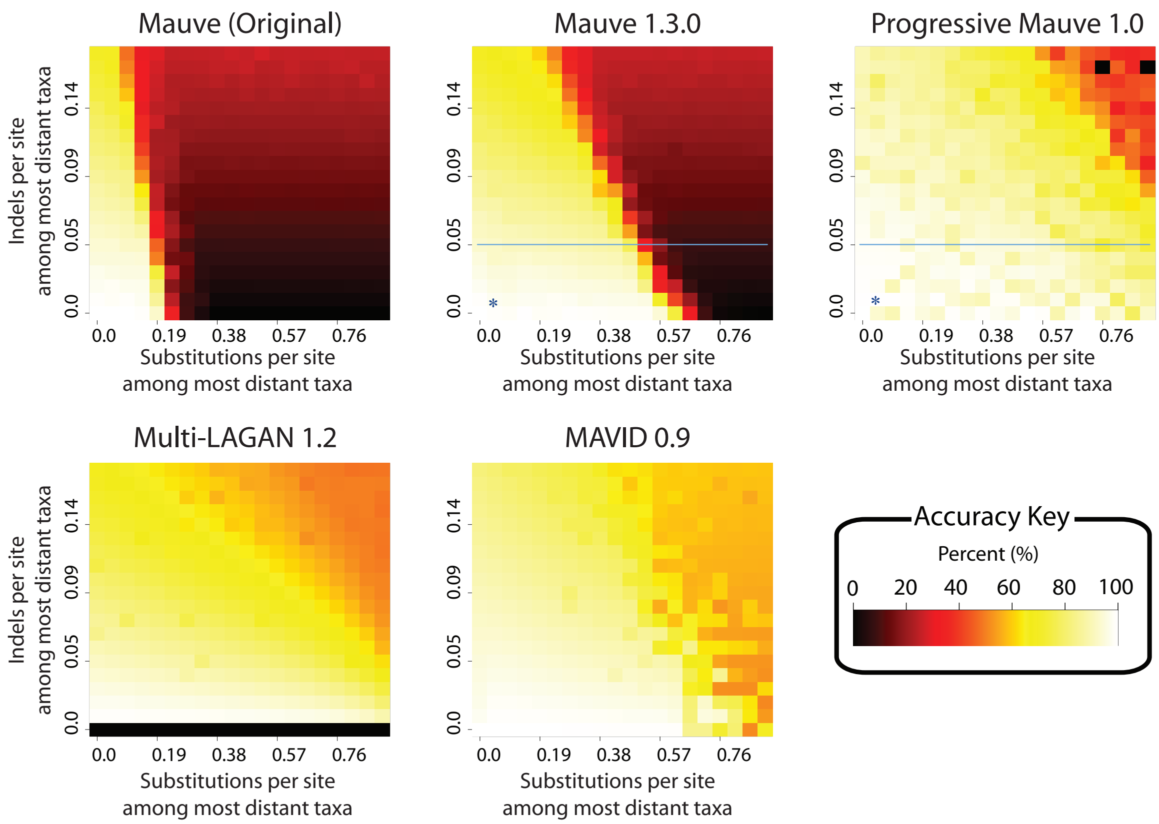

Figure 21: The sensitivity of Mauve(1), Multi-LAGAN(2), Mavid(3), Mauve 1.3.0 with spaced seeds(4), and Progressive Mauve(5) when aligning sequences evolved with increasing amounts of nucleotide substitution and indels. The exact match anchoring technique employed by the original Mauve implementation limits its ability to align distantly related sequences. The more recent Mauve 1.3.0 implementation uses approximate multi-MUMs as alignment anchors, and performs substantially better. Multi-LAGAN version 1.2 did not complete the alignments of genomes without indels, resulting in the black row at the bottom. The performance of Progressive Mauve is comparable to that of Multi-LAGAN and Mavid 0.9, outperforming these methods for certain combinations of indel and substitution rate. The thin blue line indicates the combination of indel and substitution rates that were subsequently used for tests measuring aligner robustness to inversion (Figure 24). The asterisk(*) indicates the combination of indel and substitution rates used for tests measuring aligner robustness to gene flux (Figure 25).

genomes, making them much more sensitive. Mavid appears to perform better than Multi-LAGAN at very high mutation rates, probably owing to its method of inferring ancestral states along a phylogeny and using those to compute alignment anchors. Progressive Mauve uses a progressive alignment anchoring approach, allowing it to utilize anchors present in as few as two genomes. The progressive approach provides a substantial boost in anchoring sensitivity and the performance of Progressive Mauve is similar to that of Mavid and Multi-LAGAN. For the nucleotide substitution and indel rates previously reported in the enteric data set, Mauve aligns the simulated genomes with a high degree of sensitivity.

We do not report LCB accuracy metrics for this experiment because the genomes were evolved under a model that did not include genomic rearrangement.

6.2.2 Experiment: pairs of genomes with rearrangement

We proceeded to gauge the ability of the original Mauve implementation and Shuffle-LAGAN version 1.2 to align sequences that had undergone increasing amounts of inversion and nucleotide substitution. Because Shuffle-LAGAN is a pairwise aligner, we reduced the number of taxa in our simulation from 9 to two. Three simulations were performed for each of 110 combinations of nucleotide substitution rate and inversion rate. The average nucleotide sensitivity of Mauve and Shuffle-LAGAN for each experiment are shown in Figure 22. Special considerations must be taken when scoring Shuffle-LAGAN. Because Shuffle-LAGAN attempts to identify and align both orthologous and paralogous regions but does not distinguish orthology from paralogy, a single residue in the first genome can be ambiguously aligned to multiple residues in the second genome. For the purpose of scoring Shuffle-LAGAN, we award points for correctly aligned nucleotide

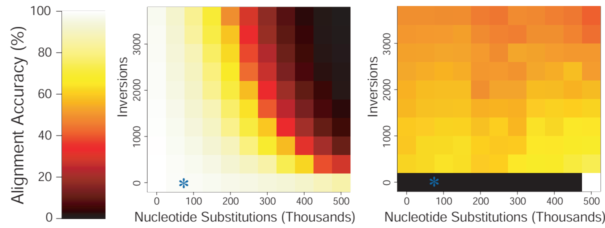

Figure 22: The performance of Mauve(left) and Shuffle-LAGAN(right) when aligning two sequences evolved with increasing amounts of nucleotide substitution and inversions. Mauve is clearly more accurate than Shuffle-LAGAN at lower substitution rates. Shuffle-LAGAN version 1.2 did not complete some alignments without rearrangements, resulting in black entries. The rate of substitution and inversion observed between *E. coli* and *Salmonella* is denoted by an asterisk(*).

pairs if the pair appears in anywhere in the alignment, even if the positions have been aligned to other, non-orthologous residues.

The experiment shows that the original Mauve implementation clearly excels at aligning rearranged sequences under lower substitution rates that do not hamper its anchoring process. Interestingly, Shuffle-LAGAN appears to perform better as the substitution rate increases. Based on our experience, we conjecture that this counter-intuitive result is related to the repetitive nature of the ancestral enterobacterial sequence. Shuffle-LAGAN appears to have difficulty selecting anchors in repetitive sequences. As the nucleotide substitution rate increases, regions that were repetitive are randomly mutated and thus no longer repetitive. Anchoring its alignment in unique subsequences provides Mauve with immunity to this phenomena.

We do not report LCB scoring metrics for this experiment because Shuffle-LAGAN does not distinguish between orthologous and paralogous segmental homology.

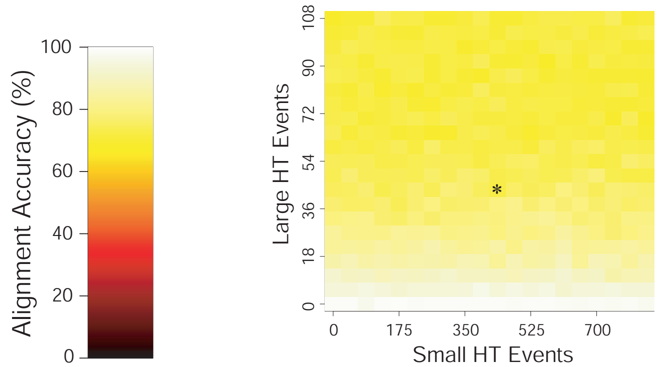

Figure 23: The performance of the original Mauve implementation when aligning sequences evolved with rates similar to those observed among a group of *E. coli* and *Salmonella* genomes. In this experiment, the substitution, indel, and inversion frequencies were held constant while the rates of small and large gene flux were modulated. The asterisk denotes the combination of large and small gene flux rates observed expected between *E. coli* and *Salmonella*. As the rate of large horizontal transfer increases the amount of lineage-specific sequence relative to backbone grows. Because Mauve can not align large lineage-specific regions the alignment sensitivity score drops. When scored only on regions considered backbone sequence the sensitivity is consistently above 98%.

6.2.3 Experiment: enterobacteria-like genomes

Our third set of experiments sought to evaluate the ability of Mauve to align genomes similar to the enterobacteria. Evolutionary rates for the simulation were extrapolated from previously published observations of the differences between *E. coli* K-12 MG1655 and O157:H7 EDL933 (Perna et al., 2001). For these two *E. coli*, there are about 75,000 observed nucleotide substitutions, about 4,000 observed indels, 40 large horizontal transfer events, 400 small horizontal transfers, and one inversion. The observed frequencies were converted to rates used to assign event frequencies to branches of the phylogenetic guide tree. It is known that among the group of enterobacteria, the *Salmonella* have higher rates of inversion and rearrangement than the *E. coli*. To compensate, the inversion rate was adjusted to result in approximately 30–40 inversion events. When varying the substitution and indel rates between 0 and 125% of the observed rates while holding horizontal transfer and inversion rates constant, Mauve alignments consistently average 80% sensitive, ± 5% (data not shown). The quality of alignment does not appear to drop as the substitution and indel rates are increased in this range. Rather, it appears that horizontal transfer rates have a more significant impact on alignment quality. As horizontal transfer rates increase, the ratio of lineage-specific sequence to backbone sequence increases and Mauve's alignment algorithm aligns decreasing amounts of the total sequence. When varying simulated horizontal tranfser rates between 100 and 200% of previously reported rates for the enterobacteria, Mauve consistently aligns with about 65% sensitivity (Figure 23). When scored only against regions of the simulated genomes considered as conserved backbone, Mauve consistently aligns with >98% sensitivity. For the purpose of scoring the alignment, we define backbone as a region in the correct alignment containing more than 50 gap-free columns without stretches of 50 or more consecutive gaps in any single genome sequence. Based on our simulations we believe the original Mauve alignment method accurately aligns regions conserved among all genomes under study, however, significant lineage-specific regions remain unaligned.

6.2.4 Experiment: high rates of rearrangement

We assessed the relative performance of Mauve 1.3.0 and Progressive Mauve when aligning genomes with high rates of genomic rearrangement and nucleotide substitution. We performed three replicates of simulated evolution at 10 increasing substitution rates and 10 inversion rates. In addition to quantifying sum-of-pairs nucleotide sensitivity, we also quantified positive predictive value and LCB accuracy on this data set. The results, shown in Figure 24, indicate that Progressive Mauve can accurately align genomes with substantially higher rates of rearrangement than previous Mauve implementations.

6.2.5 Experiment: high gene flux rates

Some bacteria have been demonstrated to rapidly acquire novel gene content from other microbes (Friedrich et al., 2001, Hsiao et al., 2005), thus we would like to know how well our alignment methods perform in the face of substantial acquisition and loss of genetic material (gene flux). We characterized the accuracy of Mauve 1.3.0 and Progressive Mauve when aligning genomes simulated to undergo high rates of both small- and large-scale gene flux, in addition to modest rates of substitution, indels, and rearrangement. We use an ancestral sequence of 500,000nt.

The results, shown in Figures 25, indicate that the algorithm used by Mauve 1.3.0 falters when faced with large-scale gene flux, while Progressive Mauve performs significantly better. Both Mauve and Progressive Mauve tolerate the small-scale gene flux–modeled here as insertions and deletions of sequence with geometrically distributed average lengths of 200nt. As the rates of gene flux increase, the probability that any given pair of genomes share orthologous sequence deteriorates and eventually reaches zero in the limit of an infinitely high rate of gene flux.

6.3 Simulated phylogenetic ladders

A common experimental design in comparative genomics studies involves sequencing the genomes of a group of organisms believed to have a phylogenetic relationship that approximates a so-called phylogenetic ladder (Clark et al., 2003, Thomas et al., 2003). Such experimental designs typically aim to identify genomic regions that are conserved at increasing levels of sequence divergence. A benefit of sequencing phylogenetic intermediates in a ladder-type experiment is that multi-genome comparisons may allow nucleotide homology to be identified among pairs of organisms that are too divergent for pairwise comparison.

We attempt to gauge the ability of our alignment algorithm to exploit additional information available by sequencing phylogenetic intermediates between two divergent organisms. Beginning with two divergent taxa (a and q in Figure 26), we simulate genome evolution with rearrangement, horizontal transfer, nucleotide substitution, and indels. The sensitivity of our method in aligning the pair of simulated genomes for a variety of branch lengths is given in Figure 27A. We then add a single taxon which evenly splits the branch from the root to taxon a and evaluate the alignment sensitivity. We continue by repeatedly adding taxa at points which evenly divide the previous taxa into a phylogenetic ladder with increasing resolution. The alignment sensitivity results for ladders with 0, 1, 3, 7, and 15 taxa in addition to a and q are shown in Figure 27.

Rather than evaluate alignment sensitivity among all taxa, we evaluate sensitivity only among genomes a and q. The pairwise measurement allows us to inspect whether adding intermediate rungs on the phylogenetic ladder allows our algorithm to climb higher than otherwise possible. The results suggest that in general, Progressive Mauve can produce substantially better alignments when given additional sequence information for intermediate taxa.

6.4 Discussion

The simulation studies reveal several important features of current genome alignment algorithms. In the absence of genomic rearrangement, aligners such as MAVID, Multi-LAGAN, and Progressive Mauve offer comparable performance and are able to align extremely divergent genomes, up to .75 average substitutions per site in our study. When significant amounts of gene flux or rearrangement have taken place, the multiple genome alignments computed by Progressive Mauve offer an unprecedented level of accuracy. Progressive Mauve outperforms both Mauve and TBA for nucleotide-level alignment and outperforms Mauve for detection of LCBs indicative of orthology or xenology. Progressive Mauve's ability to accurately localize the breakpoints of genomic rearrangement should permit automated study of sequence patterns (such as repeats or mobile elements) associated with genomic rearrangement.

6.5 Acknowledgments

Portions of this chapter appeared as Darling, Mau, Blattner, and Perna (2004a).

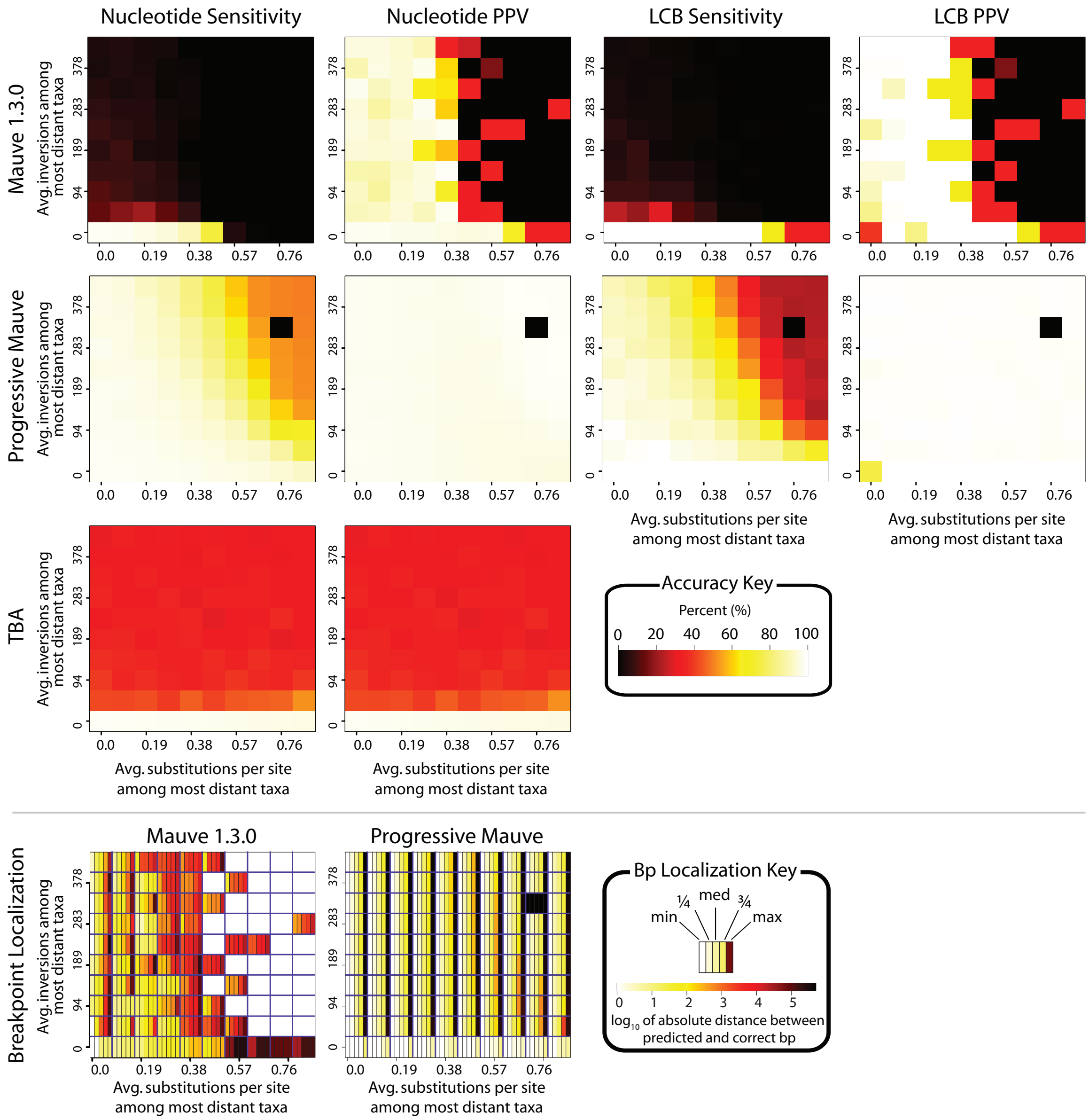

Figure 24: Accuracy of Mauve 1.3.0 (first row), Progressive Mauve (second row), and TBA (third row) when aligning genomes with increasing amounts of nucleotide substitution and inversions. The inversion rate increases along the y-axis and the substitution rate increases along the x-axis. Colors indicate a percentage scale ranging from 0% (black) to 100% (white). Progressive Mauve clearly outperforms Mauve 1.3.0 and TBA over the entire space of mutation rates. We do not report LCB accuracy for TBA because it does not identify monotoporthologous LCBs. The lower portion of the figure illustrates the ability of Mauve and Progressive Mauve to localize the breakpoints of rearrangement. For correctly predicted LCBs, the absolute distance between the predicted breakpoint and true breakpoint is recorded. Each cell is a composite of five values, showing the min, first quartile, median, third quartile, and maximum error in breakpoint localization. The entirely white cells in the bp localization results for Mauve 1.3.0 occur when Mauve 1.3.0 makes no LCB predictions at all, thus achieving perfect positive predictive value. The black cells in Progressive Mauve indicate runs which did not complete.

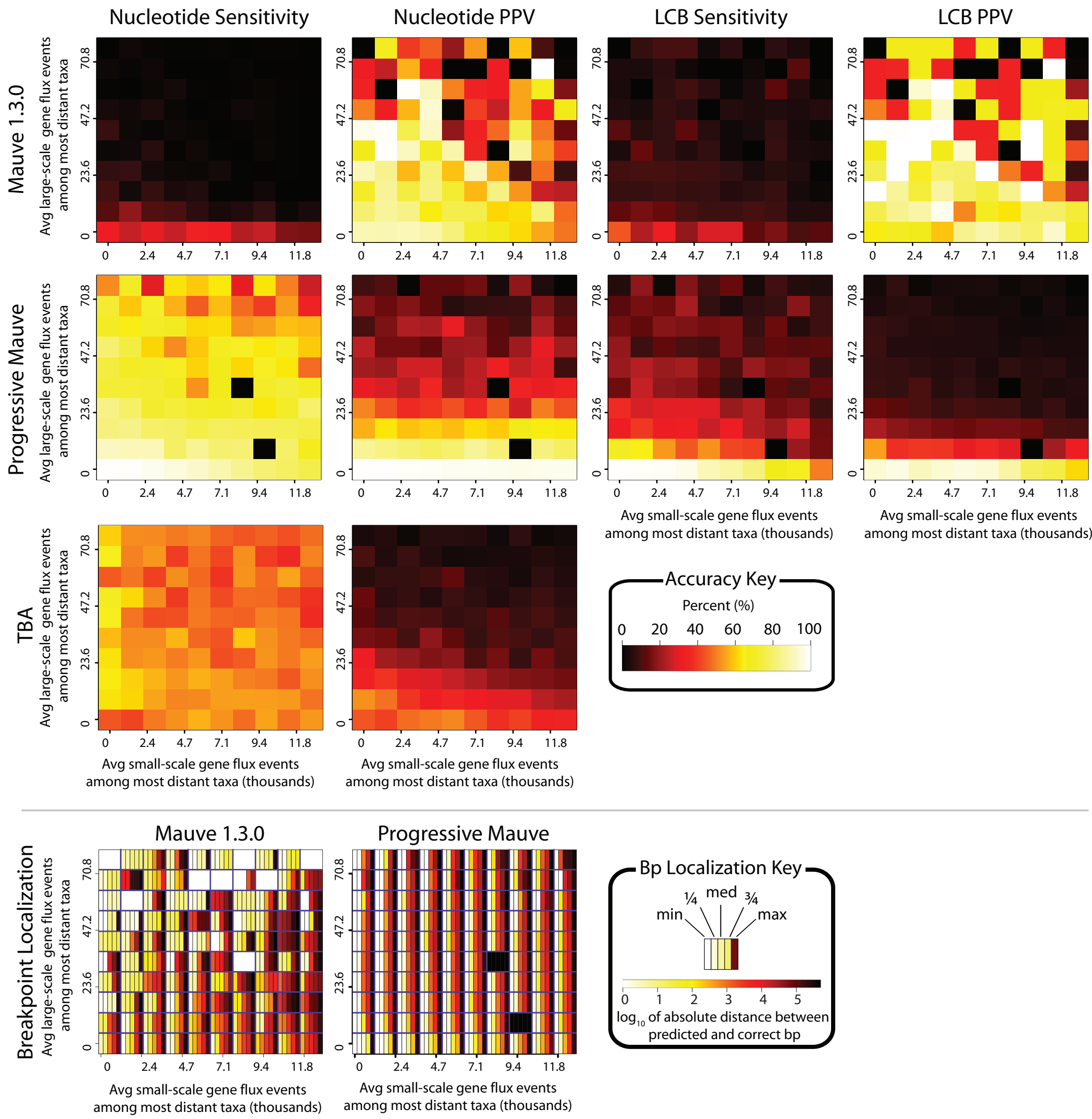

Figure 25: Accuracy of Mauve 1.3.0 (first row) and Progressive Mauve (second row) when aligning genomes with increasing amounts of small-scale and large-scale gene flux. The y-axis gives the average number of large gene flux events between the most distant taxa shown in Figure 20. The x-axis gives the average number of small gene flux events between the most distant taxa. Colors indicate a percentage scale ranging from 0% (black) to 100% (white). The substitution rate and indel rate were fixed at the combination indicated by the asterisk in Figure 21. The inversion rate was set to a value which results in 42 average inversions among the most distant taxa. Progressive Mauve clearly outperforms Mauve 1.3.0 and TBA over the entire space of mutation rates, although all methods tend to break down in the face of substantial large-scale gene flux. Again, we do not report LCB accuracy for TBA because it does not identify monotoporthologous LCBs.

