Reversing Gene Erosion—Reconstructing Ancestral Bacterial Genomes from Gene-Content and Order Data

Joel V. Earnest-DeYoung[1], Emmanuelle Lerat[2], and Bernard M.E. Moret[1,3]

Abstract

In the last few years, it has become routine to use gene-order data to reconstruct phylogenies, both in terms of edge distances (parsimonious sequences of operations that transform one end point of the edge into the other) and in terms of genomes at internal nodes, on small, duplication-free genomes. Current gene-order methods break down, though, when the genomes contain more than a few hundred genes, possess high copy numbers of duplicated genes, or create edge lengths in the tree of over one hundred operations. We have constructed a series of heuristics that allow us to overcome these obstacles and reconstruct edges distances and genomes at internal nodes for groups of larger, more complex genomes. We present results from the analysis of a group of thirteen modern γ-proteobacteria, as well as from simulated datasets.

1 Introduction

Although phylogeny, the evolutionary relationships between related species or taxa, is a fundamental building block in much of biology, it has been surprisingly difficult to automate the process of inferring these evolutionary relationships from modern data (usually molecular sequence data). These relationships include both the evolutionary distances within a group of species and the genetic form of their common ancestors.

In the last decade, a new form of molecular data has become available, gene-content and gene-order data; this new data has proved useful in shedding light on these relationships [5, 23, 7, 12]. The order and the orientation in which genes lie on a chromosome changes very slowly, in evolutionary terms, and thus provides a rich source of information for reconstructing phylogenies.

Until very recently, a major bottleneck has been that algorithms using this type of data required that all genomes have identical gene content with no duplication. Because most sets of genomes found in nature do not meet these requirements, researchers either were limited to very simple genomes (such as chloroplast organelles) or had to reduce their data by deleting all genes not present in every genome and then delete all "copies" of each gene but one (e.g., using the *exemplar* strategy [19]); the former was frustrating to biologists wanting to study more complex organisms, while the latter resulted in data loss and consequent loss of accuracy in reconstruction [22].

Our group recently developed a method to compute the distance between two nearly arbitrary genomes [11] and another to reconstruct phylogenies based on gene-content and gene-order in the

[1]Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, USA, joeled,moret@cs.unm.edu

[2]Department of Ecology and Evolutionary Biology, University of Arizona, Tucson, AZ 85721, USA, lerat@email.arizona.edu

[3]to whom all correspondence should be addressed

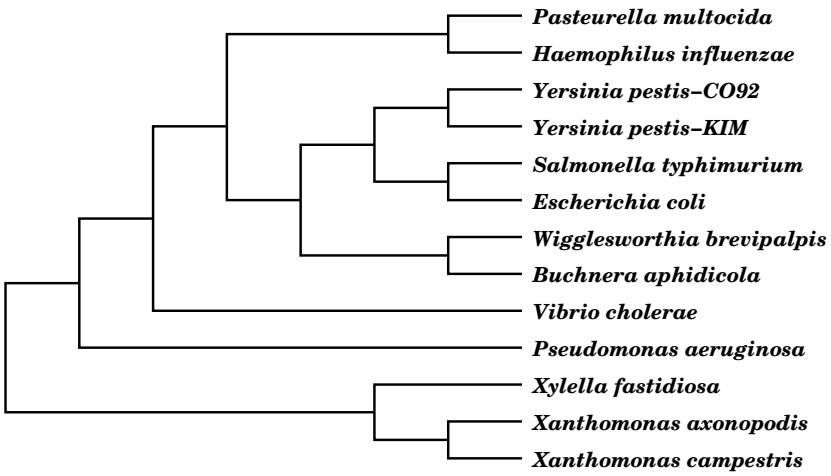

Figure 1: The 13 gamma-proteobacteria and their reference phylogeny. Construction of gene families and the tree is described in more detail in [10]. Gene orders were constructed using whole-genome sequences. Protein-coding sequences were divided into gene families, and only families present in at least three genomes were retained.

presence of mildly unequal gene content [22]. In this paper, we bring together several of these algorithms and heuristics in framework that enables us to reconstruct the gene orders of the common ancestors of the 13 modern bacteria shown in Figure 1 (from [10]). This is an ancient group of bacteria, at least 500 millions years old [4]. It is extremely diverse, including endosymbiotic, commensal, and pathogenic species, and most of the species are medically or economically important. The evolutionary history of the group is also quite complex, including high levels of horizontal gene transfer [9, 18, 20] and in the case of *Buchnera aphidicola* and *Wigglesworthia brevipalpis* massive levels of gene loss. All of this makes reconstruction of these bacteria's evolutionary history both interesting and challenging.

The rest of this paper is organized as follows. Section 2 discusses the problem in some detail. Section 3 summarizes prior work on phylogenetic reconstruction from gene-content and gene-order data. Section 4 presents our framework for tackling the problem of ancestral genome reconstruction given a reference phylogeny; it is itself divided into three subsections, one on each of our three main tools: median-finding, content determination, and gene clustering. Section 5 discusses our approach to the testing of our framework: given that we have only one dataset and that ancestral genomes for that dataset are entirely unknown, our testing was of necessity based on simulations. Section 6 presents the results of this testing. We conclude with some remarks about the significance of our work and the problems remaining to be faced.

2 The Problem

Our specific problem is the following:

- Given the gene orders of a group of genomes and given a rooted tree with these genomes at

the leaves, find gene orders for the internal nodes of the tree that minimize the sum of all edge lengths in the tree.

