GASTS: Parsimony Scoring under Rearrangements

Andrew Wei Xu and Bernard M.E. Moret

Laboratory for Computational Biology and Bioinformatics, EPFL,
EPFL-IC-LCBB INJ230, Station 14, CH-1015 Lausanne, Switzerland
andywsureway@gmail.com, bernard.moret@epfl.ch

Abstract. The accumulation of whole-genome data has renewed interest in the study of genomic rearrangements. Comparative genomics, evolutionary biology, and cancer research all require models and algorithms to elucidate the mechanisms, history, and consequences of these rearrangements. However, rearrangements lead to NP-hard problems, so that current approaches, such as the MGR tool, are limited to small collections of genomes and low-resolution data of a few hundred syntenic blocks.

We describe the first algorithm for rearrangement analysis that scales up, in both time and accuracy, to modern high-resolution genomic data. Our main contribution is GASTS, an algorithm for scoring a fixed phylogenetic tree: given a tree and a collection of genomes, one for each leaf of the tree, each genome given by an ordered list of syntenic blocks, GASTS infers genomes for the internal nodes of the tree so as to minimize the sum, taken over all tree edges, of the pairwise genomic distances between tree nodes. We present the results of extensive testing on both simulated and real data showing that our algorithm runs several orders of magnitude faster than existing approaches and scales up linearly instead of exponentially with the size of the genomes involved; on the small instances that current approaches can complete in a day, our algorithm also returns much better scores. In simulations, our tree scores stay within 0.5% of the model value for trees up to 100 taxa and genomes of up to 10,000 syntenic blocks. GASTS enables us to attack heretofore unapproachable problems, such as accurate ancestral reconstruction of large genomes and phylogenetic inference for high-resolution vertebrate genomes, as we demonstrate on a set of vertebrate genomes with over 2,000 syntenic blocks.

1 Introduction

Genomic rearrangements were discovered early in the 20th century [20], but their systematic study started with the spread of sequencing technologies. In 1987 Day and Sankoff [7] proposed two major problems about rearrangements: the *edit distance*—given two genomes and a model of rearrangements, find the shortest sequence of rearrangements that transforms one input genome into the other; and the *median*—given three genomes, construct a fourth genome that minimizes the sum of its pairwise distances to the other three. The edit distance is computable in linear time for most models, while the median is NP-hard for most models [9]. Phylogenetic reconstruction from rearrangement data attracted attention, as rearrangements are "rare genomic events" [17] and thus might help resolve difficult questions about ancient branching patterns in

T.M. Przytycka and M.-F. Sagot (Eds.): WABI 2011, LNBI 6833, pp. 351–363, 2011.

evolution, but the computational complexity of parsimonious approaches precluded widespread application of the approach. The best available tool for the purpose, MGR [6] and its extensions, scales poorly in both accuracy and running time with genome size (to a few hundred syntenic blocks at most) and also with the expected length of the phylogenetic tree.

In this paper, we describe GASTS (Generalized Adequate Subtree Tree Scoring), a tree-scoring method based on generalized adequate subgraphs. Scoring a fixed tree given its leaf genomes is the core problem of phylogenetic inference, ancestral reconstruction, and all other uses of phylogenetic trees. The problem is NP-hard, as it subsumes the median problem. GASTS scales linearly with the expected length of the tree, and, in extensive simulation tests, returns tree scores within 0.5% of the model tree score. GASTS runs in seconds on datasets that MGR fails to complete in 24 hours and returns better scores on those datasets that MGR can complete. GASTS provides accurate values within the full range of practical applications in contemporary comparative genomics.

We test GASTS on real data and on simulations, the latter to assess scalability and absolute accuracy; in addition, we test its use within the contexts of both ancestral reconstruction and phylogenetic inference. Our simulations show that GASTS enables highly accurate tree reconstruction: even for difficult datasets, the expected error remains well below a single edge. On real data, GASTS enabled us to infer in just a few minutes phylogenies from high-resolution vertebrate data (over 2'000 syntenic blocks). Our new approach provides the kind of high-throughput tool needed today in comparative genomics and opens new areas of genomics to computational investigation.