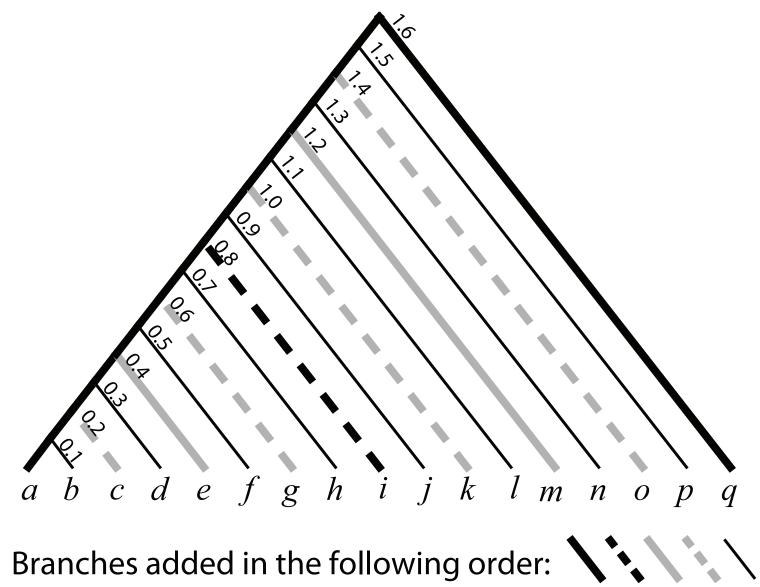

Figure 26: Phylogenetic ladder used for alignment accuracy profiling. The initial tree includes the two thick solid black branches connecting nodes a and q. We then add dashed black branches, solid grey branches, dashed grey branches, and finally thin black branches for the experiments in Figure 27 labeled B, C, D, and E, respectively. The sequence of branch additions corresponds to starting with two divergent genomes, and repeatedly sampling the genomes of phylogenetic intermediates. Thus, the first tree has two taxa, the second has three, third has five, fourth has nine, and fifth has seventeen.

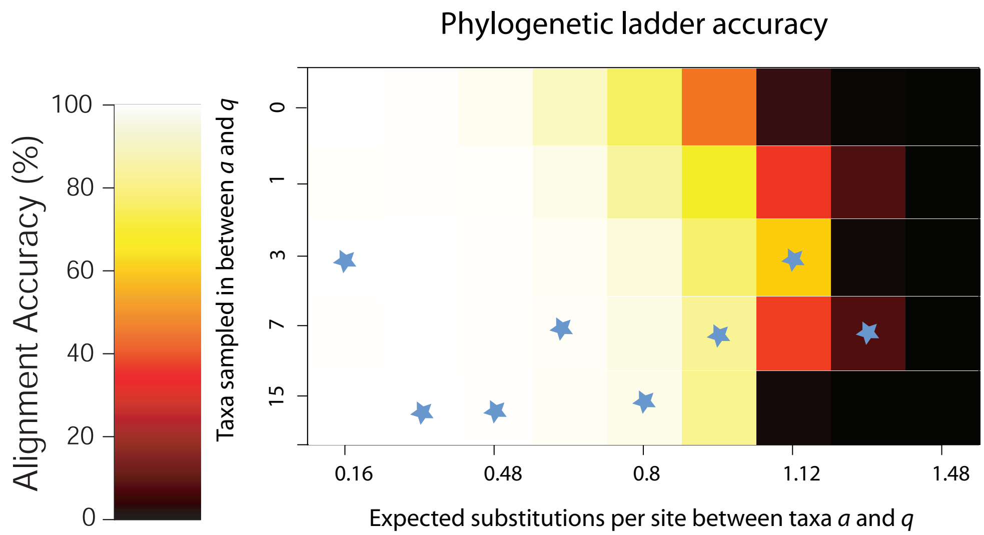

Figure 27: Accuracy of Progressive Mauve when aligning data simulated according to a phylogenetic ladder. As the number of taxa sampled increases the quality of the alignment generally increases, indicating that the aligner effectively exploits additional sequence information. Alignment quality deteriorates at high mutation rates even when a large number of taxa are sampled until a and q become unalignable at 1.48 average substitutions per site. Substitutions were sampled according the the HKY85 model with a Ts/Tv ratio of 4 and gamma-distributed rate heterogeneity with shape=1. Indels were sampled at a rate equal to 5% the rate of nucleotide substitution, and no rearrangement or gene flux was modeled. We performed five replicates of each experiment, and the average pairwise sensitivity of alignments among sequences a and q was measured. The data set with the highest average sensitivity at each mutation rate is labeled with a blue star.

Chapter 7

Detecting homologous recombination in genome alignments

7.1 Introduction

The role of lateral gene transfer (LGT) in shaping prokaryotic genomes has been the subject of intense investigation and debate in recent years (Milkman, 1997, Daubin et al., 2003, Feil et al., 1999, Spratt et al., 2001, Gogarten et al., 2002, Lawrence and Hendrickson, 2003, Lerat et al., 2003, Ochman et al., 2005, Ge et al., 2005, Beiko et al., 2005). In the pre-genomic era, the handful of examples of LGT were detected primarily as discordance between phylogenetic reconstructions with different housekeeping genes (Dykhuizen and Green, 1991, Bowler et al., 1994, Suerbaum et al., 1998, Reid et al., 2000). The explosion of publicly available bacterial genome sequences, coupled with the development of whole-genome comparison tools (Carver et al., 2005, Kurtz et al., 2004a, Darling et al., 2004a), initially focused LGT discovery on genome-wide scans for islands of sequences specific to particular lineages of bacteria (for example, (Perna et al., 2001, Parkhill et al., 2001, Tettelin et al., 2005, Hsiao et al., 2005)). Most recently, phylogenetic approaches are applied to detect LGT among genome-wide sets of putative orthologs (Daubin et al., 2003, Ge et al., 2005, Beiko et al., 2005). Together, these studies point to low, but detectable, levels of LGT among distantly related species with occasionally higher rates found among organisms that occupy similar environments. Closely related organisms show higher levels of LGT, with intraspecific comparisons showing the highest levels. Two limitations of these analyses are the lack of phylogenetic resolution, particularly among intraspecific comparisons, and the reliance on annotated boundaries of genes in delineating candidate regions.

Statistical and phylogenetic methods have been developed for detecting recombination in aligned sequences of single genes or relatively short genomic segments. One general approach, referred to as nucleotide substitution distribution methods in (Posada et al., 2002), assesses atypical clusters of nucleotide differences. Clusters come in two flavors: groups of polymorphisms exhibiting the same topologically discordant pattern (Graham et al., 2005, Stephens, 1985), or an elevated rate of mutation in a single lineage across a segment of the alignment (Maynard Smith, 1998, Qiu et al., 2004, Sawyer, 1989, Worobey, 2001). The former indicates recombination between compared strains, while the latter implies a recombination with some unknown, more divergent, strain. Phylogenetic methods are most often applied in the context of detecting recombination break points in sequence alignments (Grassly and Holmes, 1997, Husmeier and McGuire, 2002, McGuire and Wright, 2000, Minin et al., 2005). These methods require longer alignments, are computationally intensive, and have reportedly been outperformed by substitution distribution methods on simulated test data (Posada and Crandall, 2001).

Genome-scale analyses of lateral transfer events have typically relied on identification of incongruent tree topologies from phylogenetic analyses of sets of putative orthologous genes identified by reciprocal BLAST analyses (Lerat et al., 2003, Ge et al., 2005, Raymond et al., 2002). This approach can be confounded by errors associated with BLAST, such as false-positive orthologs, is limited to identifying recombination events that occur within gene boundaries, and is unlikely to identify short recombined regions within genes.

Recently, a Markov clustering algorithm was used to partition orthologous pairs of genes, determined by an all-versus-all BLAST comparison of 144 fully sequenced prokaryotic genomes, into maximally representative clusters (Beiko et al., 2005, Harlow et al., 2004). Bayesian phylogenetic analysis (for example, (Mau et al., 1999, Ronquist and Huelsenbeck, 2003)) was applied to each cluster of four or more taxa to infer lateral gene transfer against the background of a consensus 'supertree' of sequenced bacteria. This approach is most successful in determining global pathways of gene transfer between phyla and divisions of prokaryotes, where homologous recombination is unlikely to have played a significant role. Rather, these likely arise as illegitimate recombination events.

Here, we develop a method to detect segments of closely related genomes that have been replaced with a homologous copy from another conspecific lineage, that is, an allelic substitution. The method is not designed to detect non-homologous sequences that may have accompanied a homologous recombination event or homologous recombination events involving identical alleles.

The method compiles a list of polymorphism sites from a whole-genome multiple alignment, then applies score functions to locate clusters discordant with the predominant phylogenetic signal. Identified clusters can cross gene boundaries and non-coding sequence. Our use of extreme value theory furnishes us with a statistically defensible criterion to assess significance of these clusters in much the same manner as the

Karlin-Altschul statistics help interpret BLAST results (Altschul et al., 1990, Karlin and Altschul, 1990).

We apply the recombination detection method to the published genome sequences of several *E. coli* (Perna et al., 2001, Blattner et al., 1997, Jin et al., 2002, Wei et al., 2003, Hayashi et al., 2001, Welch et al., 2002). Construction of a multiple whole genome alignment facilitates a global survey of recombination among these *E. coli* isolates. Genome sequences must first be partitioned into locally collinear blocks (LCBs) - regions without rearrangement. Most LCBs contain lineage-specific sequence acquired through lateral gene transfer or differential gene loss. To further complicate matters, non-homologous sequences from different organisms can integrate into different lineages at a common locus (Perna et al., 2001). In a previous work, we developed a software package called Mauve (Darling et al., 2004a) that can construct global multiple genome alignments in the presence of rearrangement and lineage-specific content. The Mauve alignments provide a convenient starting point for locating polymorphic patterns indicative of intraspecific recombination, which we call allelic substitution.

7.2 Results

As seen in Figure 28, the Mauve genome aligner takes the four *E. coli* and two *Shigella flexneri* genome sequences and returns 34 local alignments spanning 3.4 Mb of homologous sequence common to all strains. The majority of rearrangements occur in *Shigella* genomes where inversions between copies of repetitive elements are relatively frequent (Blattner et al., 1997).

Computer-assisted screening of the Mauve output finds 733 problematic intervals

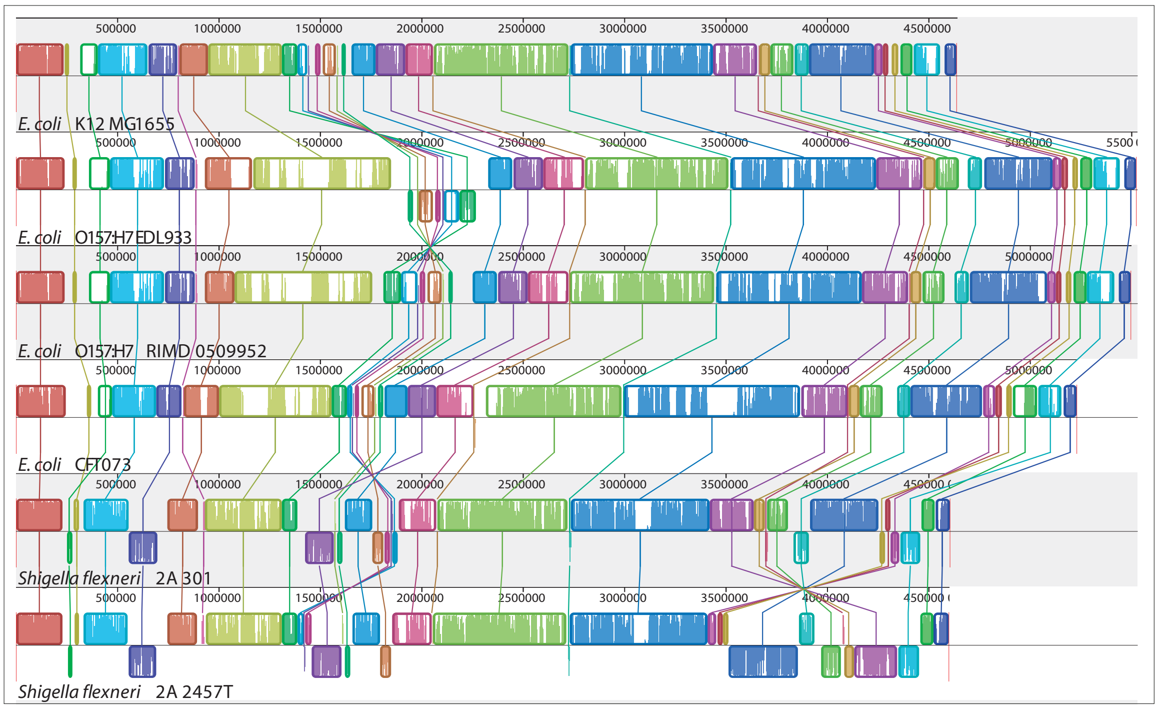

Figure 28: A multiple whole-genome alignment of six strains consists of 34 rearranged pieces larger than 1 kb. Each genome is laid out horizontally with homologous segments (LCBs) outlined as colored rectangles. Regions inverted relative to *E. coli* K-12 are set below those that match in the forward orientation. Lines collate aligned segments between genomes. Average sequence similarities within an LCB, measured in sliding windows, are proportional to the heights of interior colored bars. Large sections of white within blocks and gaps between blocks indicate lineage-specific sequence.

Bipartition (split)	Pattern KOOCS	Number of SNDs	Relative frequency
((KSSOO)C)	111211	50,354	38.73
((KSSC)OO)	122111	19,678	15.14
((KOOC)SS)	111122	18,490	14.22
((KSSOO)C)	111211	14,115	10.86
((KSS)(OOC)) = KS	122211	9,882	7.60
((KOO)(SSC)) = KO	111222	6,890	5.30
((KC)(OOSS) = KC	122122	5,874	4.52

Table 5: Common single nucleotide differences have two alleles. Each such nucleotide difference separates the six genomes into two classes. Pattern codes are represented as 6-tuples of ones and twos (for allele 1 and allele 2) in the following order: (K) *E. coli* K-12 MG1655, (O) *E. coli* O157:H7 EDL933, (O) *E. coli* O157:H7 Sakai strain RIMD0509952, (C) *E. coli* CFT073, (S) *Shigella flexneri* 2A 301, and (S) *Shigella flexneri* 2A 2457T. By convention, K-12 is always allele one. For brevity, key groupings are denoted as KS, KO, or KC. The remaining 3.6% SNDs come in over 50 different patterns, including one quadripartition. See Appendix 1 in additional data file 1 of Mau et al. (2006) for additional frequencies.

inside LCBs in which base pairs do not properly align because of gaps created by lineage-specific sequence and/or attempts to align non-homologous sequence. Deleting these intervals from the alignment yields 130,008 high quality base pair differences. Common bipartitions, constituting 96.4% of all such differences, are listed in Table 5.

We use the term 'single nucleotide difference' (SND) to describe the partition structure at a variable site in the alignment. A representative 100 base-pair (bp) segment of the 3.4 Mb alignment is presented in Figure 29 for illustrative purposes.

All but 2% of variable sites are bi-allelic, meaning each site splits six strains into two groups, called a bipartition. Nearly 80% of the bi-allelic SNDs have a minor allele unique to the CFT, K-12, O157:H7, or *S. flexneri* lineage. The remaining bi-allelic SNDs divide the lineages into three alternative pairings of sister taxa, giving rise to three alternative unrooted tree topologies denoted as: ψ_{KS} (K-12 with *S. flexneri*, CFT with O157:H7); ψ_{KO} (K-12 with O157:H7, CFT with *S. flexneri*); and ψ_{KC} (K-12 with CFT, O157:H7

```
                         START CDS mutS
AATATCAGGGAACCGGACATAACCCCATGAGTGCAATAGAAAATTTCGACGCCCATACGCCCATGATGCAGCAGTATCTCAGGCTGAAAGCCCAGCATCC K-12 MG1655
AATATCAGGGAACCGGACATAACCCCATGAGTGCAATAGAAAATTTCGACGCCCATACGCCCATGATGCAGCAGTATCTCAAGCTGAAAGCCCAGCATCC O157:H7 EDL933
AATATCAGGGAACCGGACATAACCCCATGAGTGCAATAGAAAATTTCGACGCCCATACGCCCATGATGCAGCAGTATCTCAAGCTGAAAGCCCAGCATCC O157:H7 Sakai
AACATCAGGGAGCCGGACTTAACCCCATGAGTACAATAGAAAATTTCGACGCCCATACGCCCATGATGCAGCAGTATCTCAAGCTGAAAGCCCAGCATCC CFT073
AATATCAGGGAACCGGACATAACCCCATGAGTGCAATAGAAAATTTCGACGCCCATACGCCCATGATGCAGCAGTATCTCAAGCTGAAAGCCCAGCATCC S.flexneri 2A 301
AATATCAGGGAACCGGACATAACCCCATGAGTGCAATAGAAAATTTCGACGCCCATACGCCCATGATGCAGCAGTATCTCAAGCTGAAAGCCCAGCATCC S.flexneri 2A 2457T
2855097^  2855107^  2855117^  2855127^  2855137^  2855147^  2855157^  2855167^  2855177^              Coordinates in K-12
  1        1      1             1                                                1
```

Figure 29: Small sample segment of the alignment spanning the start of the *mutS* gene (denoted in blue). Location of a mismatch is indicated by the integer '1' along the bottom row. Five columns contain SNDs: TTTCTT, AAAGAA, AAATAA, GGGAGG, and GAAAAA. The first four share the same bipartition pattern (111211) and are deemed equivalent, even though one of them results from a transversion. The fifth SND is considered distinct based on its bipartition despite having the same mutation (A to G) found in the second SND.

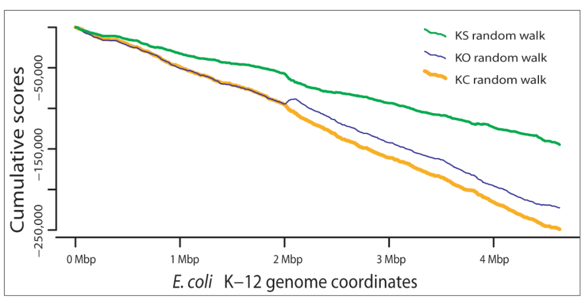

Figure 30: Three excursions (KS, KO, and KC) spanning the alignment with K-12 MG1655 as reference genome. The KS random walk plot, representing the dominant clonal topology, decreases more gradually than do the two other plots. Excursions for the discordant topologies (patterns KO and KC) run parallel to one another, except in a 100 kb region at 2 Mb where KO abruptly increases. Parallel flat gaps common to all three plots reflect K-12 lineage-specific sequence.

with *S. flexneri*).

The four lineages serve as operational taxonomic units (OTUs) in our study of allelic substitution in *E. coli*. When nucleotides at a polymorphic site exhibit a partition structure explainable by a single point mutation, the induced bipartition is said to be compatible with the enabling topology. Bipartitions labeled KS, KO, and KC in Table 5 are compatible with the topologies ψ_{KS}, ψ_{KO}, and ψ_{KC}, respectively. Note that frequency of the KS pattern exceeds that of each of its competitors by 3,000 SNDs, thus certifying ψ_{KS} as the 'species' topology. The elevated frequency of SNDs unique to CFT roots topology ψ_{KS} as (((KS)O)C). The 102,000 topologically uninformative lineage-specific SNDs nevertheless provide information that our method uses to assess recombination.

We define three complementary score functions that discriminate between KS, KO, and KC patterns. Each of these score functions assigns an integer value to each SND pattern. Moving across the chromosome of reference strain MG1655, we keep a cumulative sum of the scores assigned by each function to consecutive SNDs in the alignment. Graphical representations of cumulative scores, called random walk plots or excursions, can reveal large-scale variations in feature composition. Excursions for each of the three topologies are plotted concurrently in Figure 30.

A large phylogenetic anomaly appears midway through the alignment. Magnification of a 100 kb segment between 1.95 and 2.1 Mb reveals a core 40 kb region in which KO SNDs are the dominant pattern of substitution, flanked by transitional regions for which ψ_{KO} serves as the 'gene tree' as well.

Global random walk plots highlight grossly deviant regions. In this alignment, a solitary segment stands out. All other regions appear indistinguishable from one another in Figure 30. Unless stated to the contrary, DNA sequence and genes from the large atypical region (from *sdiA* to *gnd*) are excluded from further computations (a separate analysis of this region is included in Appendix 2 of additional data file 1 of Mau et al. (2006)).

7.2.1 Local variation in phylogenetic signal

In Figure 30, clusters of like patterns labeled KS, KC, or KO generate tiny, imperceptible bumps in the corresponding random walk plots. Examined at higher resolution (data not shown), they can be seen to punctuate each excursion. However, manual scanning of high-resolution random walk plots is tedious, time consuming, and error-prone. In Materials and methods, we describe an alternative strategy that automatically scans for clusters at the local level.

The score functions generating Figure 30 are designed to elicit large positive local scores (differences in cumulative scores evaluated at nearby positions) whenever clusters of like, topologically informative, patterns are encountered. When that local score exceeds a predetermined threshold, the interval between the delimiting SNDs is declared a high scoring segment (HSS). The strategy behind this scheme is exactly analogous to BLAST (Altschul et al., 1990), in which high scoring segments denote probable homology between the query and one or more reference sequences.

When two lineages share a nucleotide that is not the result of a single mutation in a common ancestor, a homoplasy is said to have occurred. Homoplasies arise either through multiple mutations at a common site (convergent evolution) or recombination. The former tend to be distributed randomly about an alignment, whereas a recombination event typically produces a cluster of nucleotide differences at nearby sites exhibiting the same SND pattern. Our approach identifies such clusters of nucleotide differences with a common phylogenetic partitioning pattern. Variability in mutation rates and patterns in different chromosomal regions and bacterial lineages might also lead to physical clustering of similar substitutions. Although the clustering of sites with similar patterns strongly suggests homologous recombination between lineages, we cannot rule out the possibility that some clusters arise by independent mutation-driven processes. Simple score functions alone cannot distinguish between these two possibilities, though the latter is believed to be relatively rare.