The length of an edge is defined as the minimum number of operations (from a defined set of permissible operations) needed to transform the genome at one end of the edge into the genome at the other end. The genomes have no restriction on content nor on the number of duplicate copies of individual genes. The permissible operations in our case are inversions, insertions (and duplications), and deletions; all operations are given the same cost in computing edge lengths. That we are restricting rearrangements to inversions only comes from past findings by our groups that the inversion phylogeny is robust even when other rearrangements, such as transpositions, were used in creating the data [15]. Our assignment of unit costs to all operations simply reflect insufficient biological knowledge about the relative frequency of these operations.

In our setting, one insertion may add an arbitrary number of genes to a single location and one deletion may remove a contiguous run of genes from a single location, a convention consistent with biological reality. Gene duplications are treated as specialized insertions that only insert repeats. Finally, on each edge a gene can either be inserted or deleted, but not both; the same holds for multiple copies of the same gene. Allowing deletion and insertion of the same genes on the same edge would lead to biologically ridiculous results such as deleting the entire source genome and then inserting the entire target genome, in just two operations.

Finding internal labels that minimize edge distances over the tree has been addressed by our group in prior work—this is the main optimization performed by our software suite GRAPPA [1]. However, even the most recent version of GRAPPA [22] is limited to relatively small genomes (typically of organellar size, with fewer than 200 genes), with modestly unequal content and just a few duplications, if any. In stark contrast, the bacterial genomes in our dataset contain 3,430 different genes and range in size from 540 to 2,987 genes, with seven containing over 2,300 genes; moreover, these genomes contain a large number of duplications, ranging from 5% to 45% of the genome. Thus, in our model, most pairwise genomic distances are very large: a simple pairwise comparison along the tree of Figure 1 indicates that some edges of the tree must have lengths of at least 300, lengths that are at least an order of magnitude larger than any found in prior uses of GRAPPA. The large genome size, vastly unequal gene content, large number of duplications, and large edge lengths all combine to make this data set orders of magnitude more difficult to analyze than previously analyzed genome sets.

3 Prior Work

A recent review of the current work in phylogenetic reconstruction based on gene content and gene order appears in [16]. Here we simply review the main points relevant to our work.

A heuristic used for tree-labeling in the GRAPPA software package [21] is to initialize internal labels of the tree by some method. The number of each internal node is pushed on a queue. Then each number is iteratively popped off the queue, and if a new label can be found that reduces the distance to the node's three neighbors, the existing label is replaced with the improved label and the numbers of the node's neighbors are pushed onto the queue. Label replacement over the tree stops when the queue is empty.

GRAPPA computes a new label for a node by finding the *median* of its neighbors. An optimal median of three genomes is defined as a fourth genome for which the sum of the number of operations needed to convert it into each of the three genomes is minimized. GRAPPA uses an algorithm to find optimal inversion medians which runs in worst-case exponential time but tends to finish quickly when the edge lengths are small, on the order of 10 to 40 operations per edge [14, 22].

One approach that has successfully increased the speed of equal-content gene-order data has been to treat groups of genes which occur in the same order and orientation in all genomes as a single genetic unit. This condensation of identical clusters of genes avoids wasteful computation and does not change the final result of most analyses. This approach is also used in GRAPPA [13].

A method to find the distance between two genomes with arbitrary gene content was recently developed [11]. The method employs a *duplication-renaming* heuristic that matches multiple copies of genes between genomes and renames each pair and each unmatched copy to a new, unused gene number. This allows arbitrary genomes to converted into duplication-free genomes. A secondary result of [11] is that, given two genomes with unequal gene content and no duplications, any optimal sequence of inversions, deletions, and insertions to convert one genome into the other can be rearranged to contain first insertions, then inversions, and finally deletions—a type of *normal form* for the edit sequence. Deletions here are genes unique to the source genome, while insertions are genes found only in the target genome. Using the duplication-free genomes produced by the duplication-renaming method of [11], an optimal inversion sequence is calculated using a method that runs in time quadratic in the size of the consensus genomes [2, 3]. The number of deletions is calculated by counting the number of Hannenhalli-Pevzner cycles that contain deletions, as described in [6]. Finally, the number of insertions is estimated by calculating all possible positions in the source genome to which the inversion sequence could move insertions, then choosing the final position for each insertion that minimizes the number of groups of inserted genes.

In some genomes, especially bacterial or bacteria-derived genomes, genes with similar function are often located together on one strand of a chromosome; these functional units are called *operons*. In bacteria, at least, while the order of genes in an operon may change, the gene content of the operon is much less likely to do so [17]. In gene-order data, an operon thus appears as a cluster of gene numbers, all with the same sign. The cluster will have the same gene content across genomes, but its genes may be in different orders. A *cluster-finding* algorithm has been developed that can identify these operon-like clusters of genes within equal-content genomes in linear time [8].

In a recent paper which carries out a similar reconstruction of gene orders for poxviruses [12], the gene content of internal nodes was decided by assuming that the phylogenetic tree contained a single point of origin for each gene family in the modern genomes. That point of origin was assigned to the internal node which minimized the number of loss events necessary to achieve the gene content of the leaf genomes.

4 Designing a Algorithmic Framework

We have brought together algorithms and heuristics from a variety of different sources in order to tackle the general problem of finding internal gene-order labels for genomes at the level of complexity that we see in the gamma-proteobacteria. We use condensation of gene clusters in

order to reduce the size of the genomes; we devised a procedure similar in spirit to that used by McLysaght *et al.* [12] to predetermine the gene content of every internal node; and we developed a new heuristic to compute the median of three very different genomes.