2 Rearrangements and Phylogenetic Analysis

Rearrangement data was used in phylogenetic analysis 80 years ago by the Sturtevant and Dobzhansky [21]. Blanchette, Bourque, and Sankoff [5] introduced the first algorithmic approach to the reconstruction of a phylogenetic tree from rearrangement data, BPAnalysis. The algorithm seeks the tree and internal genomes which together minimize the total number of *breakpoints*—adjacencies present in one genome, but absent in the other. Moret *et al.* [14] reimplemented this approach in their GRAPPA tool and extended it to *inversion distances*—inversions are the best documented of the hypothesized mechanisms of genomic rearrangements; they also published the first studies of the median problem [13, 18]. Their work focused on unichromosomal genomes; to handle multichromosomal genomes, Bourque and Pevzner [6] proposed MGR, based on GRAPPA's distance computations. Whereas BPAnalysis and GRAPPA search all trees and report the one with the best score, MGR uses a heuristic sequential addition method to grow the tree one species at a time.

Computing the parsimony score of a fixed tree for rearrangement data is NP-hard [9], even if the tree has only three leaves, and for any of breakpoint distance, inversion distance, and *DCJ distance*—a DCJ (Double-Cut-and-Join) operation changes two adjacencies at a time and provides a general framework that subsumes all other rearrangement operations [3]. The approach in BPAnalysis and GRAPPA is iterative refinement:

```
A. assign some arrangement to each internal node
B. repeat
     select an internal node x with a neighbor
       that was just assigned a new arrangement
     compute the median of the arrangements stored
       at the three neighbors of x
     if the median improves on the arrangement
       stored at x, assign the median to x
   until no change
```

The key problem here was long held to be the computation of medians, a problem that one of us started studying 8 years ago [13, 18] and that we recently solved well enough for most practical purposes [15] in the context of unichromosomal genomes, using the concept of adequate subgraphs developed by one of us [26]. Using this median solver, we discovered that an equally crucial problem is the initialization phase: because local optima abound, an iterative refinement approach does well only when started with a very good initial assignment.

Our algorithmic contribution is a novel method for accurate initialization of internal (ancestral) genomes in a fixed tree. This contribution leads to an accurate tree-scoring algorithm, which in turn we use to run two different styles of phylogenetic inference, one through brute-force search (score all possible trees and retain the best) and one through incremental construction. One of us used GASTS to explore patterns beyond generalized adequate subgraphs on small phylogenetic trees [23], resulting in a collection of less constrained, yet more generally applicable configurations that can further reduce the running time of our approach on the hardest inputs.

3 A Heuristic for Multichromosomal Medians

Our approach is based on a fast and accurate heuristic for the inversion median that we recently developed [15] and now briefly review. This heuristic uses adequate subgraphs, developed by one of us [26, 25], which provide optimality-preserving decompositions of the median problem and thus the basis for a divide-and-conquer algorithm for the median. Finding such decompositions in every instance remains an open problem; in practice, however, a few of the simplest decompositions suffice in almost every case—and when they do not, they can be supplemented by simple heuristics to produce highly accurate solutions [15].

We extend that work to handle the *capped* version of the underlying multiple breakpoint graphs—caps being necessary to move from unichromosomal to multichromosomal genomes [24]. At each step, our new algorithm either detects a capped adequate subgraph and decomposes the current instance into subproblems, or searches a polynomial number of ways to add one more adjacency into the median genome, selecting one choice according to a criterion that matches the definition of (capped) adequate subgraphs. Since this algorithm simply combines our design for inversion medians with Xu's extensions to capped multiple breakpoint graphs, we do not give a more detailed description, with one exception. Because DCJ-based approaches can produce extra circular chromosomes, our algorithm greedily merges such circular chromosomes

with regular linear chromosomes so as to minimize the incremental increase of median scores. When tested on simulated data, these heuristics demonstrate very high accuracy, in line with our results for inversions [15].