Our method relies on the relative intensity of particular SND patterns (the one of interest versus all others) to measure cluster formation, rather than the absolute number of SNDs in any given fixed length segment of the alignment. As a result, local mutational intensity is factored out of the analysis. We assert this is legitimate provided the overall rate of mutation is not too great, and local deviations from that average are not severe. A more detailed study is presented in Appendix 5 of additional data file 1 in Mau et al. (2006). Random SNDs can and do form clusters of identical patterns simply by chance. Given the number of SNDs and their relative frequencies within the alignment, we wish to distinguish 'bumps' that are too large to have occurred by chance.

Here again, BLAST statistics (Karlin and Altschul, 1990) serve as the model for assessing significance. Random walk theory provides the tools for assessing high scoring segments, and the corresponding extreme value distributions (EVDs) guide selection of appropriate thresholds. Random walks (as opposed to random walk plots) are stochastic processes operating under a fixed set of probabilities at each stage.

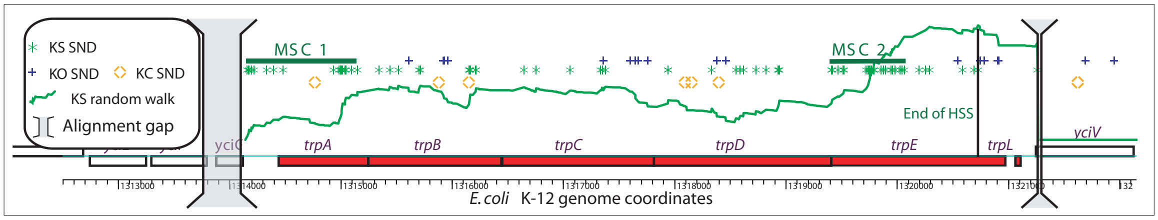

Figure 31: The KS local random walk plot showing homologous recombination in the tryptophan (*trp*) operon. Genes are rectangular boxes positioned above or below the axis based on transcribed strand. KS SNDs form two non-overlapping MSCs with significant local scores exceeding 170. Both MSCs, with a combined length under 2 kb, are contained in a single 6.5 kb HSS covering most the *trp* operon. The positions of each KO, KC, and KS SND in *E. coli* K-12 are shown above the KS excursion. Random walk values below 50 are not plotted, resulting in the absence of visible KC or KO excursions.

In the Materials and methods section, we apply the relevant theory to derive thresholds. Using the appropriate extreme value distribution as an arbiter, we chose a significance threshold of 170 for clusters of KS SNDs and the same value of 100 for both KO and KC, as their frequencies are nearly identical outside the large atypical region (4.85% versus 4.57%). These thresholds define 186 high scoring segments that span 7.5% of the sequence alignment. A breakdown by pattern and range of scores is arrayed in Tables 2 and 3.

We deviate from BLAST protocols in one important respect: a high scoring segment maximizes the local score, which is the primary goal of sequence alignment. Here, we want to isolate sub-regions within an HSS that individually exceed the significance threshold. Our rationale is that sequence between sub-regions may not have participated in the recombination, and we want to identify only those genomic intervals that possess *prima facie* evidence of recombination.

A minimal significant cluster (MSC) is a smallest subset of contiguous SNDs generating a local score above the threshold. To avoid ambiguity, overlapping MSCs supporting the same topology are merged into a single representative MSC. Most high scoring segments consist of a single such cluster, but HSSs with more than 150 SNDs often contain two or more disjoint MSCs.

HSSs and MSCs are represented graphically by modifying global random walk plots. By subtracting off the underlying negative trend, only positive local scores are displayed. Figure 31 shows a local random walk plot for the HSS covering the seven genes of the tryptophan operon. The *trp* operon was the first reported example of homologous recombination in *E. coli* (Stoltzfus et al., 1988).

Although the entire *trp* operon may have been exchanged in a single event, only *trpA* and *trpE* contain clusters of KS SNDs that individually give rise to statistically significant local scores. Moreover, the first MSC clearly includes in excess of 200 bp downstream of the *trp* operon - evidence that downstream transcription termination signals have also been subject to homologous recombination. In this manner, MSCs facilitate more precise targeting of chromosomal regions implicated in recombination. This criterion modestly increases the number of recombined segments to 216 (75, 62, 79 for KO, KC, KS, respectively) while reducing the amount of participating sequence from 251 kb to 129 kb. We outline a procedure for finding non-overlapping minimal significant clusters inside high scoring segments in Materials and methods.

HR detected	Genes	Percent Recombined	χ^2 score	Multi-Fun Level 2 categories
5	144	3.5	4.52	Ribosome and peptidoglycan structure
10	237	4.2	5.47	Cell division, cell protection, and adaptation to stress
14	279	5.0	4.35	Protein-related information
20	329	6.1	2.94	RNA-related information
386	4,035	9.6	Not Reported	All other functions, including unknown
48	357	13.5	9.24	Building block biosynthesis
16	109	13.8	3.21	DNA-related information
7	40	17.5	3.56	Group translocators (PTS)
9	46	19.6	6.24	Motility

Table 6: Categories with few members such as ribosome and peptidoglycan structure are combined together, as are three types of cell processes. We computed a χ^2 goodness-of-fit statistic for each category, but do not report p values because dependencies exist between categories.

7.2.2 Gene content of regions that underwent recent allelic substitution

Although our method identifies recombination events independently of gene boundaries, it is interesting to look at the types of genes and gene products involved in these events. To this end, we extracted a list of genes encoded in regions deemed atypical by our random walks. Among the 4,353 genes in K-12, 3,107 align across all six genomes. Of these, 271 genes intersect a minimal cluster segment. When augmented with 40 genes from the atypical region, 10% of shared genes exhibit evidence of recombination. A table of the 186 high scoring segments, subdivided into MSCs and identifying affected genes, is provided as Additional data file 2.

We examined this list of 311 genes in light of gene function assignments made using a controlled vocabulary called MultiFun (Serres and Riley, 2000) that supports multiple functional classifications for a given gene. The 3,107 genes aligned by Mauve in all six genomes have been classified with 5,550 gene functions. Nearly 2,000 genes have a single classification (many are 'Unknown function'). By contrast, six genes have seven 'Level 2' functions. This analysis revealed an over-representation of four categories and under-representation in seven others (Table 6).

Highly conserved genes that encode components of the ribosome and genes involved in peptidoglycan biosynthesis show little evidence of detectable recombination. Conversely, many genes involved in motility and chemotaxis undergo allelic substitution. Chemotaxis may also be related to elevated recombination detected among genes encoding components of phosphotransferase transport systems (PTSs) since these genes can double as sensors for substrates such as glucose and mannose (Zeppenfeld et al., 2000).

Genes involved in basic processing of cellular information, such as replication, transcription and translation, reveal an unexpected dichotomy: genes dedicated to RNA and protein metabolism are refractory to recombination, but genes involved with DNA replication, repair and recombination appear prone to allelic substitution. Equally surprising is a bias favoring evident recombination among genes involved in small molecule biosynthesis. Examples of biosynthetic genes that support the pairings in topology ψ_{KC} include members of the aromatic amino acid pathway (*aroP*, *aroD*, and *aroG*) as well as the pyrimidine producing *carB* (also known as *pyrA*). SND clusters supporting topology ψ_{KO} are present in *pyrI*, *pyrB*, and several genes in the histidine operon. Finally, *purD*, *purF*, *leuDC*, *modABC*, and two genes in the *trp* operon (Figure 31) contain clusters compatible with the clonal topology, but at much higher intensity than elsewhere in the genome.

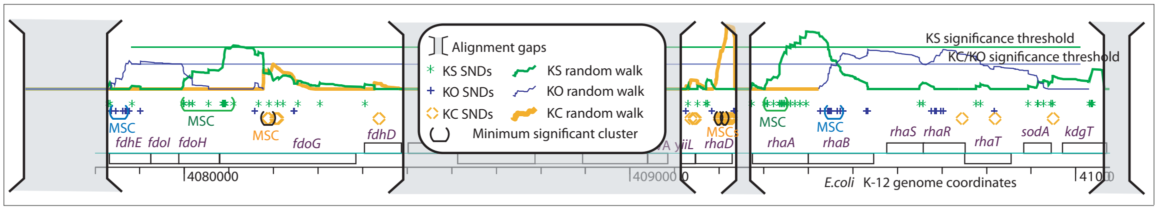

Figure 32: Mosaic operons and genes. Three of six *rha* genes (*rhaB*, *rhaA*, and *rhaD*) belong to an operon on the reverse strand. This operon is unusual because well- defined recombination events clearly fall within gene boundaries; *rhaD* contains two dense KC clusters, whereas *rhaA* and *rhaB* contain predominantly KS and KO SNDs, respectively. In a nearby operon consisting of *fdoG*, *fdoH*, *fdoI*, and *fdhE*, there has been a KC intragenic recombination event with *fdoG* a mosaic, resulting from two recombination events, one of which is shared with *fdoH*.

7.2.3 Mosaic operons and genes

With over 216 recombined segments intersecting 271 genes, this group of *E. coli* genomes is truly a patchwork of its constituent members. Although genes within the *trp* and his operons contain multiple clusters of the same pattern (KS for *trp*, KO for *his*), such uniformity across operons is atypical (Omelchenko et al., 2003). Figure 32 shows a short stretch of aligned sequence containing two mosaic operons.

Besides *fdoG* (shown in Figure 32), six other genes - *polB*, *mutS*, *speF*, *recG*, *actP*, and *yfaL* - show evidence of mosaicism. Three of these genes–*polB*, *mutS*, and *recG*–are informational genes involved in DNA replication and repair. Each mosaic gene contains two minimum significant clusters generated by different partition patterns. A closer inspection of one of these genes, *speF*, suggests that all three phylogenetic signals may be present, as shown in Figure 33.

Other mosaic genes undoubtedly exist within these strains, but their phylogenetic signal is too short or too weak to register in a genome-wide scan. Full genome scans come at a cost; one must sacrifice sensitivity to maintain specificity. At present, we are

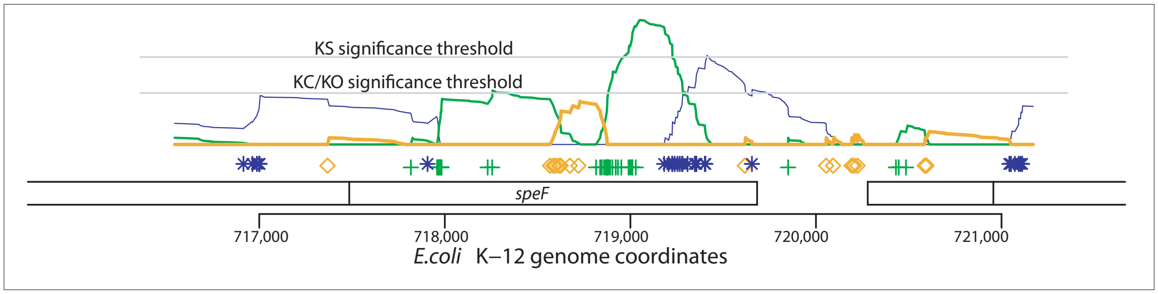

Figure 33: Random walk plots for positive local scores in the vicinity of the *speF* gene. *speF* is a mosaic gene by virtue of its KS and KO clusters. Note the small cluster of KC SNDs appears to divide a large KS segment near coordinate 718,600. This short KC spike, though not statistically significant on a whole genome scale, would undoubtedly pass a single gene substitution distribution type test.

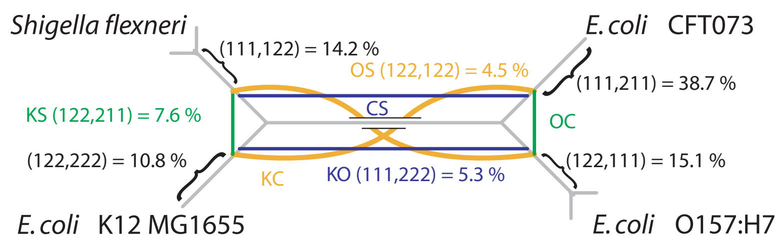

Figure 34: Percentage of SNDs supporting each of three topologies in a phylogenetic network for six *E. coli* genomes (four OTUs). Black lines describe the 'species' topology. Green, blue, and orange lines indicate the alternative pairings of sister taxa that result from KS, KO, and KC recombinations respectively. Also shown is the percentage of SNDs supporting each bipartition in Table 5.

content to underestimate the true amount of recombination in order to eliminate false positives.

7.3 Discussion

Natural transformation, transduction, and conjugation are three mechanisms for transporting foreign DNA into the cell. The relative contribution of each mechanism varies from species to species. For example, transformation is the dominant mode of transfer in bacteria such as *Neisseria meningitidis* and *Helicobacter pylori* that are naturally competent, that is, able to absorb small pieces of naked DNA. As *E. coli* is competent only under extreme conditions, typically in the laboratory, it is expected that this form of transformation may play a minor role in nature. Exogenous DNA can also enter via phage transduction or conjugation, which are expected to be the primary source of exogenous DNA for *E. coli*. Transducing phages can deliver large fragments of genomic DNA from their previous bacterial host into a recipient strain. DNA transferred via conjugative mechanisms can be even larger.

The lengths of recombined segments reported in the previous section are typically short. Half the intervals are shorter than 1 kb, and 80% are less than 2 kb. DNA fragments delivered by transducing phages might be expected to be considerably larger (30 to 60 kb). The size differential between entrance and incorporation molecules has been partially reconciled by experiments in which site-specific DNA was packaged into phages and transduced into K-12 cells (McKane and Milkman, 1995). Screening for recombinants in the proximity of the *trp* operon, the authors found average replacement sizes to be in the 8 to 14 kb range. Moreover, multiple replacements were detected in some instances. In a follow-up paper (Milkman, 1997), the level of sequence dissimilarity (from 1% to 3%) between recipient and donor strains was shown to correlate with the degree of abridgement by restriction endonucleases. The length of a typical recombinant in our study is still an order of magnitude less than that reported by McKane and Milkman (McKane and Milkman, 1995), but they based their conclusions on restriction site analysis, which has a limited ability to detect short fragments. Actual incorporations in their experiments could conceivably have been more frequent and shorter. Overlapping recombination events at particular sites are also likely to contribute to the net reductions in observed incorporation sizes.

Our approach detects significant clusters of phylogenetically informative SNDs, but does not tell us which lineages participated in the recombination. When presented with four OTUs, recombination is possible between six undirected donor-recipient pairs: KO, CS, KS, OC, KC, and OS. These alternative histories can be jointly represented as a phylogenetic network (Figure 34).

For example, a high scoring KC segment indicates that the donor and recipient lineages are either K-12 and CFT, or O157:H7 and *S. flexneri*. Exactly which pair of lineages is involved in the transfer can sometimes be determined by examining the joint distribution of all seven SND patterns. Recombinant activity in *glyS* and the four genes to its right is illustrated in Figure 35.

The colored intervals in Figure 34 share a common feature: the presence of topologically informative SNDs is accompanied by the absence of SNDs from two paired sister taxa. For example, no 'O157 only' or '*Shigella* only' SNDs are present in the KC/OS interval inside *glyS*, strongly suggesting that the O157:H7 and *S. flexneri* lineages were involved in the transfer. The other two intervals coincide with gene boundaries. When viewed in isolation, the genes *yiaA* and *yiaH* appear to be reasonable candidates for recombination. Yet only the KC recombinant inside the *glyS* gene is detectable by our whole genome significance thresholds.

Sequence divergence can reduce the likelihood that homologous recombination occurs between orthologous genes, but does not address the underlying mechanisms that lead to divergence in the presence of rampant recombination. The restriction of different lineages of bacteria to distinct niches could act to prevent gene flow, but in the case of *E. coli* and *Salmonella*, the niches overlap. The barriers to exchange might also reflect more active exclusion of foreign DNA by mechanisms such as restriction enzyme

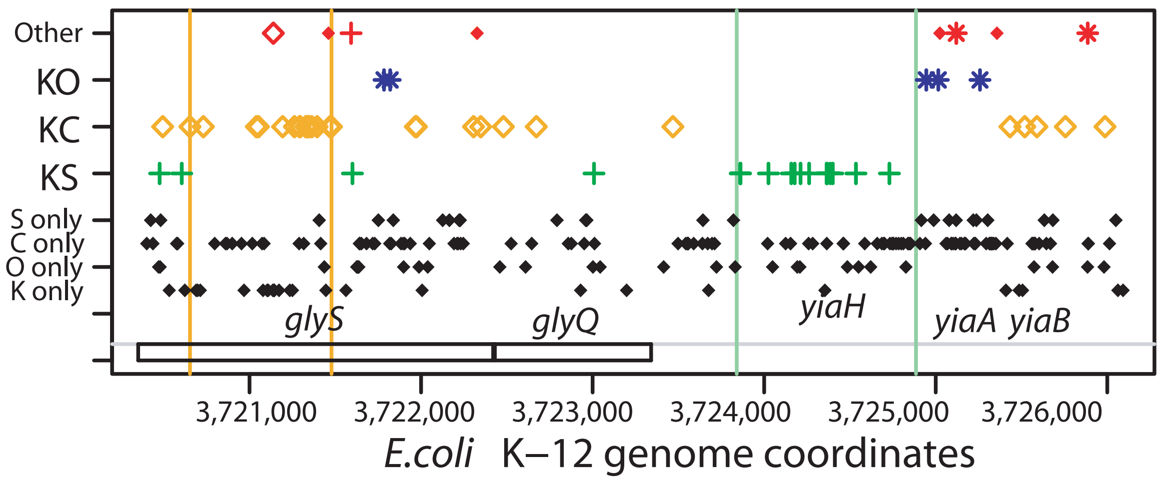

Figure 35: The location of all SNDs in a 5 kb region. In clusters demarcated by colored lines, note the corresponding absence of two more common types of SNDs. Three diamonds in lighter shades of blue, green, and red are compatible tri-partitions. Colored lines demarcate regions where the absence of lineage-specific SNDs is offset by an increase in the corresponding recombinant pattern (for example, in *yiaA*, no K-12 or *S. flexneri* only SNDs).

expression. Perhaps the most appealing explanation for the phenomenon would invoke the activity of bacteriophages, transposons and conjugation-promoting elements as the key determinants of recombinational potential between taxa. Given the propensity of these mobile elements to participate in genetic exchange within species and their often narrow host ranges, we might expect that they promote recombination within a species but cannot transfer to more diverse organisms. The lack of extensive recombination of orthologous sequences between species may result from a competition between bacteria and phage that can activate rapid evolution of barriers to phage infection. Our estimate for a higher rate of homologous recombination among *E. coli* underscores the discrepancy between rates of intraspecies recombination, which appear to be quite common, and rates of recombination of orthologous genes between species such as *E. coli* and *Salmonella*, which appear to be much less frequent (Daubin et al., 2003).

Earlier comparisons of different *E. coli* strains (Milkman, 1997, Dykhuizen and Green, 1991, Reid et al., 2000, Guttman and Dykhuizen, 1994) found recombination among several distinct sets of genes. The affected genes in these studies were not randomly selected and may not have been representative of the shared gene complement. Although our method surveys all genes, the genomes we compared are heavily skewed towards human pathogens. As additional *E. coli* strains are sequenced, the role of homologous recombination in bacterial genome evolution will become clearer, and may force reassessment of traditional methods for describing relationships among bacterial taxa (Ochman et al., 2005, Feil and Spratt, 2001).

Our analytical methods are straightforward here because the number of unrooted topologies is the same as the number of topologically informative bipartitions. This correspondence decays exponentially as more operational taxonomic units are added. Sometimes going from four OTUs to five requires a new analytic procedure (for example, see (Zhaxybayeva et al., 2004)). We leave the challenging problem of extension to more taxa for future work.

7.4 Methods

The Mauve alignment tool produces an output file containing separate alignments for each locally collinear block. Concatenation of LCBs results in a $G \times M$ matrix of nucleotides and gap symbols, where G is the number of genomes and M is the length of gapped alignments across all blocks. Each matrix column represents one site in the consolidated alignment. Restricting attention to columns containing at least one nucleotide difference but no gaps results in a $G \times M'$ sub-matrix Δ composed solely of single nucleotide differences. Automated screening of the Mauve alignment (Figure 28) filtered out SNDs in regions of poor alignment quality, resulting in a Δ with dimension 6 by 130,008.

Numerous scoring schemes have been devised to identify and assess the statistical significance of molecular sequence features on a genomic scale (Karlin and Brendel, 1992, Karlin et al., 1991). One general approach calculates average scores within a sliding window (for example, (Lobry, 1996, Scherer et al., 1994)). We use an equally versatile method that computes cumulative scores based on a score function, evaluated at each column of δ (see (Karlin and Altschul, 1990) for other applications).

Let Ξ = KS, KC, KO represent the three discordant SND patterns in Table 5, and let ψ_ξ be the unrooted topology compatible with pattern $\xi \in \Xi$. We define three complementary score functions on SNDs to filter conflicting phylogenetic signals:

$$Score_\xi(s) = \begin{cases} +D, & \text{if } \phi(s) = \xi \\ -D, & \text{if } \phi(s) \in \Xi \setminus \{\xi\} \\ -1, & \text{if } \phi(s) \cap \Xi = \emptyset \end{cases}$$

where s is a SND and $\phi(s)$ is the corresponding partition pattern in Table 5, and $D = 13$. For a given $\xi \in \Xi$, the cumulative score at the n^{th} column in Δ is the partial sum:

$$\begin{aligned} S_n^\xi &= \sum_{i=1}^{n} Score_\xi(s_i) \\ &= S_{n-1}^\xi + Score_\xi(s_n) \\ S_0^\xi &= 0 \end{aligned}$$

These score functions share a key characteristic of alignment scoring schemes; both generate high scoring segments that identify regions of interest. In the case of alignments, a high score segment represents a likely sequence homology. A significant difference between our analysis and sequence alignment is that substitution matrices are empirically derived from a test set (for example, PAM or BLOSUM). Here, D is not a parameter in an underlying stochastic model of evolution, but rather a tuning parameter in a diagnostic specifically designed to detect recombination. The value $D = 13$ was inspired by the observation that the most frequent topologically informative pattern, KS, has an observed frequency of 7.6%, approximately the reciprocal of 13. Alternative integer values were tried and rejected.

Score functions generate high scoring segments whenever they encounter a cluster of SND patterns supporting one topology but are discordant with other choices. For a given topology ψ_ξ, we define $Score_\xi(\eta)$ to take on positive values when pattern η is ξ and negative values otherwise ($\eta \neq \xi$). As discordant patterns are antithetical to one another, their weights should be equal to but opposite from the one being scanned. Neutral SND patterns are not individually disruptive to the underlying signal, but in aggregate they degrade the signal. These non-informative patterns are down-weighted and made integer-valued as in substitution matrices.