4.1 Medians

At the top-level, we use the queue-based tree-labeling heuristic described in Section 3. Since leaves contain the only labels guaranteed to be correct, we update last the nodes with the longest paths to their leaf descendants, as shown in Figure 2.

The heart of the top-level heuristic is the task of repeatedly computing the median of three genomes. Exact median-finding algorithms are limited to relatively small genomes, small edge lengths in the tree, and few changes in content—and none of these properties holds in our problem. We have therefore pursued a simple heuristic inspired by geometry. To find the geometric median of three points in a plane, we can first find the point halfway between two of the three points; the median is then one third of the way between the halfway point and the third original point. In a similar way, we find an approximation for the genomic median by generating a sequence of operations, or sorting sequence, that converts one of the three genomes into another one. Then we choose an intermediate genome partway along this sorting sequence and generate a new sorting sequence from the intermediate genome to the third genome. Finally, we choose as the median a genome one third of the way along this second sorting sequence.

We have extended the method used by Marron *et al*. [11] so that we can now enumerate all possible positions, orientations, and orderings of genes after each operation. Basically, deleted genes at the endpoint of an inversion are moved to the other endpoint if doing so avoids "trapping" the deleted genes between two consensus genes that are adjacent in the target genome. Inserted genes are moved so as to remain adjacent to one of the two consensus genes between which they lay in the target genome. By this extension, we are able to generate the genomes produced by "running" a portion of the sorting sequence. These intermediate genomes can then be used for the geometry-inspired median heuristic just described. This heuristic allows a median to be computed

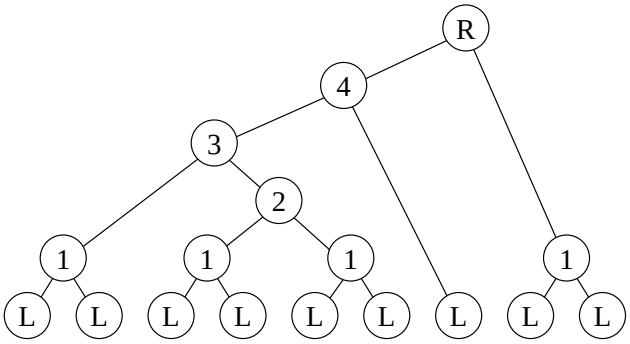

Figure 2: Ordering of internal nodes by the length of their longest path to their leaf descendents (i.e., subtree height). Each internal node is labeled with its subtree height. Nodes with lower subtree heights are updated before those with greather height. Leaves, marked with an 'L', do not need to be updated. No label is generated for the root.

in polynomial time.

By handling inserted genes in the way just described, we overestimate the edit distance, which Marron *et al*. showed at most doubles the number of operations [11]. When their original edit distance method calculates all possible positions in the source genome to which the inversion sequence could move insertions and then chooses the final position for each insertion that minimizes the number of groups of inserted genes, the method may underestimate the minimum edit distance, because the resulting grouping of inserted genes often requires inversions simultaneously to join inserted genes and to split deleted genes, which is not possible. We compared pairwise distances produced by our overestimation and their underestimation to get an upper bound of the error introduced by the overestimation. The average and maximum differences between the overestimate and underestimate were 11.3% and 24.1%, respectively.

4.2 Gene Content

We predetermine the gene content of every internal node of the tree before computing any median; moreover, once the gene content of an internal node is decided, we never change it thereafter. Since the tree is rooted, we know the direction in which time flows on each tree edge; thus, since deletions are far more likely than insertions, we are able to take a simple approach. The number of copies of each gene is considered over the entire tree and is decided independently of all other genes. The number of copies of a gene g at the internal node i is set to the maximum number of copies of g found in any of the leaves in i's subtree if: (i) there are leaves both inside and outside i's subtree which contain at least one copy of g; or (ii) there are leaves containing at least one copy of g in each half of i's subtree. Otherwise the number of copies of gene g in node i is set to zero.

This value can be calculated in $O(NG)$ time, where N is the number of nodes in the tree and G is the number of distinct genes in all the leaves, as follows. First, for each node in the tree, we determine the maximum number of copies of each gene from among the leaves of that node's subtree, using a single depth-first traversal. Next, we perform a second depth-first traversal to set the actual number of copies of each gene at each internal node. If either of the root's children returns a value of zero, then we set the root's actual number to zero as well. For each internal node other than the root, if its parent's actual number of copies is zero and at least one of its two children's subtree maximums is zero, then we set the number of copies for the gene to zero. Otherwise we set the number of copies to the node's subtree maximum for the gene. This approach is similar to that taken in the study of the gene order evolution of poxviruses [12].

A consequence of this approach to determining gene content is that internal nodes possess at least as many copies of a gene as the majority consensus of their neighbors' gene contents. An internal node will always possess a copy of a gene if two or more of its neighbors possess the gene copy. (We consider the two children of the root to be neighbors.) In addition, if the median is the nearest common ancestor of all genomes possessing the gene, it may well have more copies of the gene than its parent and one of its children, as in the case of the black node in Figure 3. The gene content of intermediate genomes along sorting sequences (as used in the determination of medians) will be a union of the gene contents of each of the starting genomes, because the sorting sequence of operations that we use always involves first insertions, then inversions, and finally deletions. Therefore, when calculating medians from sorting sequences, there are three cases in which the number

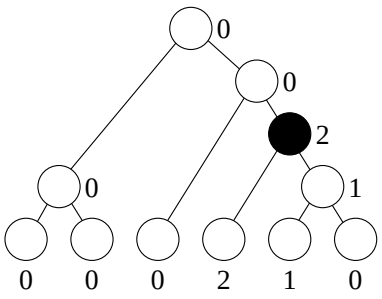

Figure 3: The number next to each node is the number of copies of a particular gene in that node. The black node has two copies, for example.