4 Initialization with Adequate Subgraphs

In their paper on BPAnalysis [5], Sankoff and Blanchette proposed several initialization methods for step (A); both they and Moret *et al.* [14] settled on two approaches as the most promising. The *nearest-neighbor* approach assigns to each internal node the median of its three nearest leaves (one in each of the three adjacent subtrees, breaking ties within a subtree arbitrarily). The more complex *adjacency-parsimony* approach finds arrangements that minimize the number of adjacencies not already present in the data. This second method uses information in a more global manner than the first, although experiments by both sets of authors using breakpoint or inversion distances showed no gain in practice. Moreover, both initialization methods fail on difficult data, as the iterative refinement step (B) typically runs at most twice on each node—in other words, both initialization methods tend to start the algorithm in a local optimum, preventing any improvements and returning poor solutions. What is needed is an initialization that avoids local optima or uses only those very close to the global optimum—all of which argues for a better use of global information.

Our new initialization method put global information to use through adequate subgraphs and thus meshes well with the refinement phase of the scoring procedure, which uses adequate subgraphs to compute medians. We initialize internal nodes progressively: in order for an internal node to be a candidate for initialization, two of its three neighbors must be already initialized (or leaves). The third, while typically not be initialized, is not devoid of information, so our method summarizes the data available in the third subtree (rooted at the uninitialized neighbor) into a set of weighted adjacencies. Thus information used in initializing a node consists of two 0-1 sets of adjacencies from the two initialized neighbors and one weighted set of adjacencies from the third neighbor. A suitable choice of the node to be initialized is thus a generalized version of a median, one that takes into account the weighted nature of adjacencies in the third node.

4.1 Weighted Adjacencies and a Weighting Schema

We define a *perspective* at a node along one of its incident edges to be the subtree rooted at the other end of that edge. In Fig. 1, numbered nodes are uninitialized nodes and labelled nodes are leaves or initialized nodes. The subtree rooted at node 2 and extending rightward is the perspective at node 1 along edge z. We do not use the entire perspective in guiding the initialization of node 1, however; instead, we consider only the *directive nodes* in the perspective, that is, initialized nodes or leaves connected to the node of interest via a path of uninitialized nodes. In Fig. 1, nodes a, b, and c are initialized and connected to node 1 through paths of uninitialized nodes and so are directive nodes.

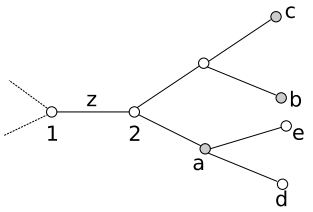


Fig. 1. A perspective at node 1 along edge z; shaded nodes are the directive nodes a, b, and c

Given a directive node g, we define the indicator function $I_x(g)$ to be 1 if adjacency x is present in g and 0 otherwise. We now define *weighted adjacencies* on perspective p at internal node i as follows: the weight w_x for each adjacency x is given by

$$w_x = \sum_{\text{directive nodes } g \text{ in } p} I_x(g) \cdot 3^{-d+1}$$

where d is the depth of node g—the number of edges on the path connecting nodes g and i. (The exponential decay reflects the exponential growth in the number of possible directive nodes in a perspective.)

Xu and Sankoff [26] showed that, under the DCJ model, if two of the three neighbors contain the same adjacency, then the median also contains it; under our weighting scheme, this property is preserved. Consider the situation depicted in Fig. 2. Say that nodes b, c, and e contain some adjacency x, while nodes a and d do not; should x be assigned to node 1? The presence of x in node b and the "fractional presence" of x in node 2 (the root of the perspective along edge z) together form a generalized adequate subgraph (for which see below) causing an assignment of x to node 1; in consequence, x also gets assigned to nodes 2 and 3.