Hence, a large local score–the equivalent of a high scoring segment–is evidence for recombination between two of the lineages paired by ξ (for example, $\xi =$ KS associates K-12 with *S. flexneri* and O157:H7 with CFT).

Random walk plots connect the dots between partial sums that are computed from SNDs as they occur in Δ. By contrast, random walks are translation invariant stochastic processes governed by the relative frequencies in Δ, irrespective of order. We augment the random walk transition probabilities with an additional 'terminator' state. Terminators break a global alignment into several smaller sub-alignments, and are used to represent alignment fragmentation caused by 'large' gaps (> 15 bp in one lineage), spurious alignments, or LCB boundaries (Figure 28). Accordingly, for each $\xi \in \Xi$, random walk increments are distributed according to the following probabilities:

$$X^{\xi}(S) = \begin{cases} +D & \text{with } \Pr(\phi(s) = \xi) = \pi_{\xi} \\ -D & \text{with } \Pr(\phi(s) \neq \xi) = \pi_{-\xi} \\ -1 & \text{with } \Pr(\phi(s) = \xi) = \pi_{other} \\ -100,000 & \text{with } \Pr(s \text{ is a break in the alignment}) = \pi_{break} \end{cases}$$

where $D = 13$, $\pi_{KO} = 0.048$, $\pi_{KS} = 0.076$, $\pi_{OS} = 0.045$, $\pi_{other} = 0.826$, $\pi_{break} = 0.005$ and $\pi_{-\xi}$ defined as:

$$\pi_{-\xi} = \sum_{\eta \in \Xi \setminus \{\xi\}} 1 - \pi_{other} - \pi_{break} - \pi_{\xi}$$

Since the expected value $E(X^{\xi}) < 0, \forall \xi$, sums of these identically distributed variables generate transient random walks. Random stopping times, defined recursively by:

$$\begin{aligned} \tau_0 &= 0 \\ \tau_1 &= \min\{i : S_i < S_0\} \\ \tau_{k+1} &= \min\{i : S_i < S_{\tau_k}\} \text{ for } S_k = \sum_{i=1}^{k} X_i^{\xi} \end{aligned}$$

form a strictly decreasing set of ladder points. Though S_k depends on ξ, we suppress it for ease of exposition. The horizontal distances between consecutive ladder points

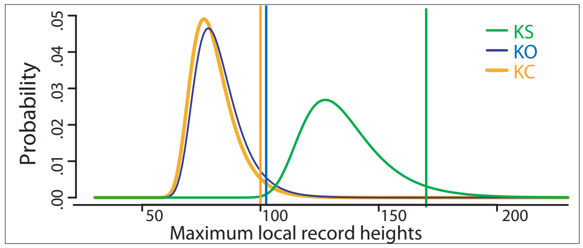

Figure 36: Statistical justification of threshold values −100, 100, and 170 for topologies KO, KC, and KS, respectively–used to identify recombination events. Values on the x-axis are maximal local scores. EVD probability densities for the maximum maximal local score attained by random walks of length M' appear as bell-shaped curves with a pronounced skew to the right. Threshold values, demarcated by vertical lines, correspond to conservative significance levels ($\alpha = 0.05$) for these distributions.

$\tau_{k+1} - \tau_k$, are called ladder epochs. The local record height (LRH) of the k^{th} epoch is defined by:

$$LRH_k = \max_{\tau_{k-1} \leq t < \tau_k} \{S_t - S_{\tau_{k-1}}\} \geq 0$$

Ladder epochs measure the size of a high scoring segment in SND units rather than base pairs (chain length M' versus M). The number of ladder epochs in a random walk of size N is denoted by $\Lambda(N)$. The distribution of the maximum value in a sequence of local record heights is an extreme value distribution (EVD) with parameterization:

$$\Pr(\max_{j \leq \Lambda(N)} LRH_j > x) = \exp(-NKe^{-\mu k})$$

Here μ is the positive solution of an equation involving the moment generating function:

$$mgf_\xi(\dots) = \sum_j \pi_j e^{\mu X^{\xi(s_j)}} = 1$$

The value of μ is solved for numerically. For ψ_{KC}, the equation:

$$mgf_{KC}(\mu) = 0.045e^{13\mu} + .124e^{-13\mu} + .826e^{-\mu} + .005e^{-100{,}000\mu} = 1$$

has a positive solution at $\mu = 0.1354$ ($\mu = 0$ is a trivial solution). The value of K can be computed as a rapidly converging infinite sum (see Appendix of (Karlin and Altschul, 1990)). We chose instead to simulate $2,000$ random walks of size $N = 10,000$ using the statistical package R (http://r-project.org). The largest local record height attained over the course of each simulation is saved. The functional form of the EVD (equation 1) is then fit to a probability histogram of $2,000$ stored maxima. The estimated values of K and Λ are combined with an $N = M'$ to adjust for the actual alignment size ($M' = 129,000$ after excluding the atypical region) in each EVD. The densities of the three EVDs are plotted in Figure 36.

Ladder points, ladder epochs, and local record heights are easily computed with a few simple R commands. Finding minimal significant clusters–a smallest possible cluster of SNDs with a significant score–is more challenging. A naïve approach takes each SND within a high scoring segment as the start of some local score, then iteratively adds successive terms to local scores in parallel until one of the sums exceeds the threshold. The SNDs producing that sum constitute the first MSC. The process continues on the remaining sums to seek out additional, non-overlapping MSCs. The algorithm is $\mathcal{O}(n^2)$ in the number of SNDs. Such a brute force approach works here because alignment gaps split the problem into 186 small pieces, the largest of which contains fewer than 700 SNDs.

7.5 Acknowledgments

A version of this chapter appeared as Mau, Glasner, Darling, and Perna (2006). NTP and BM conceived the analysis, BM and AED drafted the manuscript and analyzed data. JDG assisted with interpretation of the results.

Chapter 8

Analysis of gene flux in enterobacteria

Genome comparisons of enteric bacteria demonstrate that an isolate of any given species will commonly contain substantial novel genetic content not found in other isolates of the same species (Tettelin et al., 2005). The mechanism by which bacteria acquire and maintain such lineage-specific content remains obscure, however the consensus belief is that such content has been acquired by lateral gene transfer (Ragan and Charlebois, 2002). One hypothesis suggests that novel content, occasionally referred to as ORFans, is commonly introduced into the chromosome by phage (Daubin and Ochman, 2004, Fischer and Eisenberg, 1999), and that phage harbor a wealth of biodiversity (Edwards and Rohwer, 2005, Sullivan et al., 2006). Indeed, the high A+T content of many novel genes relative to the bacterial chromosome supports such a hypothesis. However not all novel genes show a distinct A+T content or codon usage bias relative to the average chromosomal distributions. One possibility is that genes without high A+T content are also of phage origin and had high A+T content when they originally entered the chromosome, but have since ameliorated through directional selection to appear similar to the rest of the chromosome. Thus, such genes are thought to have been resident in the bacterial chromosome for a substantially longer period of time than novel genes with high A+T content. Another likely explanation involving phage transduction is that the gene had only recently been acquired by the phage population and the sequence had not yet gained an A+T bias prior integration with the recipient bacterial chromosome.

Given that microbes somehow rapidly acquire novel content, we must also consider the pattern of gene loss that allows microbes to maintain their characteristically compact genomes. If the acquisition rate and the deletion rate are approximately equal, we might expect to see arbitrary deletions of core genome content at a frequency equal to observations of novel content, unless deletions of acquired content were strongly favored. Frequent deletion of acquired content could arise due to either selective pressure or mutation bias, or some combination thereof. Specifically, deletions in preexisting genes could be strongly selected against, or acquired genic content could be inherently unstable, for example if it were flanked by mobile genetic elements.

When novel genes integrate into the chromosome, we may ask how they go on to integrate with the host microbe's regulatory system. Do such novel genes slowly come to be expressed by chance mutations upstream of the coding region? Given that enteric bacteria appear to have a mutational bias in favor of small deletions (Mira et al., 2001), it seems difficult to believe that a gene would be maintained long enough to acquire a functional promoter through random mutation before it were to be destroyed.

Is it possible that novel genes come preloaded with functional promoters and transcription factor binding sites? If this is the case, then it seems extremely likely that the regulatory logic upstream of the novel gene evolved in a closely related host, and thus the gene could be considered to be already "naturalized" to the host microbe, with only some fine tuning necessary for optimum fitness. In this scenario the gene may appear novel simply because it is not yet part of our sequence database, but it is hardly novel to the recipient organism.

Organism	Genome size
E. coli K12 MG1655	4654221
E. coli O157:H7 EDL933	5623806
E. coli CFT073	5231428
Shigella flexneri 2457T	4988914
Salmonella enterica Typhi Ty2	4791961
Yersinia pestis KIM	4781914
Yersinia pseudotuberculosis IP32953	4840899
Erwinia chrysanthemi 3937	4922802
Erwinia caratovora SCRI1043	5064019

Table 7: These nine enteric bacteria compose a phenotypically diverse set of organisms. The *E. coli*, *Shigella*, *Salmonella*, and *Yersinia* are human pathogens, while the *Erwinia* are plant pathogens. *E. coli* K12 MG1655 is a non-pathogenic laboratory strain.

A third intriguing possibility is that the operon structure of the microbial chromosome and the microbial gene expression system has evolved to explicitly favor acquisition of novel genetic content and its rapid incorporation into the host regulatory program. In such a model, novel genes could potentially integrate into an existing operon and immediately become expressed, without disrupting the expression of neighboring genes. In fact, previous studies have demonstrated a propensity for novel genes to integrate into existing operon structure (Price et al., 2006).

To better understand the role of gene acquisition and loss in bacteria we analyze multiple-genome alignments of enteric bacteria. We first study patterns of gene flux among a group of nine enteric bacteria from a broad phylogenetic spectrum (listed in Table 7), then narrow the scope of our analysis to a group of twelve complete *E. coli* and *Shigella* genomes (Table 8). By analyzing a set of distantly related taxa and a second group of closely-related taxa, we hope to gain insight into the rate at which recent mutations become fixed in microbial populations.

Organism	Genome size	Mode of pathogenesis
E. coli K12 MG1655	4654221	Non-pathogenic
E. coli O157:H7 EDL933	5623806	EHEC
E. coli O157:H7 Sakai	5594477	EHEC
E. coli HS	4643538	Non-pathogenic
E. coli E24377A	4980187	ETEC
E. coli CFT073	5231428	Uropathogenic
E. coli UTI89	5179971	Uropathogenic
Shigella boydii 227	4646520	Invasive
Shigella flexneri 2457T	4988914	Invasive
Shigella flexneri 301	4828821	Invasive
Shigella dysenteriae 197	4551958	Invasive
Shigella sonnei 046	5039661	Invasive

Table 8: Completely sequenced *E. coli* isolates presently analyzed. Many of these *E. coli* isolates are human pathogens, possibly skewing the results of our analysis. EHEC indicates enterohaemorrhagic *E. coli*, while ETEC indicates enterotoxigenic *E. coli*.

8.1 Results

The Progressive Mauve alignment system computes an alignment of the nine enteric genomes listed in Table 7 using 24 hours of compute time on a 2.8GHz Pentium 4 CPU. The resulting alignment contains 425 Locally Collinear Blocks with a total average length of 18.7Mbp of genomic sequence. Figure 37 shows a comparison of the structure of each genome as drawn by the Mauve visualization system. We then apply the backbone detection algorithm described in Chapter 5 to detect regions conserved among two or more genomes. Using a random-walk score threshold of 2727 yields a total of 23498 segments conserved among two or more taxa. Of these, 7658 segments are less than 5nt in length and result from merging pairwise segmental homology predictions with slightly different endpoints. We discard the short segments, yielding a set of 15840 high-confidence segments conserved among two or more genomes. Inclusion of segments > 5nt present in only a single genome under study yields a total of 31197 segments.

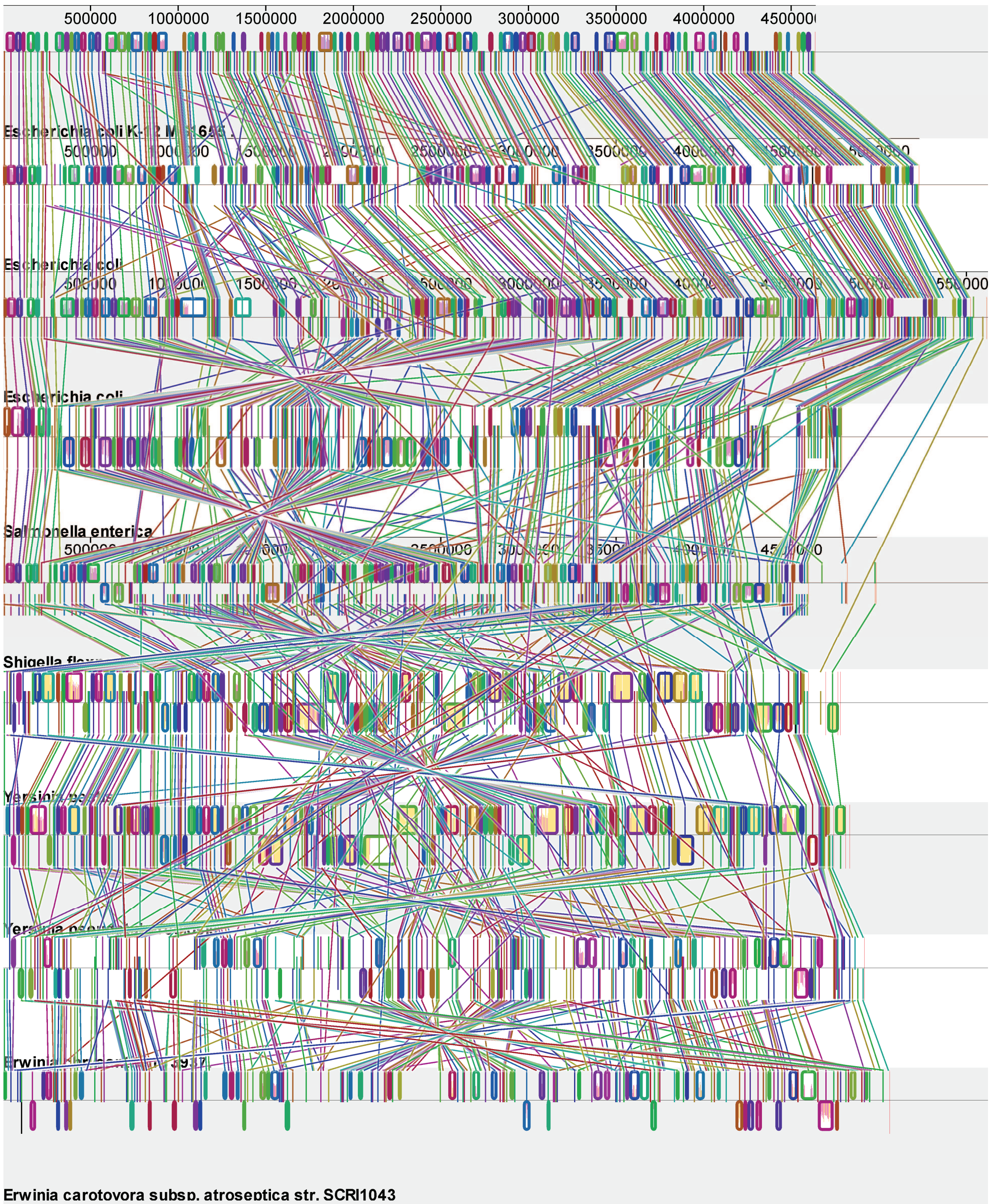

Figure 37: Mauve visualization of an alignment of four *E. coli* and *Shigella* genomes, one *Salmonella*, two *Yersinia*, and two *Erwinia* genomes. The alignment contains 346 locally collinear blocks and numerous lineage-specific segments. Each lineage has undergone substantial genomic rearrangement, resulting in the scrambled synteny portrait shown here.

Clustering of variable segments

Of the 31197 total segments, only 2810 are conserved among all taxa. If all differences in gene content arose from a single deletion or insertion event at a unique locus, the 2810 segments conserved among all taxa could accommodate a maximum of 2811 gene flux events, regardless of the phylogenetic relationship among taxa. Given that number of segments conserved among subsets of the taxa (31197-2810=28387) is much larger than 2810, it stands to reason that multiple events frequently occur at the same site and that "hotspots" of gene flux must exist.

A gene content phylogeny

We base our analysis on a genome-content guide tree computed by Progressive Mauve. The Progressive Mauve algorithm applies Neighbor-Joining to a distance matrix based on a combination of shared gene content and sequence identity. The resulting tree minimizes the total deviation between pairwise distances and branch lengths. We use the genome-content guide tree computed by Mauve as a basis for our analysis of patterns of gene flux. The inferred genome-content guide tree may conflict with a phylogeny based on nucleotide substitution data and may also conflict with the true phylogeny. For our analysis of gene flux, errors in phylogenetic inference will likely cause our subsequent analysis to underestimate the true number of gene flux events, because the tree is biased towards a topology that gives maximum conservation of gene content. Thus, we consider our estimates of gene flux to be conservative.

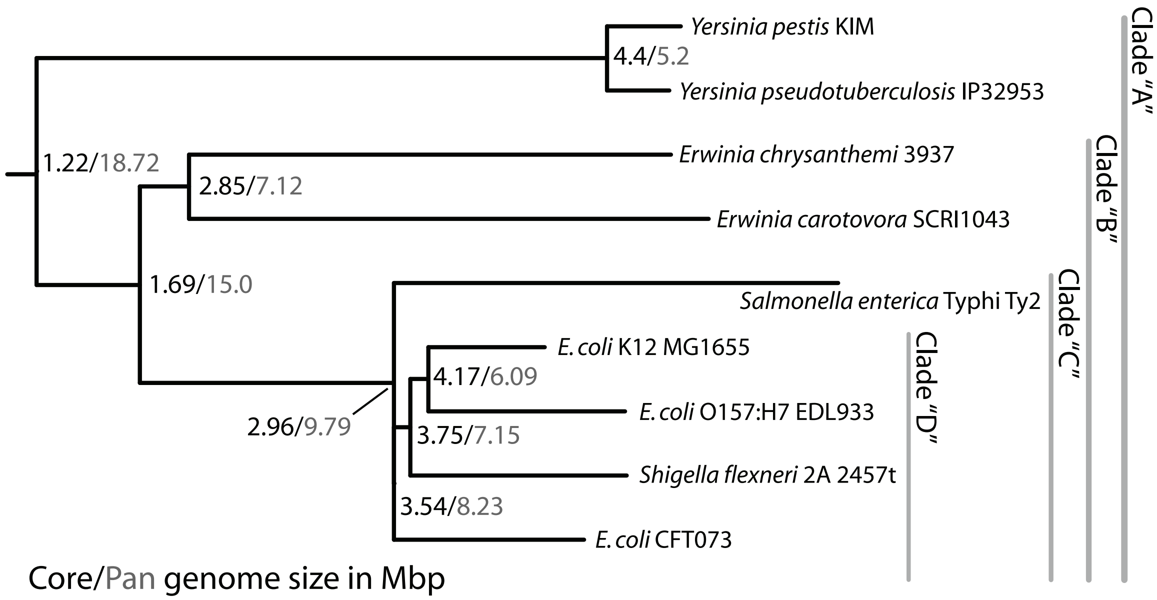

Figure 38: The pan-genome and core-genomes of clades within the family *Enterobacteriacae*. A genome-content phylogeny and multiple genome alignment was constructed for nine enteric bacteria using Progressive Mauve. The tree has been midpoint-rooted placing *Yersinia* as an outgroup here. The core genome size given at internal nodes represents the average amount of genome sequence conserved among all taxa below that node. The pan genome size represents the total amount of unique sequence present in all taxa below a given node. Homologous sequence present in two or more genomes gets counted only once towards the total pan-genome size.

8.1.1 The enteric core genome

Armed with a gene-content phylogeny, we consider the portion of the genome conserved among all members of a given clade to be the "core-genome" for that clade (Wertz et al., 2003). We define the complementary notion of a "pan-genome" as genome sequence present in any one or more members of the clade (Tettelin et al., 2005). The genome-content phylogeny for the nine enteric bacteria and the corresponding core- and pan-genome size for each clade is shown in Figure 38.

We analyze the functional distribution of genes present in the enteric core genome. Of the 4307 annotated CDS in *E. coli* K12, 29.6% of them have at least some portion conserved among all nine enteric genomes. Genes in *E. coli* K12 have been annotated with a gene function ontology called Multi-Fun, which was designed to specifically capture biological aspects of enteric bacteria (Serres and Riley, 2000). As *E. coli* K12 is the only genome with a robust Multi-Fun annotation, we restrict our analysis to clades containing K12. We label clades as "A", "B", "C", and "D", from most diverse to most specific as shown in Figure 38. Multi-Fun categories found to be under- and over-represented among genes in the core genome are shown in Table 9. We report the percent of conserved genes in each functional category, along with a χ^2 goodness-of-fit statistic for each category. We do not report p-values because a single gene may be assigned to several Multi-Fun categories, thus dependencies exist among categories.

As we would expect, several functional categories are heavily overrepresented among conserved genes. Specifically, genes with products involved in ribosomal structure, protein information transfer, cell division, and some aspects of metabolism show strong conservation. Some functional categories show significant underconservation, most notably gene products localized to the outer membrane, carbon utilization gene products, and electrochemical-driven transporter gene products.

We proceeded to compare the functional distributions of genes conserved at each successive subclade that includes *E. coli* K12, i.e. Clades "B", "C", and "D". Differences in conserved functional categories are indicated by the two leftmost columns in Tables 9, 10, and 11. Interestingly, outer membrane proteins are significantly under-conserved only at clades including the *Yersinia* genus, and carbon utilization gene products are under-conserved only when the *Erwinia* genus is included.

U	D	NumGenes	GenesInCat	Percent	χ^2	MfunLevel2Name
		1	43	2.33	10.8	cell structure; pilus"
		17	245	6.94	42.6	extrachromosomal; prophage genes and phage related functions""
		29	200	14.5	15.5	transport; Electrochemical potential driven transporterss
	*	11	75	14.7	5.67	location of gene products; outer membrane"
		236	1240	19	47	Unknown; No MultiFun Tag
		80	405	19.8	13.3	metabolism; carbon utilization
		121	285	42.5	15.8	metabolism; energy metabolism, carbon
		49	103	47.6	11.2	metabolism; macromolecule degradation
		123	255	48.2	29.8	transport; Primary Active Transporterss
		77	155	49.7	21	information transfer; DNA related
		437	828	52.8	150	location of gene products; cytoplasm"
		33	57	57.9	15.4	cell structure; peptidoglycan (murein)"
		265	442	60	137	metabolism; building block biosynthesiss
		219	359	61	119	information transfer; protein related
		45	67	67.2	31.8	cell processes; cell division"
		59	68	86.8	74.9	cell structure; ribosome"

Table 9: Annotated functions for products of genes that have some portion conserved among all nine enteric genomes. 29% of all genes annotated in *E. coli* K12 show evidence for conservation. Functional categories with a χ^2 value less than 5 not shown. An asterisk(*) in columns U and D indicates that the functional category appears differently at clades above (U) and below (D) this clade in the phylogeny.