of copies of a gene are not the same between the intermediate genome, the median genome, and the median's parent, as shown in Figure 4. In the first case (Fig. 4a), the intermediate genome has the same number of copies as the median, but fewer than the parent. An example is the black node's right child in Figure 3. Here each copy in the parent that is not match by the duplication-renaming algorithm to a copy in the intermediate genome will be excluded from the median genome. The second case (Fig. 4b) only arises when the median genome is the nearest common ancestor of all genomes containing the gene in question, as with the black node in Figure 3. Here, genomes along the intermediate sequence have the same number of copies as the median and the parent of the median contains zero copies of the gene. This case is easy to handle, since a genome generated by running part of the sorting sequence from the intermediate genome to the parent will contain the same number of copies as the median. Finally, the situation in Figure 4c can only arise when the right child of the median is the nearest common ancestor of all genomes containing the gene. The parent of the black node in Figure 3 fits this case. This case is also trivial to deal with, since all copies of the gene in the intermediate genome can be simply discarded. The genomes in the sorting sequence from the intermediate genome to the parent will then automatically contain zero copies of the gene.

Biologically, this process of finding which duplicates to include in the median corresponds to

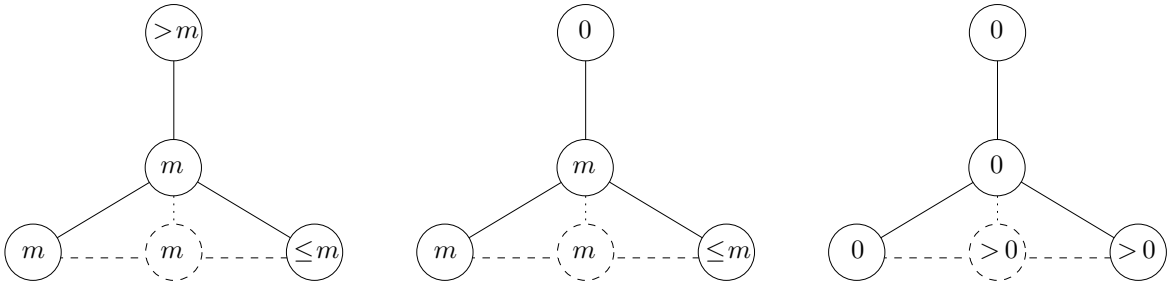

Figure 4: Three cases where the median genome and its neighbors have different numbers of copies of a gene. Each node is labeled with its number of copies. Solid lines are tree edges. A dashed line between two siblings represents the sorting sequence for that pair and the node drawn halfway along is a median of the two siblings. A dotted line represents the first third of the sorting sequence from the intermediate genome to the parent, stopping at the median.

matching orthologous duplicates of each gene between genomes and to discard unmatched paralogous duplicates. Since the original nucleotide sequences are abstracted away before the analysis begins, this ortholog matching is decided entirely on the basis of which other genes are located next to the different homologs. Fortunately, orthologs and paralogs that can be distinguished by a nucleotide-based analysis are assigned different gene numbers before our analysis begins. Therefore, our method represents a reasonable way to integrate both nucleotide and gene-order data in differentiating orthologous and paralogous homologs of genes.

4.3 Cluster Condensation

To use computational methods to extract information from larger and more complex biological data sets, we need fast algorithms with fast implementations. Faster processing generally means that a more thorough analysis can be performed and thus that results of higher quality can be obtained. In our case, the limiting factor is the size of the genomes (their number is also an issue, but a much smaller one). We thus developed a technique to identify and condense gene clusters in unequal genomes in order to reduce the size of the genomes.

Our approach is similar to the one used in equal-content genomes, but is more general. The condensation technique used in GRAPPA only condenses identical subsequences of genes—that is, the genes appear in exactly the same order in all genomes under consideration. Our method allows the condensation of clusters that can have internal reordering of genes (as long as they stay on the same strand) and also handles the difficult cases that arise out of unequal gene content (such as an insertion in the middle of a cluster).

To identify clusters, we first use the duplication-renaming technique of Marron *et al*. to create duplication-free genomes. After renaming, we remove any genes are not present in all of the genomes under examination. This step creates a group of genomes with equal gene content. We then use the cluster-finding algorithm of Heber and Stoye [8] to find equivalent clusters of genes within the equal-content genomes. Once clusters are identified, each one is condensed out of the original genomes and replaced with a single marker (as if it were a single gene).

In a set of genomes with unequal gene content, there can be genes inside a cluster that are not present in the corresponding equal-content genomes. We deal with these genes in two ways. If every occurrence of that gene is located inside the cluster in each of the genomes that possesses the gene, then the gene is condensed along with the rest of the cluster. Otherwise, the extra gene is moved to one side of the cluster, and the cluster is then condensed. When a median genome is computed, a median for each cluster is also computed, and each cluster's marker in the median genome is eventually replaced with the cluster's median. At this point, if any extra genes that were moved to the side of the cluster are still beside the cluster, the genes are move back inside the cluster to a position similar to the one they originally occupied.

4.4 Putting It All Together

Ancestral genome reconstructions are performed using these three main components. Initialization of the internal nodes of the tree is done from the leaves up by taking either the midpoint or one of the two endpoints (along an edit sequence) of an internal node's two children and discarding any

genes not allowed by the median gene content. This method accounts for all three of the cases in Figure 4 and produces labels with the desired gene content. New medians are computed locally node by node in a postorder traversal of the tree, so as to propagate information from the leaves towards the root. Whenever a median is found that reduces the local score at a node, it immediately replaces the previous label at that node; that node and all its neighbors are then marked for further update.