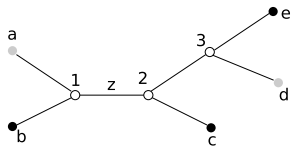


Fig. 2. A perspective at node 1 along edge z and nodes that share adjacency x (in black)

4.2 Using the Generalized Adequate Subgraphs

Adequate subgraphs capture optimal substructures, as defined and described in previous work by Xu *et al.* [15, 24, 26]. An adequate subgraph on n vertices is a connected subgraph of the multiple breakpoint graph, with which another set of edges can form at least $3n/4$ bicolored cycles. In the generalized version used in this paper, we count each cycle according to its weight, which we define to be the smallest weight along the cycle. Unlike the adequate subgraphs for the median problem, these *generalized adequate subgraphs (GAS)* do not have an optimality guarantee: they form the basis for heuristic assignment of the median in the initialization phase.

Using fractional rather than 0-1 weights greatly increases the number of GAS, so that detecting them can become a major computational task. However, each node to be initialized has two initialized (or leaf) nodes. The adjacencies in these two neighbors form bicolored cycles and paths in the breakpoint graph; we search for GAS only in these cycles and paths. This approach is not just efficient, but also reasonable: the genome to be assigned lies on the DCJ edit path between those two genomes and minimizes the average distance to the directive nodes. This genome may be slightly biased toward to the two known neighbors, when their edit distance is smaller than the real number of rearrangements. Since the genome we want to assign is well balanced between local and global information, the bias can be remedied in the second phase of the scoring procedure.

5 Testing Our New Tree-Scoring Method

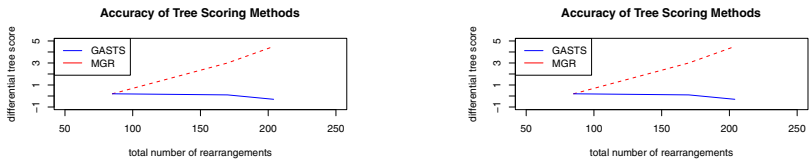
We compare GASTS with the only existing method that can handle multichromosomal data, MGR. However, MGR failed to complete within 24 hours of computation on almost every test case of medium to large size, so, in order to get some basic comparisons, we introduce a third method, purely a strawman to enable some comparison of GASTS scores and running times against at least one competitor on nontrivial datasets. This third method, denoted NNM, uses the standard nearest-neighbor initialization, but computes medians in both steps (A) and (B) using our GAS-based median solver.

We test tree-scoring quality and scalability on various model conditions with large genomes. Model conditions vary from 10 to 80 genomes, each made of up to 2,000 syntenic blocks. These models conditions are produced according to standard practice in phylogenetic reconstruction [10] as follows. We generate a rooted tree topology, assign a "genome" to the root, then simulate the evolution of the genome down the tree to the leaves. Trees are generated either following a standard birth-death process, in which case the length of each tree branch is determined during the construction of the tree topology, or by picking a tree uniformly at random among all distinct rooted trees on the assigned number of leaves, in which case we assign the same length to every edge of the tree. A tree with its branch lengths and root genome is a *model tree*; we view the length of branch of a model tree as the expected value of the length of that branch. From a model tree, we generate multiple datasets. To generate one dataset, we first assign real lengths to the edges of the tree by sampling from a Poisson distribution with a mean equal to the edge length in the model tree; we then "evolve" the root genome down the tree to obtain genomes at the leaves. To evolve a parent genome down to its child, we apply to the parent genome a number of rearrangements (chosen uniformly at random) equal to the length of that branch. The resulting leaf genomes, plus the tree topology (but not its branch lengths) form the dataset. Repeating the process on the same model tree produces new datasets. The tree with its edge lengths in each simulation is a *real tree*; the sum of its edge lengths is the *real tree score*. The sum of the edge lengths assigned to an instance by the tree-scoring procedure is the *inferred tree score*. The *differential tree score* is the difference between the inferred tree score and the real tree score. Since the rearrangement scenarios are random, the real tree score need not be the smallest score achievable for the tree.