U	D	NumGenes	GenesInCat	Percent	χ^2	MfunLevel2Name
		21	245	8.57	59.8	extrachromosomal; prophage genes and phage related functions""
		6	43	14	7.18	cell structure; pilus"
	*	54	200	27	8.14	transport; Electrochemical potential driven transporterss
		339	1240	27.3	47.8	Unknown; No MultiFun Tag
	*	118	405	29.1	11.4	metabolism; carbon utilization
*	*	182	367	49.6	9.02	metabolism; central intermediary metabolismm
	*	140	255	54.9	14.8	transport; Primary Active Transporterss
*		141	253	55.7	16.3	metabolism; macromolecules (cellular constituent)biosynthesiss
		160	285	56.1	19.4	metabolism; energy metabolism, carbon
		97	155	62.6	20.4	information transfer; DNA related
		531	828	64.1	124	location of gene products; cytoplasm"
		67	103	65	16.6	metabolism; macromolecule degradation
		246	359	68.5	75	information transfer; protein related
		336	442	76	147	metabolism; building block biosynthesiss
	*	44	57	77.2	20.2	cell structure; peptidoglycan (murein)"
		54	67	80.6	28.2	cell processes; cell division"
		62	68	91.2	45.4	cell structure; ribosome"

Table 10: 39.7% of K12 genes are conserved among members of clade "B". Functional categories with a χ^2 value less than 5 not shown. An asterisk(*) in columns U and D indicates that the functional category appears differently at clades above (U) and below (D) this clade in the phylogeny.

U	D	NumGenes	GenesInCat	Percent	χ^2	MfunLevel2Name
*		1	65	1.54	41.9	extrachromosomal; transposon related"
		31	245	12.7	109	extrachromosomal; prophage genes and phage related functions""
	*	15	43	34.9	6.77	cell structure; pilus"
	*	717	1240	57.8	17.2	Unknown; No MultiFun Tag
	*	130	155	83.9	6.17	information transfer; DNA related
		710	828	85.7	40.9	location of gene products; cytoplasm"
	*	89	103	86.4	5.46	metabolism; macromolecule degradation
		255	285	89.5	20.4	metabolism; energy metabolism, carbon
		322	359	89.7	26.2	information transfer; protein related
		227	253	89.7	18.5	metabolism; macromolecules (cellular constituent)biosynthesiss
	*	62	67	92.5	6.23	cell processes; cell division"
		413	442	93.4	44.1	metabolism; building block biosynthesiss
	*	66	68	97.1	8.81	cell structure; ribosome"

Table 11: 67.5% of K12 genes show evidence for conservation among members of clade "C". Functional categories with a χ^2 value less than 5 not shown. An asterisk(*) in columns U and D indicates that the functional category appears differently at clades above (U) and below (D) this clade in the phylogeny.

8.1.2 Variable genes, deletion, and lateral transfer

A number of segments are conserved among subsets of the genomes under study. We have analyzed these segments with an eye towards genes that have undergone lineage-specific deletion or apparent lateral transfer. Given an internal tree node at which both child nodes are also internal nodes, we define the notion of a *Hop 2* segment as a region which is present in some taxa below both child nodes, but not present in all taxa below either child. For example, a Hop 2 at the root of our tree is a segment present in only one of the two *Yersinia*, and also present in at least one member of clade "B", but not all members of clade "B". A Hop 2 pattern can only be explained by multiple independent deletions of the same segment or lateral gene transfer. Similarly, we define a *Hop 1* segment at an internal node as a region which is present in all of one child's taxa, and present in some, but not all, of the other child's taxa. An example at the root node would be a segment missing from one of the two *Yersinia* but universally present in all of *Erwinia*, *Salmonella*, *Shigella*, and *E. coli*.

We analyze the presence of Hop 1 and Hop 2 segments among members of Clades "A" and "B". Clade "A" shows evidence for 1138 Hop 1 segments, totalling 216Kbp, and 64 Hop 2 segments, totalling 9.9Kbp. The Hop 2 segments are candidates for lateral transfer between the *Yersinia* genus and members of Clade "B". Narrowing our phylogenetic scope to Clade "B", we find evidence for 1182 Hop 1 segments totalling 140Kbp. There are 238 Hop 2 segments at this clade, totalling 30.3Kbp.

Analysis of gene functions requires that the gene be present in K12. With that in mind, we analyzed the functional distribution of Hop segments in Clades "A" and "B". At Clade "A", 4.99% of K12 genes have some portion contained in a Hop 1 segment. Two functional categories show significant overrepresentation: "transport;

NumGenes	GenesInCat	Percent	χ^2	MfunLevel2Name
2	245	0.816	16.8	extrachromosomal; prophage genes and phage related functions""
80	1240	6.45	5.64	Unknown; No MultiFun Tag
82	700	11.7	9.11	cell structure; membrane"
66	555	11.9	8.02	location of gene products; inner membrane"
42	285	14.7	13.6	metabolism; energy metabolism, carbon
24	155	15.5	9.23	information transfer; DNA related
12	67	17.9	7.2	cell processes; cell division"
12	66	18.2	7.5	cell structure; surface antigens (ECA, O antigen of LPS)""
16	84	19	11.3	metabolism; metabolism of other compounds
49	253	19.4	36.2	metabolism; macromolecules (cellular constituent)biosynthesiss

Table 12: Functional categories of genes in K12 that show evidence for lineage-specific loss (Hop 1) among members of Clade "B". Several categories appear prone to lineage-specific loss. Functional categories with a χ^2 value less than 5 not shown.

Primary Active Transporters" and "location of gene products; periplasmic space" with 8.24% and 10.4% containing Hop 1 segments, respectively. Only 0.25% of K12 genes are part of Hop 2 segments at Clade "A", and no categories are significantly overrepresented.

Among members of Clade "B", 8.41% of K12 genes participate in a Hop 1 segment. Several functional categories show significant overrepresentation in Hop 1 segments at Clade "B", and are listed in Table 12. Some overrepresented categories make intuitive sense for pathogenic bacteria, for example, membrane proteins and surface antigens. Other functional categories such as DNA related information transfer show an unexpected tendency towards lineage-specific deletion. Only 0.88% of K12 genes participate in Hop 2 segments, and no functional categories show significant overrepresentation.

Choice of taxa is an important consideration for our analysis of Hop segments. Because Hop 2 segments can only be detected when both subclades below an internal node have at least two or more member genomes, our method cannot detect such segments at Clades "C" and "D". Adding another *Salmonella* genome and the *E. coli* UTI89 genome would enable detection of Hop 2 segments at "C" and "D". Moreover, sampling additional taxa at any clade would give more information about patterns of gene conservation both within and across clades.

Genes unique to *E. coli*

We continued by asking, "What, if any, genes tend to be specific to the *E. coli*?" We identified all genomic segments that showed homology only among members of clade "D", and analyzed their functional distribution. The results, shown in Table 13, indicate that very few functional categories are significantly unique to *E. coli*, while a large number are significantly non-unique. Interestingly, genes of unknown function are the only category apart from recombination-prone categories such as pili and transposons that show significant bias towards uniqueness in *E. coli*. Thus, we conclude that "We don't (yet) know what makes an *E. coli* an *E. coli*."

8.1.3 An analysis of twelve *E. coli* and *Shigella*

Having examined the gross changes in genetic content that exist among members of the *Enterobacteriacae*, we now turn towards a detailed analysis of *E. coli* and *Shigella* isolates. Although we find few functional gene categories that distinguish *E. coli* and *Shigella* from the remaining enteric bacteria, these microbes harbor a wealth of genetic diversity within their population that may be exploited to better understand their evolution.

We again apply the Progressive Mauve alignment system to align the twelve genomes listed in Table 8. The resulting alignment contains 345 Locally Collinear Blocks. There

NumGenes	**GenesInCat**	**Percent**	χ^2	**MfunLevel2Name**
2	68	2.94	17.9	cell structure; ribosome"
3	57	5.26	12.7	cell structure; peptidoglycan (murein)"
40	359	11.1	48.5	information transfer; protein related
8	67	11.9	8.37	cell processes; cell division"
54	442	12.2	53.7	metabolism; building block biosynthesiss
32	253	12.6	29.4	metabolism; macromolecules (cellular constituent)biosynthesiss
110	828	13.3	90	location of gene products; cytoplasm"
38	285	13.3	30.8	metabolism; energy metabolism, carbon
16	103	15.5	8.66	metabolism; macromolecule degradation
16	97	16.5	7.22	metabolism; energy production/transport
65	367	17.7	23.2	metabolism; central intermediary metabolism
16	90	17.8	5.63	transport; Transporters of Unknown Classificationn
22	123	17.9	7.58	cell processes; adaptation to stress
29	155	18.7	8.46	information transfer; DNA related
23	115	20	5.11	cell processes; protection
54	255	21.2	9.21	transport; Primary Active Transporterss
81	336	24.1	6.41	information transfer; RNA related
134	555	24.1	10.5	location of gene products; inner membrane"
113	459	24.6	7.65	transport; substrate
175	700	25	10.5	cell structure; membrane"
475	1240	38.3	15.9	Unknown; No MultiFun Tag
25	43	58.1	9.27	cell structure; pilus"
197	245	80.4	181	extrachromosomal; prophage genes and phage related functions""
60	65	92.3	74.3	extrachromosomal; transposon related"

Table 13: The genes that make *E. coli* an *E. coli*. Genes that have at least one segment present only in clade "D" (*E. coli* and *Shigella*) are identified and listed by functional category. Functional categories with a χ^2 value less than 5 not shown. What makes *E. coli* an *E. coli*? We don't know. K12 genes with unknown function are significantly more likely to be unique to *E. coli*.

are 1166 segments conserved among all *E. coli* and *Shigella* along with 12950 other segments present in some but not all genomes. Once again, strong evidence exists that these microbes have "hotspots" of gene flux.

Progresive Mauve computes a genome-content guide tree for the twelve genomes which places the *E. coli* and *Shigella* into separate clades (Figure 39). Studies of the phylogenetic signal in nucleotide substitutions among these microbes has revealed that they have undergone substantial amounts of homologous recombination (See Chapter 7). Each genome is a mosaic of many phylogenetic histories and thus a single 'true' whole-genome phylogeny does not exist for these taxa.

Functional distribution of conserved and lineage-specific content

We analyzed the functional distribution of genes in *E. coli* K12 that contain at least one segment conserved among all *E. coli* and *Shigella.* The results, shown in Table 14, indicate that a small number of functional categories show significant over- and under-conservation.

At the root of our genome content guide tree there are 727 Hop 1 segments with total length 340Kbp, and 1451 Hop 2 segments with total length 522Kbp. Given that *E. coli* and *Shigella* are one and the same species and undergo frequent homologous recombination (see Chapter 7), the relatively large number of Hop 2 segments relative to Hop 1 is not surprising. These segments likely result from lateral genetic transfer, although multiple independent deletion events may play a role in some cases. The number of Hop 2 segments can not be used to directly estimate the number of recombination events that have taken place, as multiple Hop 2 segments that support the same partitioning of taxa may be colocated on the chromosome and giving evidence for only a

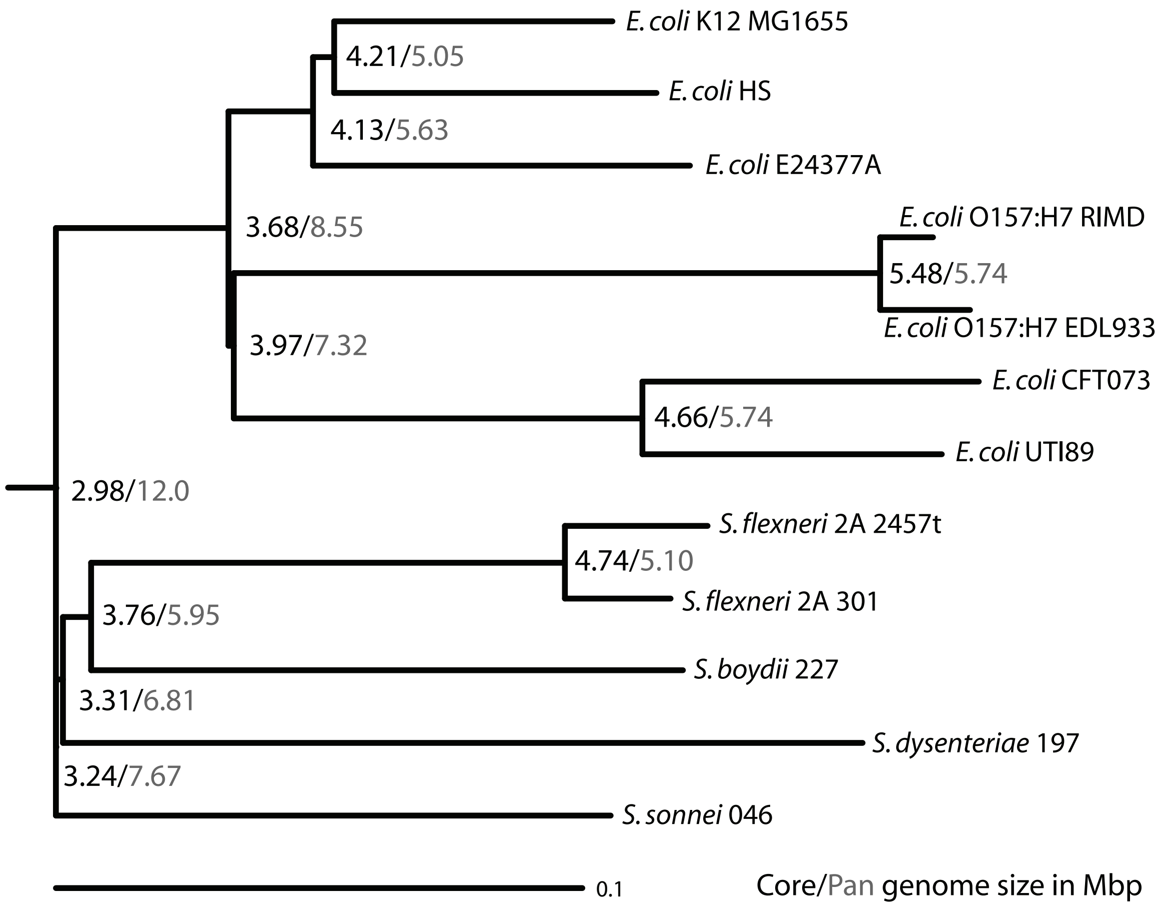

Figure 39: The pan-genome and core-genomes of *E. coli* and *Shigella*. A genome-content phylogeny and multiple genome alignment was constructed for twelve genomes using Progressive Mauve. A midpoint-root has been placed on the branch connecting *E. coli* and *Shigella*. The twelve microbes studied here are commonly considered to be the same species, yet harbor a tremendous amount of genetic diversity. Each microbe has an average genome size of 5Mbp, but on average contains only 3Mbp which is conserved among all taxa shown here. The pan-genome size of 12Mbp reflects all unique genetic content in these taxa, which averages to 750Kbp per sequenced genome.

NumGenes	GenesInCat	Percent	χ^2	MfunLevel2Name
3	65	4.62	38.8	extrachromosomal; transposon related"
28	245	11.4	117	extrachromosomal; prophage genes and phage related functions""
15	43	34.9	7.13	cell structure; pilus"
677	828	81.8	20.8	location of gene products; cytoplasm"
213	255	83.5	8.25	transport; Primary Active Transporterss
130	155	83.9	5.25	information transfer; DNA related
248	285	87	14	metabolism; energy metabolism, carbon
62	67	92.5	5.58	cell processes; cell division"
410	442	92.8	37.5	metabolism; building block biosynthesiss

Table 14: Functional distribution of genes showing conservation among all *E. coli* and *Shigella*. 68.6% of genes in *E. coli* K12 show evidence for conservation. Functional categories with a χ^2 value less than 5 not shown. Interestingly, both DNA Information Transfer and Building Block Biosynthesis categories show significantly above average conservation. These two functional categories were previously identified as especially prone to homologous recombination.

single recombination event.

Although only a small number of functional categories show unusual patterns of conservation, several functional categories show evidence for interesting patterns of gene loss and potential lateral transfer. A total of 10.9% of *E. coli* K12 genes contain Hop 1 segments, with the functional categories: "Unknown", "transport; Electrochemical potential driven transporters", and "metabolism; metabolism of other compounds" showing over-representation. 8.66% of *E. coli* K12 genes participate in Hop 2 segments at the root node, and the functional distribution is shown in Table 15.

Substantial intergenic variability

When gene flux occurs inside a pre-existing gene, it very likely breaks the gene. We evaluated the frequency with which gene flux occurs within annotated genes, versus entirely intergenic regions. To do so, we define a variable site in *E. coli* and *Shigella* as

NumGenes	GenesInCat	Percent	χ^2	MfunLevel2Name
138	1240	11.1	8.71	Unknown; No MultiFun Tag
15	75	20	11.1	location of gene products; outer membrane"
14	66	21.2	12	cell structure; surface antigens (ECA, O antigen of LPS)""

Table 15: Functional categories that are overrepresented in Hop 2 segments between *E. coli* and *Shigella*. 8.66% of genes in *E. coli* K12 participate in Hop 2 segments at this node. Functional categories with a χ^2 value less than 5 not shown.

any site between two adjacent segments conserved among all taxa (universally conserved segments). To avoid trivial variable sites due to small indels and slightly mispredicted homology boundaries, we consider only variable sites longer than 15nt. Given these criteria, there are 809 variable sites between universally conserved segments. Of these, 23 lie entirely within the boundaries of a single annotated gene and are likely multi-allelic genes or misannotated pseudogenes (a detailed inspection reveals both cases). A further 260 of the 809 variable sites have endpoints completely outside annotated CDS in all twelve genomes. 174 of the 260 variable segments with intergenic endpoints contain CDS, implying that novel genes have been either gained or lost at these sites. Finally 86 of the 260 intergenic variable segments contain no annotated CDS, implying that substantial variability exists in wholly-intergenic regions. Given that the vast majority of an enteric genome codes for protein, our observation that 260 of 809 variable segments (32%) have endpoints outside annotated gene boundaries supports the notion that a strong selective bias exists against gene flux that breaks genes.

Using the *E. coli* K12 annotation as a reference, we examined the characteristics of variable intergenic segments. Genes in enteric bacteria are frequently transcribed together in operons. Genes that are co-expressed in operons always occur adjacent to each other and are transcribed from the same strand. We classify neighboring genes as either *converging*, where the 3' end of both genes are adjacent, *diverging*, where the 5' end of both genes are adjacent, or *inline*, where the genes are adjacent and on the same strand.

Of the 260 intergenic variable sites, we find that 96 are flanked by converging CDS, 39 are flanked by diverging CDS, and 125 are flanked by inline CDS. To determine whether such a pattern would be observed merely by chance, we counted all intergenic sites with non-overlapping genes and performed a χ^2 test. There are 549 converging, 629 diverging, and 2549 inline CDS pairs in *E. coli* K12 that do not overlap, for a total of 3727 non-overlapping CDS pairs. We observe a significant overrepresentation of variable segments in converging regions ($\chi^2 = 89.17$, $p =$, 2 d.f.), the number of variable segments in diverging region does not significantly deviate from expectation, and we see a significant under-representation of variable segments between inline CDS ($\chi^2 = 14.72$, $p =$, 2 d.f.).

The high number of variable sites at converging CDS relative to diverging CDS is a pattern that would be expected if mutations at converging regions were less detrimental to the organism than mutations at diverging regions. In cases where new genes were not gained or lost, our observations of intergenic variability at inline CDS could be an artifact of subtle tuning of the microbes regulatory program by forming or destroying operon structures. In cases where genes have been acquired, they may be incorporating into existing operon structure.

Variability around tRNA and small regulatory RNAs

We examined the propensity of variable segments to cluster in the neighborhoods of tRNA and small non-coding RNAs annotated as misc_RNA in the *E. coli* K12 genome.

There are 49 annotated misc_RNA features in *E. coli* K12. Of our 260 variable intergenic segments, 16 of them either contain (7) or immediately neighbor (9) a misc_RNA feature. We find much greater variability in the neighborhood of misc_RNA than would be expected by chance alone ($\chi^2 = 50.44$, $p \leq 0.001$, 1 d.f.). tRNA are well known to be associated with so-called Genomic Islands of variability (Hacker and Kaper, 2000). There are 88 annotated tRNA in *E. coli* K12. We find 20 variable segments that either immediately neighbor (3), or contain (17) tRNA features. As expected, tRNA are associated with variable segments to a greater degree than chance would dictate ($\chi^2 = 34.78$, $p \leq 0.001$, 1 d.f.).

Alternalogs

When a variable site has undergone a single insertion or deletion event it partitions the taxa into two groups: those with a "null" allele and those with either novel content or the ancestral content. If multiple insertion or deletion events occur at the same site, we may see a pattern where each genome has an alternate non-null allele at a the variable site. We refer to such variable sites which have at least two different non-null alleles as *alternalogs*.

Of the 809 total variable sites, 285 of these fit our definition of an *alternalog*. Seven of these are completely contained within annotated gene boundaries in all twelve genomes and are likely multi-allelic genes. 97 alternalog sites have intergenic endpoints, of which 21 contain no annotated CDS internally implying they are entirely intergenic alternalogs. The remaining alternalog sites span gene boundaries, but are not entirely contained in any gene. A small number of alternalogs neighbor or contain misc_RNA features in *E. coli* K12, however the distribution is not as skewed as when all variable sites are considered. There are 14 alternalog sites that either neighbor (1) or contain (13) tRNA annotated in K12, a significant deviation from what would be expected by chance ($\chi^2 = 45.27$, $p \leq 0.001$, df=1).

Figure 40 illustrates a series of genes related to fimbriae and pilus production where multiple gene flux events have collocated. The resulting genomic structure is a patchwork with many genes differentially lost or gained in each genome.

8.2 Discussion

We have demonstrated that populations of enteric bacteria harbor a wealth of genetic diversity. Any *E. coli* isolate is likely to have between 10% and 20% sequence content not observed in other *E. coli* isolate. As we consider a progressively broader taxonomic scope in our analysis, the total amount of core genome content decreases, eventually reaching approximately 1Mbp. Given the extreme amount of diversity within the *E. coli* and *Shigella*, it is clear that portions of the core-genome are resistant to gene flux, otherwise no conserved sequence would remain over the long period of divergence between the enteric species we study here. Thus, it appears that novel content is usually transient, but occasionally becomes fixed in the population through positive selection.