5 Testing

We used our label reconstruction method on the bacterial dataset as well as on simulated datasets. With simulated datasets, we know the true labels for the internal nodes as well as the exact evolutionary events along each edge, so that we can test the accuracy of the reconstruction—whereas the reconstruction for the biological dataset only provides us with a conjecture. The goal of our experiments was to generate datasets roughly comparable to our biological dataset so that our experimental results would enable us to predict a range of accuracy for the results on the biological dataset.

The simulated data was created using the same tree as for the bacterial dataset; edge lengths were assigned to the tree based on our best estimate of the edge lengths for the bacterial genomes. To keep the data consistent, edge lengths were interpreted as the number of operations per gene rather than as an absolute number, which allows us to use the same value for genomes of different sizes. The tree was labeled by first constructing a root genome, then transforming it along each edge with the prescribed number of operations. The allowed operations are insertions, deletions, and inversions. In moving from the root to the leaves, a particular gene can only be inserted along one edge of the tree—multiple insertions, even along separate paths, are not allowed. Once all nodes have thus been assigned genomes, the leaf genomes are used in our reconstruction procedure and the results of the reconstruction, in terms of gene content and gene order at each internal node, compared with the "true" tree, i.e., the tree generated in the simulation.

We also tested the cluster condensation on triples among the bacterial genomes that lay close to each other on the tree. The number of genes in the three genomes that formed clusters was measured.

6 Results

Reconstruction of ancestral genomes for the bacterial genomes took around 24 hours. The midpoint-initialization proved quite strong: the only genomes to be updated in the subsequent local improvement procedure were the two children of the root. (These two genome, nodes 1 and 3 in Figure 5, are the most likely to be updated since they are the only neighboring genomes in which one neighbor was not used to create the other.) When we used endpoint-initialization, three internal nodes were updated (nodes 1, 5 and 7 in Figure 5), and the score of the entire tree was 2.8% lower than the score when using midpoint-initialization. This finding may indicate that the initialization is very good, but it may also reflect the large numbers of local optima in the search space—a similar finding was reported for the simpler GRAPPA [13]. It should be noted that, when calculating

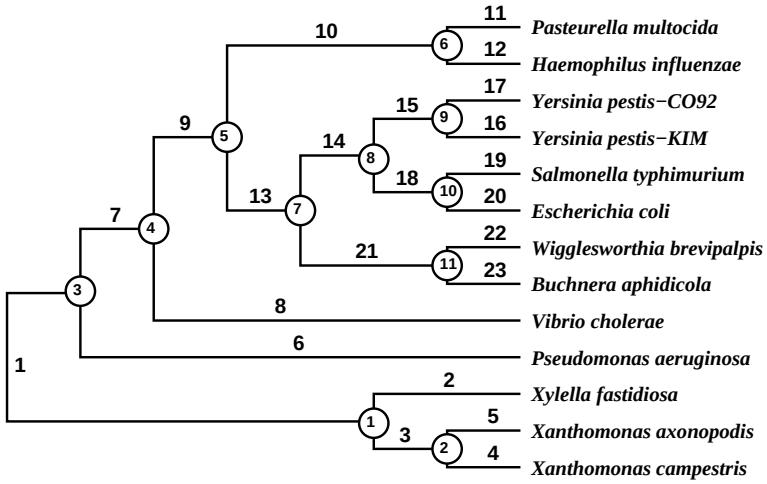

Figure 5: The bacterial tree, indicating the numbering of the edges and internal nodes used in Figures 7 and 6.

medians, only four different midpoints in the child-to-child sorting sequence are used; from each of these midpoints, only three midpoints in the sorting sequence from the intermediate genome to the parent are tested. Thus we only perform a very shallow search and could easily miss a better solution. Interestingly, though, when we did a slightly more thorough search with ten midpoints from child to child and four midpoints from intermediate to parent, using endpoint-initialization, the tree score was slightly worse than in the shallower analysis (less than 1%). The running time was 85 hours, but the same three internal nodes were updated. Still, this represents a very small part of the possible search space; the reason for this very restricted range is that the heuristic of Marron *et al*. [11], used to match and rename duplicate copies of genes, runs very slowly, consuming over 90% of the computing time. (This heuristic is thus also an obvious target for algorithmic engineering; a faster implementation will enable us to conduct a broader search.)

We simulated 100 labelings of the tree with a root genome size of 200 genes for each of five scenarios: inversion only, no deletions, no insertions, low levels of insertion and deletion, and high levels of insertion and deletion. Endpoint-initialization was used in all scenarios. The leaf genomes produced in our simulations ranged in size from 70 genes to 400 genes. We compared the predicted gene content of the internal nodes with the actual gene content. As expected (due to our restriction on generation), the predicted gene content always matched, except when a gene copy that was present at an internal node was lost in all leaves. Failure to detect this kind of missing gene is unavoidable in a gene-order analysis since the deletion from all leaves means that no historical record is left to attest the presence of that gene in ancestral genomes. When we compared the number of operations over all edges in reconstructed trees versus the original simulated tree, the score for the tree was suboptimal, as illustrated in Table 1. These suboptimal results are to be expected, and in fact the rather tight distribution in overestimating the score for the tree indicates that the error is not a random process, but a result of some aspect of our reconstruction method, one that may lend itself to reverse mapping.