MGR's computational limitations led us to generate a first group of model conditions of modest scope: 10 genomes of size 100, with rearrangements limited to 80 to 340 inversions. As our own procedures, GASTS and NNM, can handle much larger trees and genomes, we generated a second group of model conditions with from 10 to 80 genomes, each with 20 linear chromosomes and a total of 2'000 syntenic blocks, using inversions, translocations, fissions, and fusions. In these datasets, model tree scores vary very widely, from 0.5 to 23 times the number of syntenic blocks. Finally, to test the computational scaling of GASTS, we generated a smaller number of large datasets, with 10 to 80 genomes, each made of 20 chromosomes with a total of 10'000 syntenic blocks.

On the first group of model conditions, GASTS and NNM ran in less than one second on every instance, whereas MGR took a very long time (often days) on over half of the instances—those derived from model trees with large scores. In order to run enough datasets, we set an arbitrary cutoff of 24 hours of computation per instance (on a dedicated CPU). Fig. 3 shows the difference between the inferred tree score and the real tree score for GASTS and MGR for the first group of model conditions. (Tree scores obtained from NNM were similar to, but less accurate than, those obtained from GASTS, so we do not present them here.) The horizontal axis denotes the real tree score, while the vertical axis denotes the differential tree score for each of GASTS and MGR. Inferred tree scores obtained by GASTS correctly trend downwards (for larger model tree scores we expect parsimony scores to be smaller than the model tree scores) and are consistently better than those obtained by MGR, which trend upwards, indicating increasing errors.

In the second group we used 50 different model conditions, with 10 datasets each. Here our comparison is between GASTS and NNM, because MGR could not complete any of these datasets within several weeks. On these datasets, GASTS runs in a few seconds for the smaller datasets and in a few minutes for the larger ones. Since both methods use the same median solver and share the same refinement step, the results indicate the differences in the initialization method. The NNM approach suffers when the number of leaves grows, as it must then compute medians of very divergent leaves, which can take significant time; in contrast, GASTS initialization ensures that every median computation is a median of three neighboring genomes. The speed difference is large enough that, with the NNM approach, we could not complete instances for trees with 20 or more taxa. Fig. 4 shows the results, including higher-resolution plots of the final differential tree scores for GASTS. For both methods, we report the differential



(a) on birth-death trees (b) on random trees

Fig. 3. Difference between inferred and model tree scores for GASTS and MGR, only on those datasets completed by MGR within 24 hours

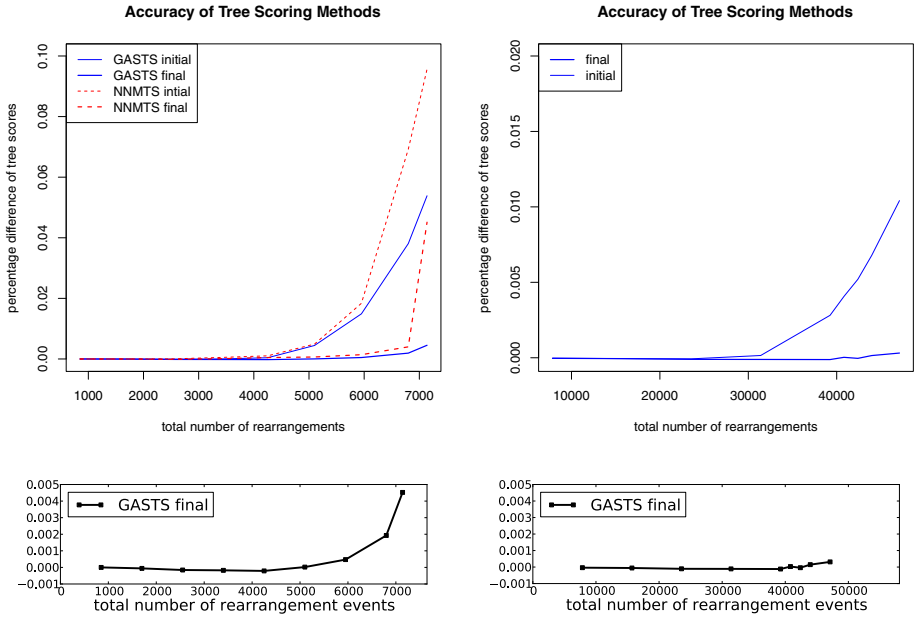
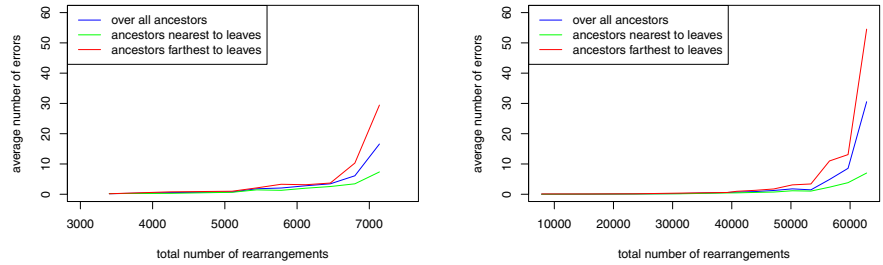