In some cases, newly acquired content may appear to replace content that previously existed at a given locus. The novel content may initially "infect" the first member of the population through simple insertion, and subsequent deletion of adjacent content would yield an apparent replacement, or alternalog. If the novel content is advantageous, population members with the replacement may experience positive selection. Previous studies suggest that the population size of *E. coli* may be very large (Berg, 1996). If

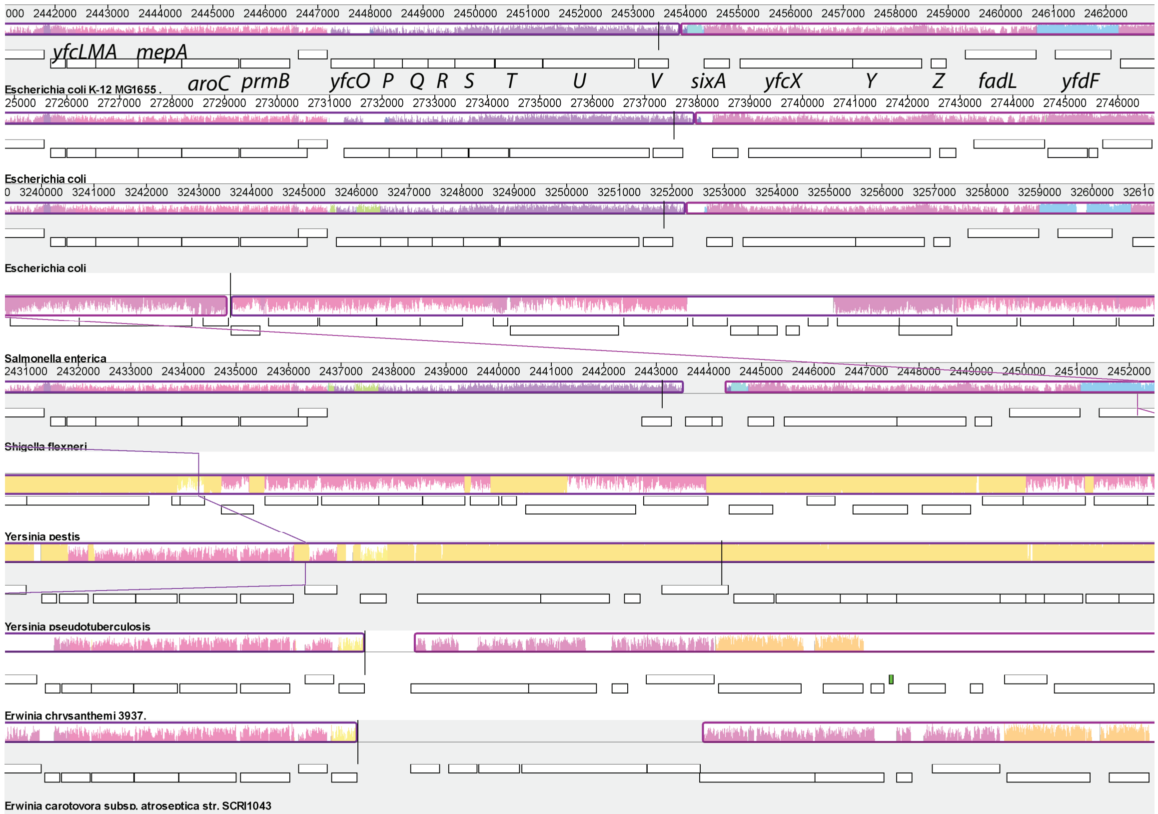

Figure 40: Mauve visualization of the mosaic structure of the *yfcOPQRSTUV* gene cluster and neighboring regions. The *yfc* gene products have fimbrial and pilus-related functions. Regions conserved among all nine taxa are shown in pink, and the height of the pink similarity plot indicates the degree of conservation for such regions. Segments conserved among only the *Yersiniae* are shown in yellow, while other colors represent regions conserved among different subsets of the taxa. The white rectangular blocks indicate the locations of annotated genes. The *yfc* gene cluster is present only in the *E. coli* and *Shigella*. The *yfcO* gene appears to have three different alleles, one shown as green in the third and fifth genomes (O157 and *Shigella*) , the other two alleles shown as white in the first and second genomes (K12 and CFT073).

microbial population sizes are indeed large, we would expect genetic drift to fix neutral acquisitions or deletions at a very low rate. In such a scenario we expect the same neutral acquisition or deletion to be observed in more than one independently sampled member of the population very rarely unless the mutation occurred a “long” time ago. If microbes have very high recombination rates, however, the process of genetic drift could be substantially accelerated (Novozhilov et al., 2005), and recent acquisitions could rapidly “invade” the population even if they are neutral or mildly deleterious. Unlike sexual organisms, intraspecific recombination in microbes is not tied to generation time, but rather appears to be episodic (REEVES, 1960). Without an upper bound on recombination rate, it may prove difficult to distinguish alleles whose frequency in the population has recently increased due to genetic drift from those under strong positive selection.

It may be possible to estimate the overall recombination rate in microbes by investigating patterns of shared novel content and deletion mutations in conjunction with nucleotide substitution data. Given baseline estimates of recombination rates along with (unrealistic) assumptions that the recombination rate is constant over time and that all portions of the chromosome are uniformly subject to recombination, it becomes possible to identify novel acquisitions and deletions that have been subject to positive selection. Detailed knowledge of the selective forces at play during the process of gene flux would be a great boon to the field of microbial population genetics and our understanding of nature as a whole.

Finally, we have identified significant amounts of gene flux in entirely intergenic segments, and discovered an unexpected correlation between gene flux and annotated misc_RNA features. misc_RNA features are typically small non-coding RNAs that play a role in gene regulation. Although it has been previously known that small RNAs are rarely conserved across species, the extent of their diversity within species was heretofore unappreciated. Further study will undoubtedly shed light on the role gene flux plays in the evolution of gene regulation in enteric bacteria.

Chapter 9

Bayesian models of genome evolution

9.1 Background

Current genome alignment systems make several simplifying assumptions that limit their value for characterizing rates and patterns of large-scale evolution. Genome aligners typically report the single highest scoring genome alignment according to their scoring metric without considering uncertainty in the best-scoring alignment. Uncertainty in the alignment affects every aspect of downstream analysis of the alignment, from phylogenetic shadowing for functional inference, to investigation of the breakpoints of recombination. Clearly, uncertainty should be considered if at all possible.

Assessing uncertainty in genome alignments requires a more statistically rigorous treatment of genome alignment than that used by state-of-the-art genome alignment methods. Previous studies of uncertainty in gapped alignments indicate that analytical calculation of alignment probability is far too expensive even for short alignments of few taxa with simple evolutionary models (Miklòs et al., 2004). For this reason, Bayesian MCMC methods must be employed. Their slow adoption has been in part due to the complexity of implementation and in part due to the the computational cost of sampling many alignments versus calculating a single highest-scoring alignment. However, recent advances in Bayesian alignment sampling have demonstrated its feasibility for short sequences (Lunter et al., 2005, Redelings and Suchard, 2005, Suchard and Redelings, 2006, Fleissner et al., 2005).

We presently describe a Bayesian model of genome evolution that can be applied for analysis of microbial genomes. The model has not been implemented, however, we discuss practical considerations for its implementation.

9.2 A model of genome evolution

The first step towards development of a statistical method for genome alignment is the elucidation of a stochastic model of evolution which captures the most important aspects of genome evolution. A tradeoff exists in model complexity, as increasingly complex models promise to provide more accurate descriptions of the evolutionary process, but come at the cost of requiring increasingly large amounts of data for accurate model parameterization and greater computational effort for inference. Keeping that tradeoff in mind, I propose a simplistic model of genome evolution that incorporates several of the major evolutionary forces we have observed to affect enteric bacteria.

At a bare minimum, a probabilistic model of genome evolution must incorporate the following mutation operators: nucleotide substitution, insertion and deletion of arbitrarily sized segments, and rearrangement by inversion. To maintain model simplicity, we do not incorporate rearrangement by transposition or duplication/loss processes, as a series of overlapping inversion events could produce similar genome arrangements, albeit with additional rearrangement events. Acquisition and loss of entire genes and operons can be modeled by the indel process with arbitrarily long segments. The model assumes a phylogenetic tree relating the genome sequences, with branch lengths that represent divergence times. Our previous observation of significant heterotachy in mutation rates for genome rearrangement and gene flux suggests that each mutation type should have per-branch rates. A full list of model parameters is given in Table 16.

The proposed model can be viewed as a merge and extension of two previously described stochastic models of evolution. We incorporate the long-indel model of sequence evolution used by Bali-Phy (Redelings and Suchard, 2005), extending the model slightly to separate branch-lengths from mutation rates and allowing indel rates to be independent of substitution rates. We then incorporate the model of genome rearrangement by inversion described by Larget et al. (2004), also allowing inversion events to have branch-specific rates.

9.2.1 Notation

Multiple sequence alignments are typically displayed in row-column format with gap characters spacing the sequences such that homologous regions align in columns. The row-column format mutliple alignment is imprecise, however, because more than one row-column alignment can encode identical homology information, differing only in the placement of gap characters. We adopt a homology structure based on a partial order graph, which yields an unambiguous means to record homology infomation (Lee et al., 2002). A genome alignment consists of several homology structures–one for each Locally Collinear Block (LCB)–the set of which are denoted $\mathbf{H}$. In the proposed model the set of LCBs is denoted by $\mathbf{Y}$. To simplify calculation each LCB is defined as an interval of at least one nucleotide present in all of the k genomes under study. Thus, a given LCB Y_i can be parameterized by its left and right-end coordinates in each genome: $Y_i = \{\langle Y_i.left_1, Y_i.right_1\rangle, \ldots, \langle Y_i.left_k, Y_i.right_k\rangle\}$. All or part of the region covered

param	Parameter Description	prior
$\mathbf{G}$	Observed set of genome sequences	fixed
Ψ	Tree topology with k leaves	uniform
τ	A vector of branch lengths for Ψ	$\tau \sim \Gamma(\tau_\alpha, \tau_\lambda)$
τ_α	Branch length gamma distribution hyperparameter	fixed
τ_λ	Branch length gamma distribution hyperparameter	fixed
N_b	Per-branch rates of nucleotide substitution	$N_b \sim \Gamma(N_\alpha, N_\lambda)$
N_α	Substitution rate gamma distribution hyperparameter	fixed
N_λ	Substitution rate gamma distribution hyperparameter	fixed
Q	Substitution rate matrix	fixed
α	Gamma-distributed substitution rate heterogeneity shape parameter	uniform
D_l	Mean indel length	uniform$(0 \ldots 100)$
D_b	Per-branch indel rates	$D_b \sim \Gamma(D_\alpha, D_\lambda)$
D_x	Per-branch indel counts	$D_x \mid D_b\tau_b \sim$ Poisson$(D_b\tau_b)$
D_r	Per-branch set of indel sites	$length(D_r) \sim$ Geom(D_l)
D_s	Per-branch set of indel event times	uniform$(0, \tau_b)$
D_α	Indel rate gamma distribution hyperparameter	fixed
D_λ	Indel rate gamma distribution hyperparameter	fixed
$\mathbf{D}$	The set of all indel variables, excluding D_α and D_λ	
I_b	Per-branch inversion rates	$I_b \sim \Gamma(I_\alpha, I_\lambda)$
I_x	Per-branch inversion counts	$I_x \mid I_b\tau_b \sim$ Poisson$(I_b\tau_b)$
I_r	Per-branch set of inversion event breakpoints	uniform$(\mathbf{G})$
I_s	Per-branch set of inversion event times	uniform$(0, \tau_b)$
I_α	Inversion rate gamma distribution hyperparameter	fixed
I_λ	Inversion rate gamma distribution hyperparameter	fixed
$\mathbf{I}$	The set of all inversion variables, excluding I_α and I_λ	
$\mathbf{Y}$	The set of locally collinear blocks (nuisance parameter)	uniform
$\mathbf{H}$	Set of per-LCB homology structures	uniform

Table 16: Parameters for a Bayesian model of genome evolution.

by the LCB can be homologous among the two sequences, as dictated by a homology structure H_i. In this model, every nucleotide in every genome is part of *some* LCB. In addition to providing a framework for the homology structures, the LCBs allow the genome sequences to be reduced to signed permutations for rearrangement history inference.

Given a set of genome sequences $\mathbf{G} = \{g_1, \ldots, g_k\}$, we denote the length of the i^{th} genome sequence as $|g_i|$.

9.3 The posterior distribution

We write the complete set of model parameters as $\mathbf{\Theta} = \{\Psi, \tau, N_b, \alpha, \mathbf{D}, \mathbf{I}, \mathbf{Y}, \mathbf{H}\}$, and the set of fixed data as $\mathbf{\Omega} = \{\mathbf{G}, N_\alpha, N_\lambda, Q, D_\alpha, D_\lambda, I_\alpha, I_\lambda\}$. The unnormalized joint posterior distribution of model parameters can be expressed as:

$$
\begin{aligned}
P(\mathbf{\Theta}|\mathbf{\Omega}) \quad \propto \quad & P(\Psi)P(\mathbf{Y})P(\tau|\tau_\alpha, \tau_\lambda)P(N_b|N_\alpha, N_\lambda) \cdot \\
& P(D_b|D_\alpha, D_\lambda)P(D_l)P(D_x|D_b\tau)P(D_r|D_x, D_l)P(D_s|\tau, D_x) \cdot \\
& P(I_b|I_\alpha, I_\lambda)P(I_x|I_b\tau)P(I_r|I_x)P(I_s|\tau, I_x) \cdot \\
& P(\mathbf{H}|\mathbf{G}, \tau, Q, N_b)P(\mathbf{Y}|\mathbf{I})P(\mathbf{H}|\mathbf{D})P(\mathbf{G}|\mathbf{HY})
\end{aligned}
$$

where

$$P(\Psi) = \frac{1}{(2k-5)!!}$$

when $k > 2$ (more than two genomes). The number of possible LCB configurations, denoted $Y\#$ can be expressed as

$$Y\# = \sum_{i=1}^{n} (i!2^i)^{k-1} \prod_{j=1}^{k} \binom{|g_j|-1}{i-1} \tag{9.1}$$

where n is the length of the shortest genome. Thus, we can write $P(\mathbf{Y}) = \frac{1}{Y\#}$. Intuitively, we can think of $Y\#$ as counting all possible LCB structures among the genomes. The sum term accounts for the fact that there are anywhere between 1 and n collinear segments in each genome, and the second (product) term considers all possible ways the collinear segments could be combined across genomes into LCBs.

The conditional probabilities for $\mathbf{I}$ follow from (Larget et al., 2004). Briefly, we define a set of per-branch inversion rates I_b coming from a gamma distribution with shape parameter I_α and scale parameter I_λ. I_b is a vector with $2k-3$ elements, the number of edges in the tree. We then define a total per-branch count of inversions I_x which is Poisson distributed with per-branch intensities equal to $I_b\tau$, i.e. the product of inversion rate and branch time. We go on to define I_r as the actual inversion events that took place along each branch, and we define a set of per-branch inversion event times I_s, which are uniformly distributed along the branch (which has I_x events and τ units of time).

The conditional probabilities for $\mathbf{D}$ are similar to those for $\mathbf{I}$, but include some bias towards particular indel sizes, whereas our prior on inversion events treats all events as equally likely. Again we define D_b as a per-branch mutation rate for indels, gamma-distributed with shape and scale D_α and D_λ, respectively. We sample a per-branch count of indel events, which is Poisson distributed with per-branch intensities equal to $D_b\tau$. D_r represents the actual indel events taking place along each branch, and D_s are the corresponding event times uniformly distributed along the branch. The term $P(D_r|D_x, D_l)$ reflects the probability of observing a series of D_x indel events given that indel lengths are distributed according to a geometric distribution with mean D_l. $P(D_l)$ is the prior probability of a given mean indel length, which we take to be uniformly distributed between 0 and 100.

The term $P(\mathbf{H}|\mathbf{G}, \tau, Q, N_b)$ calculates the probability of the homology structure given the genome sequences. The probability of the homology structure depends on the probability of the nucleotide substitution events among members of $\mathbf{G}$ implied by the homology structure. Substitution probabilities can be calculated using Felsenstein's peeling algorithm (Felsenstein, 2004).

The final three terms in the unnormalized posterior are indicator terms whose probability is 1 if the proposed structures are consistent with the data. Specifically, we write these as:

$$\begin{aligned}
P(\mathbf{Y}|\mathbf{I}) &= \mathbf{1}_{\{(\Psi, I_x, I_r) \hookrightarrow \mathbf{Y}\}} \\
P(\mathbf{H}|\mathbf{D}) &= \prod_{i=1}^{|\mathbf{H}|} \mathbf{1}_{\{(\Psi, D_x, D_r) \hookrightarrow H_i\}} \\
P(\mathbf{G}|\mathbf{H}, \mathbf{Y}) &= \mathbf{1}_{\{(\mathbf{Y}, \mathbf{H}) \hookrightarrow \mathbf{G}\}}
\end{aligned}$$

Where $P(\mathbf{Y}|\mathbf{I})$ indicates whether the proposed rearrangement events are consistent with the proposed LCB structure $\mathbf{Y}$. The term $P(\mathbf{H}|\mathbf{D})$ indicates whether the proposed indel events are consistent with the proposed homology structure. Finally, the term $P(\mathbf{G}|\mathbf{H}, \mathbf{Y})$ has value 1 when the genome sequence data is consistent with the proposed homology structure and LCB structure.

Inference under the model

The marginal probability distribution of model variables provides a basis for biological insight. For example, a probability distribution over the breakpoints of rearrangement encoded by I_r can identify likely positions on the chromosome where a rearrangement event was initiated. By studying the sequence motif at that site, it may be possible to infer whether the rearrangement event was mediated by homologous recombination, an IS or transposable element, or illegitimate recombination. The probability distribution over homology structures can inform us which regions are likely to have been conserved throughout evolution, and also places a distribution over endpoints of gene acquisition and differential gene loss. We can then investigate the surrounding sequence for evidence of phage involvement or other recombination mechanisms.

9.3.1 Sampling from the model

Due to the complexity of the model, direct analytical calculation of marginal probabilities for each variable is not possible. Instead, it will be necessary to sample likely values for each of the above listed variables using Markov-chain Monte-Carlo. Towards this end, the sampling methodology and model for rearrangement events follows the lead of Larget et al. (2002) whereby inversions were specified by an event count per branch (I_x) with event times given by a Poisson process (I_s). At proposal steps requiring modification of the rearrangement scenario, a method similar to Larget et al. (2004) would be used to propose a plausible rearrangement scenario. Their method proposes an inversion that reduces the overall inversion distance with high probability, and with low probability, proposes inversions that either maintain the same inversion distance or increase the distance. It is likely that use of a parallel Metropolis-coupled sampling strategy would be necessary to improve mixing speed.

Given an alignment and a phylogenetic tree, it is possible to quickly calculate the minimum number of indel events that could give rise to the observed alignment. Thus the indel events can be parameterized in a manner similar to rearrangements, namely by having per-branch parameters for the actual series of events, their times, and an event count whose prior is biased by the minimum possible number of events. We can then sample indel events as rearrangements are sampled; specifically, indels that reduce the total number of remaining events required to explain the homology structure are sampled with high probability. Indels that leave the number of remaining events constant are sampled with small probability while other indel events are sampled with an even smaller probability.

Because genome sequences can be several megabases in length, the alignment sampling method must use anchored alignment techniques. With some high probability, the sampler proposes a set of anchors and LCBs ($\mathbf{Y}$) consistent with the high scoring local alignments. LCBs inconsistent with the set of high scoring local alignments should be proposed with lower probability. Among the high probability anchor proposals, it may be possible to bias the proposal distribution toward LCB configurations with fewer rearrangement breakpoints.

Given a set of LCBs ($\mathbf{Y}$) and alignment anchors, an alignment can be proposed by combining the traditional dynamic programming approach for anchored alignment with a stochastic traceback step. In stochastic traceback, rather than selecting the highest scoring path at each step of the traceback procedure, a path is chosen randomly with probability proportional to its score. Lunter et al. (2005) describes how to calculate proposal probabilities for standard alignment, and we anticipate extending the methodology to anchored alignment. Bali-Phy uses a slightly different mechanism to propose new alignments among taxa which appears to offer better mixing (Redelings and Suchard, 2005). Thus, if their approach can be combined with an anchoring strategy it may be preferable.

The MCMC sampler moves through a series of states $\mathbf{X} = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$, each of which is represented by a particular set of parameter values. Transitions between states are achieved by a set of proposal update mechansisms. The quality of proposal updates is critical to achieving high acceptance ratios and good mixing behavior for the Markov chain. The sampler uses the following proposal update mechanisms:

1. Update tree topology (using mechanisms such as NNI and TBR)
2. Update a pair of breakpoint positions for a sequence subject to existing anchor constraints (recalculate alignment in new regions)
3. Sample a new anchor for a position in a sequence (update rearrangement scenario)
4. Disable an anchor (possibly update rearrangement scenario)
5. Disable an entire LCB
6. Sample a new rearrangement scenario
7. Sample a new indel scenario
8. Resample part of the alignment

The first proposal mechanism, an update to the tree topology, requires a corresponding update of rearrangement scenarios and indel events, although the homology structure is invariant. The second proposal mechanism would require changes to the homology structure and possibly indel events, although the LCB structure and rearrangement events could remain invariant. Finally, resampling the alignment would also require corresponding updates to the indel scenarios. Future work to derive Metropolis-Hastings acceptance ratios for each proposal type will be required before any of these proposal mechanisms can be implemented in software.

9.4 Discussion

The proposed model takes an intentionally simplified view of the forces at play during genome evolution. The model ignores rearrangement mediated by transposition, block interchange, and duplication-loss processes. The model does not include segmental duplication, which we feel is an acceptable simplification when modeling bacterial genomes that appear to have strong selective pressure to maintain small genome size.

Perhaps more importantly, the model does not include any notion of lateral transfer among population members. Isolates of enteric bacteria have provided strong evidence for homologous recombination's role in exchanging genetic material among members of a population. When such recombination takes place, a single tree topology no longer represents the true history of the genomes under study. Thus, the proposed model may have serious shortcomings in its representation of population-level evolution. However, cross-species recombination has been demonstrated to be much rarer than intraspecific recombination (Beiko et al., 2005, Mau et al., 2006). Therefore it seems plausible that the model could be applied to a set of genomes so long as no two genomes are members of the same species (*i.e.* little homologous recombination has taken place).

Although other work has used a single likelihood calculation for the probability of a tree given both indels and nucleotide substitutions in a TKF91 model, the method can only accomodate single nucleotide indels. Because larger indels obviously occur we consider our model more realistic. Our more realistic model comes at the expense of sampling full indel histories for the genomes under study. It remains to be seen whether the approach is computationally tractable.