We compared edge lengths in the reconstructed trees with those in the true trees by calcuating

Table 1: Error Percentage in Tree Scores

	Avg error	Min error	Max error
Inversion only	63.2%	57.3%	67.4%
No deletions	62.6%	54.8%	70.7%
No insertions	45.2%	37.6%	54.3%
Low insertion/deletion	56.4%	46.7%	64.8%
High insertion/deletion	34.9%	25.1%	46.4%

the ratio of the lengths for each edge (Figure 6). A perfect reconstruction would give a ratio of 1.0. Edges further from leaves have average ratios further from 1.0 and also have higher variances. About half of the 23 edges are within a factor of two of the true edge length, and another quarter are within a factor of four.

We also calculated the number of operations needed to convert the reconstructed genome labels at internal nodes into the corresponding labels from the true tree. Distances are normalized by dividing by the size of the tree tree genome. For this graph, a perfect reconstruction would give edit distances of zero. Here again, internal nodes closer to leaves are much closer to the true ancestral gene orders.

We tested the cluster condensation on triples of closely-related bacterial genomes. The number of genes that fell into clusters, and thus the number of genes that could be condensed away, is a lower bound on the clustering potential in the actual tree, because the neighbors of an internal node should be more closely related than three leaves in the tree. Condensation would remove the same number of genes from each genome, so the maximum possible condensation is determined by the smallest of the three genomes considered. In the cases we examined, it was possible to condense away on average 21% of the size of the smallest genome (ranging from 13% to 31%). This was a relatively encouraging result. Unfortunately, the cluster condensation is heavily dependent on the heuristic that matches and renames duplicate gene copies. As long as the code for the renaming procedure is such a bottleneck in the larger analysis, the benefits of working with smaller genomes will be lost due to the time necessary to condense the genomes down to the smaller size.

7 Conclusions

We have successfully produced a framework under which we are able to compute ancestral gene orders for modern bacteria. The number of operations over the tree is suboptimal, but not unreasonable. Reconstructed edges and internal labels which are closer to the modern genomes are much more accurate than those further in the tree from "known" data. We also have shown that, under certain simplifying assumptions, we are able to recover consistently the gene content of the ancestral genomes of simulated genomes. The size and complexity of the genomes mean that only a very shallow search of the space of possible ancestral genomes is possible: our results are undoubtedly heavily impacted by that problem, but we have pushed the size boundary for phylogenetic analysis