Fig. 4. Accuracy of GASTS and NNM, for 10-taxon trees (left) and 80-taxon trees (right), as a function of the total number of rearrangement events. Thin lines show initial scores, thick lines final scores. The lower plots zoom in on the final score produced by GASTS.

tree score right after initialization and at the completion of the scoring procedure. Our initialization method clearly dominates the NNM method, particularly for large model tree scores. The second phase of scoring—the iterative refinement—has little effect for smaller distances, but quite dramatic ones for large distances, showing that our initialization method helps the refinement procedure in avoiding local optima. Note that the inferred score keeps tracking the real score even at very large evolutionary distances, as evidenced in the lower plots. Thus GASTS is both highly accurate and very robust—in particular, tree scores inferred by GASTS can meaningfully be compared.

6 Ancestral Genome Reconstruction

In scoring trees, we assign arrangements to the internal nodes. Such assignments should not be confused with true ancestral reconstruction, as they do not obey specific biological constraints, but simply optimize a distance function. Yet many ancestral reconstruction approaches to date use median computations as part of their guidance. Given the high accuracy of our scoring algorithm, we may expect it to assign arrangements to internal nodes that come quite close to the "true" ancestral genomes. Since our simulations create these "true" ancestral genomes, we can compare them to those reconstructed by our scoring procedure. Indeed, the genomes assigned by GASTS are very close to the true ancestral genomes, as shown in Fig. 5. Interestingly, ancestral nodes farthest from the leaves are not much worse than those closest to the leaves, which also



(a) on random tree with 10 species (b) on random tree with 80 species

Fig. 5. Average total DCJ distance between genomes assigned to internal nodes and the corresponding "true" ancestors

supports the quality of the scoring. (For the largest distances, direct reconstruction fails due to the enormous number of optimal solutions, as was shown for a collection of gamma proteobacteria [8].) Over all 22 model conditions, the average total distance was less than 30; in 20 of the 22 model conditions it was less than 10 and, in 10 model conditions, it was less than 1. Thus the high accuracy of GASTS indeed provides a good starting point for ancestral reconstruction.

7 Phylogenetic Reconstruction

With good tree scoring, several reconstruction methods used with sequence data can be adapted to rearrangement data. We present experimental results for two different uses of GASTS in phylogenetic inference: an exhaustive search method that scores every tree and returns the best, and a heuristic method that builds a tree by sequential addition. Exhaustive tree search may give the best results, but is forcibly limited to at most 20 taxa—we did not attempt to use the many speed-up mechanisms of GRAPPA, as our aim was to test GASTS, not to produce a better reconstruction algorithm. Adding one taxon at a time is a very simple heuristic, but one with a long history in phylogenetic analysis, including incremental parsimony [4] and quartet-puzzling [19] for sequence-based data; in rearrangement-based phylogenetic work, MGR uses this approach. Unlike these three methods, our sequential addition method rescores every tree after each addition step. We compare these two approaches with MGR rather than with GRAPPA, so as to test performance on multichromosomal as well as unichromosomal data.