Appendix A

Palindromic seed patterns

Weight	Pattern	Seed Rank by Sequence Identity					
		65%	70%	75%	80%	85%	90%
5	`1**111**1`	2	2	2	2	2	7
6	`11**1*1**11`	2	2	2	2	2	3
7	`1*11***1***11*1`	2	2	2	2	2	2
8	`111**1*1**111`	2	2	2	2	2	2
9	`111**1**1**1**111`	3	2	2	2	2	2
10	`111*1**1**1**1*111`	5	3	2	2	2	2
11	`111*1*1**1**1*1*111`	3	2	2	2	2	2
12	`1111*1**11**1*1111`	1	1	3	3	2	3
13	`111*1*11**1**11*1*111`	2	1	2	2	2	2
14	`1111*1*11**11*1*1111`	1	1	2	2	2	2
15	`1111*11**1*1*1**11*1111`	3	2	2	2	3	4
16	`111*111**1*11*1**111*111`	5	4	2	2	2	2
18	`11111*1*11**11**11*1*11111`	2	2	2	2	2	2
19	`11111*1*11**111**11*1*11111`	6	4	2	3	4	6
20	`11111*11*111**111*11*11111`	1	1	8	> 10	> 10	> 10
21	`111111**11*1*111*1*11**111111`	> 10	3	2	1	1	1

Table 17: Second-most sensitive palindromic spaced seeds used by `procrastAligner`. The sensitivity ranking of a seed at various levels of sequence identity is given in the columns at right. A seed with rank 1 is the most sensitive seed pattern for a given weight and percent sequence identity. The default seeds used by `procrastAligner` are listed in Chapter 3, while these seeds are the second-most sensitive set of optional seeds.

Weight	Pattern	Seed Rank by Sequence Identity					
		65%	70%	75%	80%	85%	90%
5	11**1**11	3	3	3	3	3	2
6	11*1*1*11	3	3	3	3	3	1
7	11*1***1***1*11	3	3	3	3	3	3
8	11**1*1*1*1**11	4	4	3	4	4	4
9	111**1*1*1**111	2	3	3	3	3	3
10	111*1**11**1*111	2	2	3	3	3	3
11	111**1**1*1*1**1**111	9	6	3	3	3	3
12	111*11*1***1*11*111	3	2	2	2	3	6
13	111*1**11*1*11**1*111	5	3	4	3	4	6
14	1111*1*1**11**1*1*1111	4	4	3	3	4	5
15	1111**11*1*1*1*11**1111	5	3	3	3	2	2
16	11111**11*1*1*11**11111	4	3	4	3	3	4
18	1111*11**11*1*1*11**11*1111	> 10	6	3	3	3	3
19	1111*11*111*1*111*11*1111	1	1	4	10	> 10	> 10
20	11111*1*111**11**111*1*11111	> 10	> 10	1	2	3	3
21	111111*1*11*111*11*1*111111	3	2	4	10	> 10	7

Table 18: Third-most sensitive palindromic spaced seeds used by `procrastAligner`. The sensitivity ranking of a seed at various levels of sequence identity is given in the columns at right. A seed with rank 1 is the most sensitive seed pattern for a given weight and percent sequence identity. The default seeds used by `procrastAligner` are listed in Chapter 3, while these seeds are the third-most sensitive set of optional seeds.

Appendix B

Description of the Mauve Multi-MUM search algorithm

The multi-MUM search algorithm described herein is a seed-and-extend method based on the method that can identify both multi-MUMs occurring in all genomes under study in addition to those occurring only in subsets of the genomes being searched. The multi-MUM search algorithm has time complexity $O(G^2n + Gn \log Gn)$ where G is again the number of genomes and n the length of the longest genome. Further, the random-access memory requirements are proportional to the number of multi-MUMs found, not n, allowing it to efficiently tackle large data sets. $O(Gn)$ disk space is used to store sequentially accessed data structures.

The algorithm proceeds by constructing a sorted list of k-mers for each genome $g \in G$. The sorted k-mer lists are then scanned to identify kmers that occur in two or more sequences but that occur at most once in any sequence. If a multi-MUM that subsumes the k-mer match has not yet been discovered, then the match seeds an extension in each genome until a mismatch occurs. When a mismatch occurs an extension is seeded in the subset of sequences that are still identical, but only if a subsuming multi-MUM has not yet been discovered.

Given a match seed, a key feature of our algorithm is its ability to efficiently determine whether an existing multi-MUM subsumes the seed. Mauve uses a hash table to track known matches. The hash function $h(M)$ for a match M yields a quantity we refer to as the generalized offset of a match M. Using the notation of multi-MUMs introduced in the primary manuscript, $h(M)$ can be written as $h(M) = \sum_{j=1}^{G} |M.S_j \dots M.S_1|$. In order to mitigate the effects of potential hash collisions, each bucket of the hash table uses a binary search tree to store matches.

For the purposes of time complexity analysis, the matching algorithm can be deconstructed into four primary components: Sorted Mer List (SML) construction, seed match identification, seed lookup in the known match hash table, and seed extension. SML construction can be accomplished in $O(Gn)$ (linear) time using radix sort methods. Identifying seed matches from the Sorted Mer Lists requires a single sequential scan through each SML and is thus also $O(Gn)$. The seed lookup phase can be executed at most once for every multi-MUM seed. Because there are Gn mers, the largest possible number of unique mer-matches is $\frac{Gn}{2}$. If all of these mer-matches were to hash to the same bucket then a tree search and insertion would be required for every seed match. Using a splay tree (Sleator and Tarjan, 1985), the amortized time complexity for Gn tree lookups and insertions is $O(Gn \log Gn)$. The amount of match extension depends on the number and size of multi-MUMs identified. Because we are identifying MUMs, each nucleotide can be a part of at most 2 MUMs on the forward strand and 2 MUMs on the reverse strand, for a total of 4 MUMs. Furthermore, it holds that any 2 nucleotide can be a part of at most 4 multi-MUMs with a given multiplicity. Thus each nucleotide can be a part of $4G$ multi-MUMs, or just $O(G)$ multi-MUMs. For a given multiplicity m, the largest possible amount of extension work depends on the maximum possible number of matching mers at that multiplicity: $\frac{Gn}{m}$. Further, each extension at a particular multiplicity m requires m character comparisons. Thus the maximum number of character comparisons for a given multiplicity is $m\frac{Gn}{m}$ or just Gn, and since there are G multiplicity levels, the maximum number of comparisons to find all multi-MUMs is G^2n.

By adding the contributions each of the algorithmŠs four components make toward the total running time, we arrive at $Gn+Gn+Gn\log Gn+G^2n$. In asymptotic notation, the Gn terms are subsumed by G^2n, leaving $O(G^2n+Gn\log Gn)$. It is important to note that although suffix tree algorithms provide better asymptotic time complexity than our seed-and-extend method, in practice our implementation is very fast and space efficient. Furthermore, the seed matching technique can be easily modified to use weighted/spaced seeds, allowing inexact string matching not possible with suffix tree-like data structures in the same low asymptotic time complexity.

Appendix C

Partitioning matches into collinear subsets

As part of the anchor selection process, Mauve must partition the initial set of multi-MUMs $\mathbf{M}$ into collinear subsets. To do so, Mauve implements a breakpoint analysis algorithm based on the description of breakpoints given by Blanchette et al. (1997). We refer to the resulting collinear sets of multi-MUMs as LCBs. An LCB can be defined formally as a maximal collinear subset of the matches in $\mathbf{M}$, or $lcb \subseteq \mathbf{M}$ where M_i is the i^{th} multi-MUM in the LCB. The MUMs that constitute an LCB must satisfy a total ordering property such that $M_i.S_j \leq M_{i+1}.S_j$ holds for all i, $1 \leq i \leq |lcb|$ and all j, $1 \leq j \leq G$.

Mauve uses a standard breakpoint determination algorithm to partition the set of multi-MUMs into a set of LCBs. First, Mauve orders the multi-MUMs in M on $|M_i.S_0|$. Next, a monotonically increasing label between 1 and $|M|$ is assigned to each MUM corresponding to the index of the MUM in the ordering on $|M_i.S_0|$. We will refer to the label of the i^{th} multi-MUM as $M_i.label$. Note that $M_i.label \in \mathbb{N}$. Next, the set of multi-MUMs is repeatedly reordered based on $|M_i.S_j|$ for $j = 2 \ldots G$. After each reordering, the set of multi-MUMs are examined for breakpoints. A breakpoint exists between M_i and M_{i+1} if $M_i.label + 1 \neq M_{i+1}.label$ and both M_i and M_{i+1} are in the forward orientation, or if $M_i.label - 1 \neq M_{i+1}.label$ and both M_i and M_{i+1} are in the reverse complement orientation. A breakpoint also exists if M_i is in a different orientation than M_{i+1} in sequence j, e.g. the sign of $M_i.S_j$ is different than the sign of $M_{i+1}.S_j$. Finally, the multi-MUMs are re-ordered on $M.label$ and the LCBs are then any maximal length subsequence of multi-MUMs $M_i \ldots M_{i+j}$ that does not contain any recorded breakpoints between multi-MUMs.

Bibliography

M. I. Abouelhoda and E. Ohlebusch. CHAINER: Software for comparing genomes. *Proc Int Conf Intell Syst Mol Biol*, 12:1–3, 2004.

M. Achtman, K. Zurth, G. Morelli, G. Torrea, A. Guiyoule, and E. Carniel. *Yersinia pestis*, the cause of plague, is a recently emerged clone of *Yersinia pseudotuberculosis*. *Proc Natl Acad Sci U S A*, 96(24):14043–14048, November 1999.

S. F. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *J Molec Biol*, 215(3):403–410, 1990.

C. Ane and M. Sanderson. Missing the forest for the trees: phylogenetic compression and its implications for inferring complex evolutionary histories. *Syst Biol*, 54 (1):I311–I317, 2005.

T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. *Proc Int Conf Intell Syst Mol Biol*, 3:21–29, 1995.

S. Batzoglou, L. Pachter, J. P. Mesirov, B. Berger, and E. S. Lander. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res*, 10(7):950–8, 2000.

R. G. Beiko, T. J. Harlow, and M. A. Ragan. Highways of gene sharing in prokaryotes. *Proc Natl Acad Sci U S A*, 102(40):14332–14337, October 2005.

O. G. Berg. Selection intensity for codon bias and the effective population size of *Escherichia coli*. *Genetics*, 142(4):1379–1382, April 1996.

M. Blanchette, G. Bourque, and D. Sankoff. Breakpoint Phylogenies. *Genome Inform Ser Workshop Genome Inform*, 8:25–34, 1997.

M. Blanchette, W. J. Kent, C. Riemer, L. Elnitski, F. A. Smit, Arian, K. M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. D. Green, D. Haussler, and W. Miller. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res*, 14(4):708–15, 2004.

F. R. Blattner, G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mayhew, J. Gregor, N. W. Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau, and Y. Shao. The complete genome sequence of *Escherichia coli* K-12. *Science*, 277(5331):1453–74, 1997.

G. Bourque and P. A. Pevzner. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Res*, 12(1):26–36, January 2002.

G. Bourque, P. A. Pevzner, and G. Tesler. Reconstructing the genomic architecture of ancestral mammals: Lessons from human, mouse, and rat genomes. *Genome Res*, 14(4):507–516, April 2004.

L. D. Bowler, Q. Y. Zhang, J. Y. Riou, and B. G. Spratt. Interspecies recombination between the penA genes of *Neisseria meningitidis* and commensal *Neisseria* species during the emergence of penicillin resistance in *N. meningitidis*: natural events and laboratory simulation. *J Bacteriol*, 176(2):333–337, January 1994.

N. Bray and L. Pachter. MAVID multiple alignment server. *Nucleic Acids Res*, 31 (13):3525–6, 2003.

N. Bray, I. Dubchak, and L. Pachter. AVID: A global alignment program. *Genome Res*, 13(1):97–102, 2003.

M. Brudno and B. Morgenstern. Fast and sensitive alignment of large genomic sequences. *Proc IEEE Comput Soc Bioinform Conf*, 1:138–147, 2002.

M. Brudno, B. Do, Chuong, M. Cooper, Gregory, M. F. Kim, E. Davydov, E. D. Green, A. Sidow, and S. Batzoglou. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res*, 13(4):721–31, 2003a.

M. Brudno, S. Malde, A. Poliakov, B. Do, Chuong, O. Couronne, I. Dubchak, and S. Batzoglou. Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19 Suppl 1:I54–I62, 2003b.

P. P. Calabrese, S. Chakravarty, and T. J. Vision. Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics*, 19 Suppl 1:i74–80, 2003.

T. J. Carver, K. M. Rutherford, M. Berriman, M. A. Rajandream, B. G. Barrell, and J. Parkhill. ACT: the Artemis Comparison Tool. *Bioinformatics*, 21(16): 3422–3423, August 2005.

S. L. L. Chen, C.-S. S. Hung, J. Xu, C. S. S. Reigstad, V. Magrini, A. Sabo, D. Blasiar, T. Bieri, R. R. R. Meyer, P. Ozersky, J. R. R. Armstrong, R. S. S. Fulton, J. P. P. Latreille, J. Spieth, T. M. M. Hooton, E. R. R. Mardis, S. J. J. Hultgren, and J. I. I. Gordon. Identification of genes subject to positive selection in uropathogenic strains of *Escherichia coli*: A comparative genomics approach. *Proc Natl Acad Sci U S A*, 103(15):5977–82, April 2006.

F. Chiaromonte, V. B. Yap, and W. Miller. Scoring pairwise genomic sequence alignments. *Pac Symp Biocomput*, pages 115–126, 2002.

K. Choi, P, F. Zeng, and L. Zhang. Good Spaced Seeds For Homology Search. *Bioinformatics*, 20:1053–1059, 2004.

A. Clark, G. Gibson, T. Kaufman, B. McAllister, E. Myers, and P. O'Grady. Proposal for *Drosophila* as a model for comparative genomics. Technical report, National Human Genome Research Institute, June 2003.

M. Csuros and I. Miklos. A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer. In A. Apostolico, C. Guerra, S. Istrail, P. A. Pevzner, and M. S. Waterman, editors, *Proceedings of RECOMB 2006, Tenth Annual International Conference on Research in Computational Molecular Biology*, volume 3909 of *LNBI*, pages 206–220, Berlin, 2006. Springer Verlag.

A. C. E. Darling, B. Mau, F. R. Blattner, and N. T. Perna. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res*, 14 (7):1394–403, 2004a.

A. E. Darling, B. Mau, F. R. Blattner, and N. T. Perna. GRIL: Genome rearrangement and inversion locator. *Bioinformatics*, 20(1):122–124, 2004b.

A. E. Darling, T. J. Treangen, L. Zhang, C. Kuiken, X. Messeguer, and N. T. Perna. Procrastination leads to efficient filtration for local multiple alignment. In B. M. E. Moret, editor, *Lecture Notes in Bioinformatics*, volume 4175, page In press. Springer-Verlag, 2006.

V. Daubin and H. Ochman. Bacterial genomes as new gene homes: the genealogy of ORFans in *E. coli*. *Genome Res*, 14(6):1036–1042, June 2004.

V. Daubin, N. A. Moran, and H. Ochman. Phylogenetics and the cohesion of bacterial genomes. *Science*, 301(5634):829–832, August 2003.

A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg. Alignment of whole genomes. *Nucleic Acids Res*, 27(11):2369–76, 1999.

W. Deng, V. Burland, G. Plunkett, A. Boutin, G. F. Mayhew, P. Liss, N. T. Perna, D. J. Rose, B. Mau, S. Zhou, D. C. Schwartz, J. D. Fetherston, L. E. Lindler, R. R. Brubaker, G. V. Plano, S. C. Straley, K. A. McDonough, M. L. Nilles, J. S. Matson, F. R. Blattner, and R. D. Perry. Genome sequence of *Yersinia pestis* KIM. *J Bacteriol*, 184(16):4601–4611, August 2002.

W. Deng, S.-R. Liou, G. Plunkett, G. F. Mayhew, D. J. Rose, V. Burland, V. Kodoyianni, D. C. Schwartz, and F. R. Blattner. Comparative genomics of

Salmonella enterica serovar Typhi strains Ty2 and CT18. *J Bacteriol*, 185(7): 2330–7, 2003.

C. N. Dewey and L. Pachter. Evolution at the nucleotide level: the problem of multiple whole-genome alignment. *Hum Mol Genet*, 15 Suppl 1, April 2006.

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

A. J. Drummond, S. Y. Ho, M. J. Phillips, and A. Rambaut. Relaxed phylogenetics and dating with confidence. *PLoS Biology*, 4(5):699–710, May 2006.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*, chapter 5, pages 100–133. Cambridge University Press, Cambridge, UK, 1998.

D. E. Dykhuizen and L. Green. Recombination in *Escherichia coli* and the definition of biological species. *J Bacteriol*, 173(22):7257–7268, November 1991.

R. C. Edgar. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5(1):113, 2004.

R. C. Edgar and E. W. Myers. PILER: identification and classification of genomic repeats. *Bioinformatics*, 21 Suppl 1:i152–i158, June 2005.

R. A. Edwards and F. Rohwer. Opinion: Viral metagenomics. *Nature Reviews Microbiology*, 3(6):504–510, June 2005.

E. J. Feil and B. G. Spratt. Recombination and the population structures of bacterial pathogens. *Annu Rev Microbiol*, 55:561–590, 2001.

E. J. Feil, M. C. Maiden, M. Achtman, and B. G. Spratt. The relative contributions of recombination and mutation to the divergence of clones of *Neisseria meningitidis*. *Mol Biol Evol*, 16(11):1496–1502, November 1999.

J. Felsenstein. *Inferring Phylogenies.* Sinauer Associates, Inc, Sunderland, MA USA, 2004.

D. F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol*, 25(4):351–360, 1987.

D. Fischer and D. Eisenberg. Finding families for genomic ORFans. *Bioinformatics*, 15(9):759–762, September 1999.

W. M. Fitch. Homology a personal view on some of the problems. *Trends Genet*, 16(5):227–231, May 2000.

J. Flannick and S. Batzoglou. Using multiple alignments to improve seeded local alignment algorithms. *Nucleic Acids Res*, 33(14):4563–4577, 2005.

R. Fleissner, D. Metzler, and A. von Haeseler. Simultaneous statistical multiple alignment and phylogeny reconstruction. *Syst Biol*, 54(4):548–561, August 2005.

A. Friedrich, T. Hartsch, and B. Averhoff. Natural transformation in mesophilic and thermophilic bacteria: identification and characterization of novel, closely related competence genes in Acinetobacter sp. strain BD413 and Thermus thermophilus HB27. *Appl Environ Microbiol*, 67(7):3140–3148, July 2001.

F. Ge, L. S. Wang, and J. Kim. The cobweb of life revealed by genome-scale estimates of horizontal gene transfer. *PLoS Biol*, 3(10), October 2005.

J. P. Gogarten, W. F. Doolittle, and J. G. Lawrence. Prokaryotic evolution in light of gene transfer. *Mol Biol Evol*, 19(12):2226–2238, December 2002.

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

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

D. S. Guttman and D. E. Dykhuizen. Clonal divergence in *Escherichia coli* as a result of recombination, not mutation. *Science*, 266(5189):1380–1383, November 1994.

B. J. Haas, A. L. Delcher, J. R. Wortman, and S. L. Salzberg. DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics*, 20(18): 3643–3646, December 2004.

J. Hacker and J. B. Kaper. Pathogenicity islands and the evolution of microbes. *Annu Rev Microbiol*, 54:641–679, 2000.

S. Hampson, A. McLysaght, B. Gaut, and P. Baldi. LineUp: statistical detection of chromosomal homology with application to plant comparative genomics. *Genome Res*, 13(5):999–1010, 2003.

S. E. Hampson, B. S. Gaut, and P. Baldi. Statistical detection of chromosomal homology using shared-gene density alone. *Bioinformatics*, 21(8):1339–1348, April 2005.

S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 178–189, 1995.

T. J. Harlow, J. P. Gogarten, and M. A. Ragan. A hybrid clustering approach to recognition of protein families in 114 microbial genomes. *BMC Bioinformatics*, 5, April 2004.

M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *J Mol Evol*, 22(2):160–174, 1985.

T. Hayashi, K. Makino, M. Ohnishi, K. Kurokawa, K. Ishii, K. Yokoyama, C. G. Han, E. Ohtsubo, K. Nakayama, T. Murata, M. Tanaka, T. Tobe, T. Iida, H. Takami, T. Honda, C. Sasakawa, N. Ogasawara, T. Yasunaga, S. Kuhara, T. Shiba, M. Hattori, and H. Shinagawa. Complete genome sequence of enterohemorrhagic *Escherichia coli* O157:H7 and genomic comparison with a laboratory strain K-12. *DNA Res*, 8(1):11–22, 2001.

S. Henz, D. Huson, A. F. Auch, K. Nieselt-Struwe, and S. Schuster. Whole-genome prokaryotic phylogeny. *Bioinformatics*, 21(10):2329–2335, 2005.

M. Hohl, S. Kurtz, and E. Ohlebusch. Efficient multiple genome alignment. *Bioinformatics*, 18 Suppl 1:S312–20, 2002.

M. Holder and P. O. Lewis. Phylogeny estimation: traditional and Bayesian approaches. *Nat Rev Genet*, 4(4):275–84, 2003.

I. Holmes and W. J. Bruno. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, 17(9):803–20, 2001.

W. W. Hsiao, K. Ung, D. Aeschliman, J. Bryan, B. B. Finlay, and F. S. Brinkman. Evidence of a large novel gene pool associated with prokaryotic genomic islands. *PLoS Genetics*, 1(5):e62+, November 2005.

http://r-project.org. The R project for statistical computing software.

J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–5, 2001.

D. Husmeier and G. McGuire. Detecting recombination with MCMC. *Bioinformatics*, 18 Suppl 1, 2002.

D. B. Jaffe, J. Butler, S. Gnerre, E. Mauceli, K. Lindblad-Toh, J. P. Mesirov, M. C. Zody, and E. S. Lander. Whole-genome sequence assembly for mammalian genomes: Arachne 2. *Genome Res*, 13(1):91–96, January 2003.

Q. Jin, Z. Yuan, J. Xu, Y. Wang, Y. Shen, W. Lu, J. Wang, H. Liu, J. Yang, F. Yang, X. Zhang, J. Zhang, G. Yang, H. Wu, D. Qu, J. Dong, L. Sun, Y. Xue, A. Zhao, Y. Gao, J. Zhu, B. Kan, K. Ding, S. Chen, H. Cheng, Z. Yao, B. He, R. Chen, D. Ma, B. Qiang, Y. Wen, Y. Hou, and J. Yu. Genome sequence of *Shigella flexneri* 2a: insights into pathogenicity through comparison with genomes of *Escherichia coli* K12 and O157. *Nucleic Acids Res*, 30(20):4432–41, 2002.

T. Jukes and C. Cantor. *Mammalian Protein Metabolism*, volume 3, chapter Evolution of protein molecules, pages 21–132. Academic Press, New York, 1969.