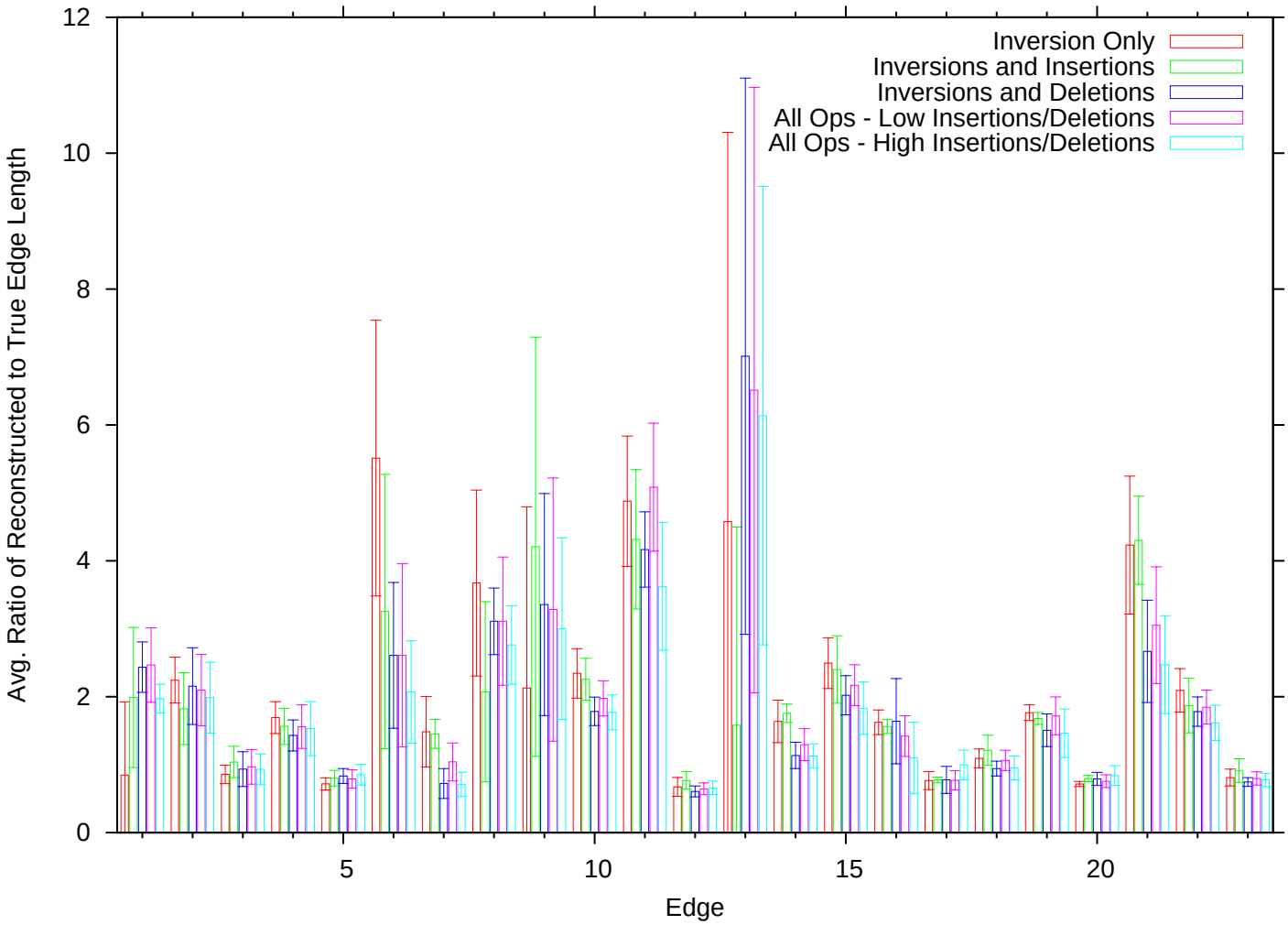

Figure 6: The average ratio between the reconstructed edge length and the corresponding true edge length for each edge in the tree. Error bars are standard deviation. Edge numbers are as shown in Figure 5.

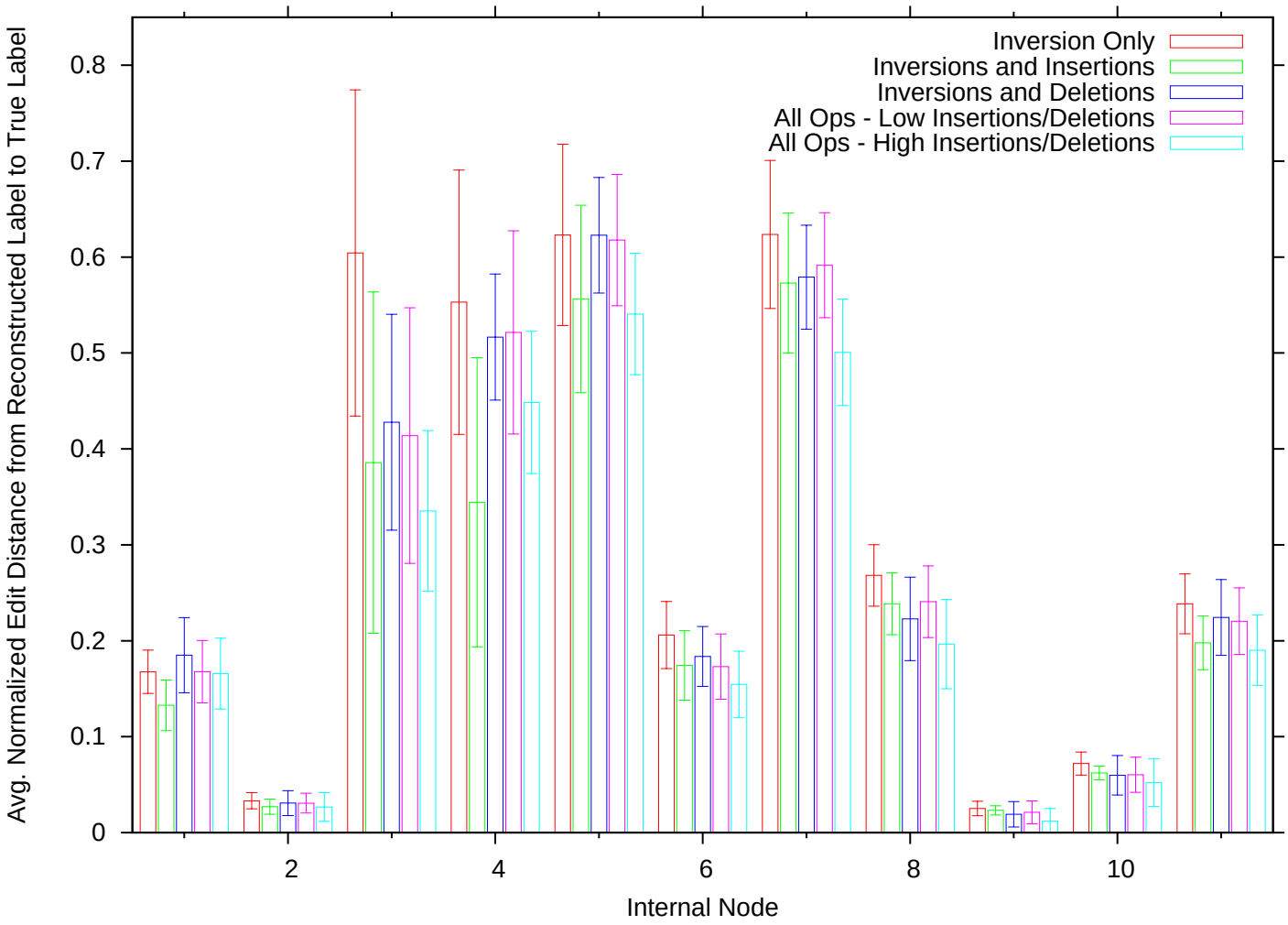

Figure 7: The average normalized edit distance from each reconstructed label to the true label for each internal node of the tree. Error bars are standard deviation. Internal node numbers are as shown in Figure 5.

with gene orders by an order of magnitude.

8 Acknowledgments

We thank our colleagues at the University of Arizona, whose help in this work has been invaluable: Nancy Moran (E. Lerat's postdoctoral advisor) and Howard Ochman and his postdoctoral student Vincent Daubin. We also thank Jens Stoye for providing the source code to the cluster-finding programi and our local colleagues Mark Marron and Krister Swenson for many useful discussions. J.V. Earnest-DeYoung and B.M.E. Moret gratefully acknowledge support by the National Science Foundation under grants EF 03-31654, IIS 01-13095, IIS 01-21377, and DEB 01-20709; E. Lerat, J.V. Earnest-DeYoung, and B.M.E. Moret (the latter two through a subcontract to the U. of Arizona) gratefully acknowledge support by the NIH under grant 2R01GM056120-05A1.

References

[1] D.A. Bader, B.M.E. Moret, T. Warnow, S.K. Wyman, and M. Yan. *GRAPPA (Genome Rearrangements Analysis under Parsimony and other Phylogenetic Algorithms)*. `www.cs.unm.edu/~moret/GRAPPA/`.

[2] A. Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. In *Proc. 12th Ann. Symp. Combin. Pattern Matching (CPM'01)*, volume 2089 of *Lecture Notes in Computer Science*, pages 106–117. Springer Verlag, 2001.

[3] A. Bergeron and J. Stoye. On the similarity of sets of permutations and its applications to genome comparison. In *Proc. 9th Int'l Conf. Computing and Combinatorics (COCOON'03)*, volume 2697 of *Lecture Notes in Computer Science*, pages 68–79. Springer Verlag, 2003.

[4] M.A. Clark, N.A. Moran, and P. Baumann. Sequence evolution in bacterial endosymbionts having extreme base composition. *Mol. Biol. Evol.*, 16:1586–1598, 1999.

[5] M.E. Cosner et al. An empirical comparison of phylogenetic methods on chloroplast gene order data in Campanulaceae. In D. Sankoff and J. Nadeau, editors, *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*, pages 99–121. Kluwer Academic Pubs., Dordrecht, Netherlands, 2000.

[6] N. El-Mabrouk. Genome rearrangement by reversals and insertions/deletions of contiguous segments. In *Proc. 11th Ann. Symp. Combin. Pattern Matching (CPM'00)*, volume 1848 of *Lecture Notes in Computer Science*, pages 222–234. Springer Verlag, 2000.

[7] S. Hannenhalli, C. Chappey, E. Koonin, and P.A. Pevzner. Genome sequence comparison and scenarios for gene rearrangements: A test case. *Genomics*, 30:299–311, 1995.

[8] S. Heber and J. Stoye. Algorithms for finding gene clusters. In *Proc. 1st Int'l Workshop on Algorithms in Bioinformatics WABI'01*, volume 2149 of *Lecture Notes in Computer Science*. Springer Verlag, 2001.

[9] J.G. Lawrence and H. Ochman. Amelioration of bacterial genomes: Rates of change and exchange. *J. Mol. Evol.*, 44:383–397, 1997.

[10] E. Lerat, V. Daubin, and N.A. Moran. From gene trees to organismal phylogeny in prokaryotes: The case of the γ-proteobacteria. *PLoS Biology*, 1:101–109, 2003.

[11] M. Marron, K.M. Swenson, and B.M.E. Moret. Genomic distances under deletions and insertions. In *Proc. 9th Int'l Conf. Computing and Combinatorics (COCOON'03)*, volume 2697 of *Lecture Notes in Computer Science*, pages 537–547. Springer Verlag, 2003.

[12] A. McLysaght, P.F. Baldi, and B.S. Gaut. Extensive gene gain associated with adaptive evolution of poxviruses. *Proc. Nat'l Acad. Sci. USA*, 100(26):15655–15660, 2003.

[13] B.M.E. Moret et al. A new implementation and detailed study of breakpoint analysis. In *Proc. 6th Pacific Symp. on Biocomputing (PSB'01)*, pages 583–594. World Scientific Pub., 2001.

[14] B.M.E. Moret, A.C. Siepel, J. Tang, and T. Liu. Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. In R. Guigó and D. Gusfield, editors, *Proc. 2nd Int'l Workshop on Algorithms in Bioinformatics WABI'02*, volume 2452 of *Lecture Notes in Computer Science*, pages 521–536. Springer Verlag, 2002.

[15] B.M.E. Moret, J. Tang, L.-S. Wang, and T. Warnow. Steps toward accurate reconstructions of phylogenies from gene-order data. *J. Comput. Syst. Sci.*, 65(3):508–525, 2002.

[16] B.M.E. Moret, J. Tang, and T. Warnow. Reconstructing phylogenies from gene-content and gene-order data. In O. Gascuel, editor, *Mathematics of Evolution and Phylogeny*. Kluwer Acad. Publ., 2004.

[17] R. Overbeek et al. The use of gene clusters to infer functional coupling. *Proc. Nat'l Acad. Sci. USA*, 96(6):2896–2901, 1999.

[18] J. Parkhill et al. Complete genome sequence of a multiple drug resistant *Salmonella enterica* serovar Typhi CT18. *Nature*, 413:848–852, 2001.

[19] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):990–917, 1999.

[20] C.K. Stover et al. Complete genome sequence of *Pseudomonas aeruginosa* PAO1, an opportunistic pathogen. *Nature*, 406:959–964, 2000.

[21] J. Tang and B.M.E. Moret. Scaling up accurate phylogenetic reconstruction from gene-order data. In *Proc. 11th Int'l Conf. on Intelligent Systems for Molecular Biology ISMB'03*, volume 19 (Suppl. 1) of *Bioinformatics*, pages i305–i312, 2003.

[22] J. Tang, B.M.E. Moret, L. Cui, and C.W. dePamphilis. Phylogenetic reconstruction from arbitrary gene-order data. In *Proc. 4th Int'l IEEE Conf. on Bioengineering and Bioinformatics BIBE'04*. IEEE Press, 2004.

[23] R.H. Waterston et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–562, 2002.