J. Jurka, V. V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany, and J. Walichiewicz. Repbase Update, a database of eukaryotic repetitive elements. *Cytogenet Genome Res*, 110(1-4):462–467, 2005.

T. Kahveci, V. Ljosa, and A. K. Singh. Speeding up whole-genome alignment by indexing frequency vectors. *Bioinformatics*, 20(13):2122–2134, September 2004.

S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Natl Acad Sci U S A*, 87(6):2264–2268, March 1990.

S. Karlin and V. Brendel. Chance and statistical significance in protein and dna sequence analysis. *Science*, 257(5066):39–49, July 1992.

S. Karlin, P. Bucher, V. Brendel, and S. F. Altschul. Statistical methods and insights for protein and DNA sequences. *Annu Rev Biophys Biophys Chem*, 20: 175–203, 1991.

W. J. Kent. BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–664, April 2002.

W. J. Kent and A. M. Zahler. Conservation, regulation, synteny, and introns in a large-scale C. briggsae-C. elegans genomic alignment. *Genome Res*, 10(8):1115–25, 2000.

L. B. Koski and G. B. Golding. The closest BLAST hit is often not the nearest neighbor. *J Mol Evol*, 52(6):540–542, June 2001.

C. Kuiken, K. Yusim, L. Boykin, and R. Richardson. The Los Alamos hepatitis C sequence database. *Bioinformatics*, 21(3):379–84, 2005.

S. Kurtz, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich. Computation and visualization of degenerate repeats in complete genomes. *Proc Intell Syst Mol Biol*, 8:228–38, 2000.

S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biol*, 5(2), 2004a.

S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biol*, 5(2):R12, 2004b.

B. Larget and D. Simon. Markov chain monte carlo algorithms for the bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, 16:750–759, 1999.

B. Larget, D. L. Simon, and J. Kadane. On a Bayesian approach to phylogenetic inference from animal mitochondrial genome arrangements. *Journal of the Royal Statistical Society*, B 64:681–693, 2002.

B. Larget, D. Simon, J. Kadane, and D. Sweet. A Bayesian Analysis of Metazoan Mitochondrial Genome Arrangements. *Mol Biol Evol*, 2004.

C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, October 1993.

J. G. Lawrence and H. Hendrickson. Lateral gene transfer: when will adolescence end? *Mol Microbiol*, 50(3):739–749, November 2003.

C. Lee, C. Grasso, and M. F. Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3):452–64, 2002.

J. Lefebvre, N. El-Mabrouk, E. Tillier, and D. Sankoff. Detection and validation of single gene inversions. *Bioinformatics*, 19 Suppl 1:I190–I196, 2003.

E. Lerat, V. Daubin, and N. A. Moran. From gene trees to organismal phylogeny in prokaryotes: the case of the gamma-Proteobacteria. *PLoS Biol*, 1(1), October 2003.

L. Li, C. J. Stoeckert, and D. S. Roos. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res*, 13(9):2178–2189, September 2003.

M. Li, B. Ma, and L. Zhang. Superiority and complexity of the spaced seeds. In *SODA*, pages 444–453, 2006.

J. R. Lobry. Asymmetric substitution patterns in the two dna strands of bacteria. *Mol Biol Evol*, 13(5):660–665, May 1996.

C. L. Lu, T. C. Wang, Y. C. Lin, and C. Y. Tang. ROBIN: a tool for genome rearrangement of block-interchanges. *Bioinformatics*, 21(11):2780–2782, June 2005.

A. Lunter, G, I. Miklòs, S. Song, Y, and J. Hein. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J Comput Biol*, 10 (6):869–89, 2003.

G. Lunter, I. Miklòs, A. Drummond, J. L. Jensen, and J. Hein. Bayesian coestimation of phylogeny and sequence alignment. *BMC Bioinformatics*, 6, 2005.

B. Ma, J. Tromp, and M. Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, March 2002a.

B. Ma, J. Tromp, and M. Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–5, 2002b.

M. Margulies, M. Egholm, and 54 other authors. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

W. S. Martins, J. del Cuvillo, W. Cui, and G. R. Gao. Whole genome alignment using a multithreaded parallel implementation. *Symposium on Computer Architecture and High Performance Computing*, pages 1–8, September 10–12 2001.

B. Mau, M. A. Newton, and B. Larget. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. *Biometrics*, 55(1):1–12, March 1999.

B. Mau, A. E. Darling, and N. T. Perna. Identifying evolutionarily conserved segments among multiple divergent and rearranged genomes. In J. Lagergren, editor, *Lecture Notes in Bioinformatics*, volume 3388, pages 72–84. Springer-Verlag, 2004.

B. Mau, J. D. Glasner, A. E. Darling, and N. T. Perna. Genome-wide detection and analysis of homologous recombination among sequenced strains of *Escherichia coli*. *Genome Biology*, 7:R44+, May 2006.

N. H. Maynard Smith. Detecting recombination from gene trees. *Mol Biol Evol*, 15(5):590–599, May 1998.

G. McGuire and F. Wright. TOPAL 2.0: improved detection of mosaic sequences within multiple alignments. *Bioinformatics*, 16(2):130–134, February 2000.

M. McKane and R. Milkman. Transduction, restriction and recombination patterns in *Escherichia coli*. *Genetics*, 139(1):35–43, January 1995.

D. Metzler, R. Fleissner, A. Wakolbinger, and A. von Haeseler. Assessing variability by joint sampling of alignments and mutation rates. *J Mol Evol*, 53(6): 660–669, December 2001.

I. Miklos. MCMC genome rearrangement. *Bioinformatics*, 19 Suppl 2:II130–II137, 2003.

I. Miklòs, G. Lunter, and I. Holmes. A Long Indel model for evolutionary sequence alignment. *Mol Biol Evol.*, 21(3):529–40, 2004.

R. Milkman. Recombination and population structure in *Escherichia coli*. *Genetics*, 146(3):745–750, July 1997.

V. N. Minin, K. S. Dorman, F. Fang, and M. A. Suchard. Dual multiple change-point model leads to more accurate recombination detection. *Bioinformatics*, 21 (13):3034–3042, July 2005.

A. Mira, H. Ochman, and N. A. Moran. Deletional bias and the evolution of bacterial genomes. *Trends Genet*, 17(10):589–596, October 2001.

G. Mitchison and R. Durbin. Tree-based maximal likelihood substitution matrices and hidden Markov models. *J Mol Evol*, 41:1139–1151, 1995.

G. J. Mitchison. A probabilistic treatment of phylogeny and sequence alignment. *J Mol Evol*, 49(1):11–22., 1999.

N. Nagarajan, N. Jones, and U. Keich. Computing the P-value of the information content from an alignment of multiple sequences. *Bioinformatics*, 21 Suppl 1, June 2005.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53, 1970.

R. Nielsen, editor. *Statistical methods in molecular evolution*, chapter 14, pages 375–406. Springer-Verlag, New York, NY, 2005.

L. Noé and G. Kucherov. Improved hit criteria for DNA local alignment. *BMC Bioinformatics*, 5, October 2004.

C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302(1):205–17, 2000.

A. S. Novozhilov, G. P. Karev, and E. V. Koonin. Mathematical modeling of evolution of horizontally transferred genes. *Mol Biol Evol*, 22(8):1721–1732, August 2005.

H. Ochman and A. C. Wilson. Evolution in bacteria: evidence for a universal substitution rate in cellular genomes. *J Mol Evol*, 26(1-2):74–86, 1987.

H. Ochman, E. Lerat, and V. Daubin. Examining bacterial species under the specter of gene transfer and exchange. *Proc Natl Acad Sci U S A*, 102 Suppl 1: 6595–6599, May 2005.

M. V. Omelchenko, K. S. Makarova, Y. I. Wolf, I. B. Rogozin, and E. V. Koonin. Evolution of mosaic operons by horizontal gene transfer and gene displacement in situ. *Genome Biol*, 4(9), 2003.

I. Ovcharenko, G. G. Loots, B. M. Giardine, M. Hou, J. Ma, R. C. Hardison, L. Stubbs, and W. Miller. Mulan: Multiple-sequence local alignment and visualization for studying function and evolution. *Genome Res*, 15(1):184–94, 2005.

J. Parkhill, G. Dougan, K. D. James, R. Thomson, N, D. Pickard, J. Wain, C. Churcher, K. L. Mungall, S. D. Bentley, M. T. Holden, M. Sebaihia, S. Baker, D. Basham, K. Brooks, T. Chillingworth, P. Connerton, A. Cronin, P. Davis, R. M. Davies, L. Dowd, N. White, J. Farrar, T. Feltwell, N. Hamlin, A. Haque, T. T. Hien, S. Holroyd, K. Jagels, A. Krogh, T. S. Larsen, S. Leather, S. Moule, P. O'Gaora, C. Parry, M. Quail, K. Rutherford, M. Simmonds, J. Skelton, K. Stevens, S. Whitehead, and B. G. Barrell. Complete genome sequence of a multiple drug resistant *Salmonella enterica* serovar Typhi CT18. *Nature*, 413 (6858):848–52, 2001.

N. T. Perna, G. Plunkett, V. Burland, B. Mau, J. D. Glasner, D. J. Rose, G. F. Mayhew, P. S. Evans, J. Gregor, H. A. Kirkpatrick, G. Posfai, J. Hackett, S. Klink, A. Boutin, Y. Shao, L. Miller, E. J. Grotbeck, N. W. Davis, A. Lim, E. T. Dimalanta, K. D. Potamousis, J. Apodaca, T. S. Anantharaman, J. Lin, G. Yen, D. C. Schwartz, R. A. Welch, and F. R. Blattner. Genome sequence of enterohaemorrhagic *Escherichia coli* O157:H7. *Nature*, 409(6819):529–33, 2001.

P. Pevzner and G. Tesler. Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res*, 13(1):37–45, 2003a.

P. Pevzner and G. Tesler. Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. *Proc Natl Acad Sci U S A*, 100(13): 7672–7, 2003b.

P. Pevzner, H. Tang, and G. Tesler. De novo repeat classification and fragment assembly. *Genome Research*, 14(9):1786–96, 2004.

D. Posada and K. A. Crandall. Evaluation of methods for detecting recombination from DNA sequences: Computer simulations. *Proc Natl Acad Sci U S A*, 98(24): 13757–13762, November 2001.

D. Posada, K. A. Crandall, and E. C. Holmes. Recombination in evolutionary genomics. *Annu Rev Genet*, 36:75–97, 2002.

A. Prakash and M. Tompa. Statistics of local multiple alignments. *Bioinformatics*, 21(Suppl 1):i344–i350, 2005.

M. N. Price, A. P. Arkin, and E. J. Alm. The life-cycle of operons. *PLoS Genet*, 2(6), June 2006.

B. Qian and R. A. Goldstein. Detecting distant homologs using phylogenetic tree-based HMMs. *Proteins*, 52(3):446–53, 2003.

W. G. Qiu, S. E. Schutzer, J. F. Bruno, O. Attie, Y. Xu, J. J. Dunn, C. M. Fraser, S. R. Casjens, and B. J. Luft. Genetic exchange and plasmid transfers in Borrelia burgdorferi sensu stricto revealed by three-way genome comparisons and multilocus sequence typing. *Proc Natl Acad Sci U S A*, 101(39):14150–14155, September 2004.

M. A. Ragan and R. L. Charlebois. Distributional profiles of homologous open reading frames among bacterial phyla: implications for vertical and lateral transmission. *Int J Syst Evol Microbiol*, 52(Pt 3):777–787, May 2002.

A. Rambaut and N. C. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput Appl Biosci*, 13(3):235–8, 1997.

B. Raphael, D. Zhi, H. Tang, and P. Pevzner. A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res*, 14(11): 2336–46, 2004.

J. Raymond, O. Zhaxybayeva, J. P. Gogarten, S. Y. Gerdes, and R. E. Blankenship. Whole-genome analysis of photosynthetic prokaryotes. *Science*, 298(5598):1616–1620, November 2002.

B. Redelings and M. Suchard. Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–418, June 2005.

P. REEVES. Role of Hfr mutants in F-plus x F-minus crosses in *Escherichia coli* K12. *Nature*, 185:265–266, January 1960.

S. D. Reid, C. J. Herbelin, A. C. Bumbaugh, R. K. Selander, and T. S. Whittam. Parallel evolution of virulence in pathogenic *Escherichia coli*. *Nature*, 406(6791): 64–67, July 2000.

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

F. Ronquist and J. P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–4, 2003.

N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–25, 1987.

M. Sammeth and J. Heringa. Global multiple-sequence alignment with repeats. *Proteins*, April 2006.

M. Sammeth, T. Weniger, D. Harmsen, and J. Stoye. Alignment of Tandem Repeats with Excision, Duplication, Substitution and Indels (EDSI). In *Proceedings of the 5th International Workshop on Algorithms in Bioinformatics, WABI 2005*, volume 3692 of *LNBI*, pages 276–290, Berlin, 2005. Springer Verlag.

S. Sawyer. Statistical tests for detecting gene conversion. *Mol Biol Evol*, 6(5): 526–538, September 1989.

S. Scherer, M. S. McPeek, and T. P. Speed. Atypical regions in large genomic DNA sequences. *Proc Natl Acad Sci U S A*, 91(15):7134–7138, July 1994.

S. Schwartz, W. J. Kent, A. Smit, Z. Zhang, R. Baertsch, R. C. Hardison, D. Haussler, and W. Miller. Human-mouse alignments with BLASTZ. *Genome Res*, 13 (1):103–7, 2003.

M. H. Serres and M. Riley. MultiFun, a multifunctional classification scheme for *Escherichia coli* K-12 gene products. *Microb Comp Genomics*, 5(4):205–222, 2000.

J. Shendure, G. J. Porreca, N. B. Reppas, X. Lin, J. P. Mccutcheon, A. M. Rosenbaum, M. D. Wang, K. Zhang, R. D. Mitra, and G. M. Church. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science*, 309(5741):1728–1732, September 2005.

R. Siddharthan, E. D. Siggia, and E. van Nimwegen. PhyloGibbs: A Gibbs aampling motif finder that incorporates phylogeny. *PLoS Comput Biol*, 1(7), December 2005.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7, 1981.

B. G. Spratt, W. P. Hanage, and E. J. Feil. The relative contributions of recombination and point mutation to the diversification of bacterial clones. *Curr Opin Microbiol*, 4(5):602–606, October 2001.

J. C. Stephens. Statistical methods of DNA sequence analysis: detection of intragenic recombination or gene conversion. *Mol Biol Evol*, 2(6):539–556, November 1985.

A. Stoltzfus, J. F. Leslie, and R. Milkman. Molecular evolution of the *Escherichia coli* chromosome. I. Analysis of structure and natural variation in a previously uncharacterized region between trp and tonB. *Genetics*, 120(2):345–358, October 1988.

M. A. Suchard and B. D. Redelings. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics*, May 2006.

S. Suerbaum, J. M. Smith, K. Bapumia, G. Morelli, N. H. Smith, E. Kunstmann, I. Dyrek, and M. Achtman. Free recombination within *Helicobacter pylori*. *Proc Natl Acad Sci U S A*, 95(21):12619–12624, October 1998.

M. B. B. Sullivan, D. Lindell, J. A. A. Lee, L. R. R. Thompson, J. P. P. Bielawski, and S. W. W. Chisholm. Prevalence and evolution of core photosystem II genes in marine Cyanobacterial viruses and their hosts. *PLoS Biol*, 4(8), July 2006.

Y. Sun and J. Buhler. Designing multiple simultaneous seeds for DNA similarity search. *J Comput Biol*, 12(6):847–861, 2005.

R. Szklarczyk and J. Heringa. AuberGene–a sensitive genome alignment tool. *Bioinformatics*, 22(12):1431–1436, June 2006.

R. Szklarczyk and J. Heringa. Tracking repeats using significance and transitivity. . *Bioinformatics*, Suppl 1:I311–I317, 2004.

J. Tang and B. M. Moret. Scaling up accurate phylogenetic reconstruction from gene-order data. *Bioinformatics*, 19 Suppl 1:i305–i312, 2003.

H. Tettelin, V. Masignani, M. J. Cieslewicz, C. Donati, D. Medini, N. L. Ward, S. V. Angiuoli, J. Crabtree, A. L. Jones, A. S. Durkin, R. T. Deboy, T. M. Davidsen, M. Mora, M. Scarselli, I. Margarit y Ros, J. D. Peterson, C. R. Hauser, J. P. Sundaram, W. C. Nelson, R. Madupu, L. M. Brinkac, R. J. Dodson, M. J. Rosovitz, S. A. Sullivan, S. C. Daugherty, D. H. Haft, J. Selengut, M. L. Gwinn, L. Zhou, N. Zafar, H. Khouri, D. Radune, G. Dimitrov, K. Watkins, K. J. O'Connor, S. Smith, T. R. Utterback, O. White, C. E. Rubens, G. Grandi, L. C. Madoff, D. L. Kasper, J. L. Telford, M. R. Wessels, R. Rappuoli, and C. M. Fraser. Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". *Proc Natl Acad Sci U S A*, 102(39):13950–13955, September 2005.

J. W. Thomas, J. W. Touchman, R. W. Blakesley, G. G. Bouffard, S. M. Beckstrom-Sternberg, E. H. Margulies, M. Blanchette, A. C. Siepel, P. J. Thomas, J. C. McDowell, B. Maskeri, N. F. Hansen, M. S. Schwartz, R. J. Weber, W. J. Kent, D. Karolchik, T. C. Bruen, R. Bevan, D. J. Cutler, S. Schwartz, L. Elnitski,

J. R. Idol, A. B. Prasad, S. Q. Lee-Lin, V. V. Maduro, T. J. Summers, M. E. Portnoy, N. L. Dietrich, N. Akhter, K. Ayele, B. Benjamin, K. Cariaga, C. P. Brinkley, S. Y. Brooks, S. Granite, X. Guan, J. Gupta, P. Haghighi, S. L. Ho, M. C. Huang, E. Karlins, P. L. Laric, R. Legaspi, M. J. Lim, Q. L. Maduro, C. A. Masiello, S. D. Mastrian, J. C. McCloskey, R. Pearson, S. Stantripop, E. E. Tiongson, J. T. Tran, C. Tsurgeon, J. L. Vogt, M. A. Walker, K. D. Wetherby, L. S. Wiggins, A. C. Young, L. H. Zhang, K. Osoegawa, B. Zhu, B. Zhao, C. L. Shu, P. J. De Jong, C. E. Lawrence, A. F. Smit, A. Chakravarti, D. Haussler, P. Green, W. Miller, and E. D. Green. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, 424(6950):788–793, August 2003.

J. D. Thompson, D. G. Higgins, and T. 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 Res*, 22(22): 4673–80, 1994.

J. D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res*, 27(13):2682–90, 1999.

J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J Mol Evol*, 33(2):114–24, 1991.

J. L. Thorne, H. Kishino, and J. Felsenstein. Inching toward reality: an improved likelihood model of sequence evolution. *J Mol Evol*, 34(1):3–16, 1992.

T. Treangen and X. Messeguer. M-GCAT: Multiple genome comparison and alignment tool. *Submitted*, 2006.

S. G. Tringe, C. von Mering, A. Kobayashi, A. A. Salamov, K. Chen, H. W. Chang, M. Podar, J. M. Short, E. J. Mathur, J. C. Detter, P. Bork, P. Hugenholtz, and E. M. Rubin. Comparative metagenomics of microbial communities. *Science*, 308 (5721):554–557, April 2005.

A. Ureta-Vidal, L. Ettwiller, and E. Birney. Comparative genomics: Genome-wide analysis in metazoan eukaryotes. *Nat Rev Genet*, 4(4):251–62, 2003.

C. C. Venter, K. Remington, J. F. Heidelberg, A. L. Halpern, D. Rusch, J. A. Eisen, D. Wu, I. Paulsen, K. E. Nelson, W. Nelson, D. E. Fouts, S. Levy, A. H. Knap, M. W. Lomas, K. Nealson, O. White, J. Peterson, J. Hoffman, R. Parsons, H. Baden-Tillson, C. Pfannkoch, Y.-H. Rogers, and H. O. Smith. Environmental genome shotgun sequencing of the sargasso sea. *Science*, 304(5667):66–74, April 2004.

I. Wallace, O. O'sullivan, and D. Higgins. Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics*, 21(8):1408–14, 2005.

J. Wei, M. B. Goldberg, V. Burland, M. M. Venkatesan, W. Deng, G. Fournier, G. F. Mayhew, G. Plunkett, D. J. Rose, A. Darling, B. Mau, N. T. Perna, S. M. Payne, L. J. Runyen-Janecky, S. Zhou, D. C. Schwartz, and F. R. Blattner. Complete genome sequence and comparative genomics of *Shigella flexneri* serotype 2a strain 2457T. *Infect Immun*, 71(5):2775–86., 2003.

R. A. Welch, V. Burland, G. Plunkett, P. Redford, P. Roesch, D. Rasko, L. Buckles, E, S.-R. Liou, A. Boutin, J. Hackett, D. Stroud, F. Mayhew, G, J. Rose, D, S. Zhou, C. Schwartz, D, T. Perna, N, L. T. Mobley, H, S. Donnenberg, M, and R. Blattner, F. Extensive mosaic structure revealed by the complete genome sequence of uropathogenic *Escherichia coli*. *Proc Natl Acad Sci U S A*, 99(26): 17020–4, 2002.

J. E. Wertz, C. Goldstone, D. M. Gordon, and M. A. Riley. A molecular phylogeny of enteric bacteria and implications for a bacterial species concept. *J Evol Biol*, 16(6):1236–1248, November 2003.

M. Worobey. A novel approach to detecting and measuring recombination: new insights into evolution in viruses, bacteria, and mitochondria. *Mol Biol Evol*, 18 (8):1425–1434, August 2001.

J. Xu, D. G. Brown, M. Li, and B. Ma. Optimizing Multiple Spaced Seeds for Homology Search. In *Lecture Notes in Computer Science*, volume 3109, pages 47–58. Springer, 2004.

S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16): 3340–3346, August 2005.

T. Zeppenfeld, C. Larisch, J. W. Lengeler, and K. Jahreis. Glucose transporter mutants of *Escherichia coli* K-12 with changes in substrate recognition of IICB(Glc) and induction behavior of the ptsG gene. *J Bacteriol*, 182(16):4443–4452, August 2000.

Y. Zhang and M. S. Waterman. An Eulerian path approach to local multiple alignment for DNA sequences. *Proc Natl Acad Sci U S A*, 102(5):1285ÂŰ1290, 2005.

Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A greedy algorithm for aligning DNA sequences. *J Comput Biol*, 7(1-2):203–214, 2000.

O. Zhaxybayeva, L. Hamel, J. Raymond, and J. P. Gogarten. Visualization of the phylogenetic content of five genomes using dekapentagonal maps. *Genome Biol*, 5 (3), 2004.

E. Zuckerkandl and L. Pauling. Molecules as documents of evolutionary history. *J Theor Biol*, 8(2):357–366, March 1965.