We used datasets of modest size so as to allow a comparison with MGR. Rearrangement include: (i) inversions on unichromosomal circular genomes; (ii) inversions, translocations, fusions, and fissions on multichromosomal genomes; and (iii) 80% inversions and 20% transpositions on unichromosomal circular genomes. We tested trees generated from three different models: the birth-death model, the random model, and the beta-splitting model [2] with $\beta = -1$. We report results only for the birth-death and uniform random model, as results for the beta-splitting model were similar to those for

the uniform random model. For each dataset completed by MGR in 24 hours, we compute the *Robinson-Foulds* (RF) error rate to the real tree—half of the number of edges present in one tree, but not in the other [16], divided by the number of internal edges in the model tree.

MGR finished only the smaller datasets in each group, taking about 20 minutes per dataset with tree scores of 85 and over 12 hours per dataset with tree scores of 200, its running time increased exponentially as a function of the tree score. Our exhaustive search method took just one minute on each dataset, except for a few datasets with tree scores of 340, where it took around 12 minutes. Finally, our sequential addition method took less than one second per dataset, except for a few datasets with tree scores of 340, where it took 4 seconds. Table 1 shows the error rates for the three methods averaged over 6 model conditions, using birth-death trees. All three show reasonable accuracy, with the exhaustive search the most accurate, the sequential addition second, and MGR third. The main drawback of MGR here is running time (empty table entries correspond to model conditions under which MGR could not complete more than a few test instances), not accuracy: the phylogenetic signal in the data is strong enough that correct inferences can be made even if the reconstruction method does not take full advantage of the data. The introduction of transpositions worsened the results for all three approaches, most significantly for MGR—as might be expected, since a transposition takes two operations in the DCJ framework used in our methods, but three operations in the inversion framework used by MGR.

Table 1. RF error rates (in %)on datasets of 10 genomes of size 100 on birth-death trees, as a function of the number of rearrangements

(a) circular genome, inversions & transpositions

	85	170	204	238	255	340
exhaustive	0.0	0.0	0.0	5.2	2.6	5.2
sequential	0.0	0.0	1.8	5.2	4.3	14.3
MGR	0.0	7.9	7.1			

(b) linear genome, all rearrangements

	85	170	204	238	255	340
exhaustive	0.0	0.0	0.0	0.0	2.9	5.7
sequential	0.0	1.4	0.0	0.0	4.3	2.9
MGR	0.0					

To test the accuracy of the sequential addition heuristic on large trees with large genomes, we generated eight model conditions (again with 10 datasets each) using birth-death trees on 80 taxa, with genomes of 2,000 syntenic blocks allotted among 20 linear chromosomes. Four of the model conditions use a mix of 20% transpositions,

Table 2. Average RF error rates (in %) and running times (in mins.) for sequential addition on datasets of 80 genomes of 20 chromosomes with 2,000 blocks as a function of the number of rearrangements

(a) no transpositions

	5,000	10,000	15,000	20,000
error	0.0	0.0	0.0	0.78
time	21.0	21.8	26.7	88.3

(b) 20% transpositions

	5,000	10,000	15,000	20,000
error	0.0	0.0	1.56	1.95
time	22.9	28.7	74.3	845.0

while the other four do not use transpositions; the total number of rearrangement events ranges ranges from 5,000 to 20,000. Table 2 shows error rates and running times; the approach shows excellent scalability and good accuracy, with error rates consistently smaller than 2%.

8 Applications on Real Data

We applied our exhaustive approach to two biological datasets. The first is quite small: the well studied Campanulaceae chloroplast dataset, with 13 taxa, each genome a circular chromosome with 105 genes. The second is a collection of 8 vertebrate genomes (7 mammals and chicken), from a 13-way genomic alignment downloaded from ENSEMBL, from which we retained the best assembled genomes. We then generated synteny blocks at five different resolutions: 1Kbp, 3Kbp, 10Kbp, 30Kbp, and 100Kbp—meaning that the resolution value was used as a lower bound on the size of acceptable blocks. (We ignored contigs that contained a single synteny block.)

Previous studies on the Campanulaceae dataset reported best tree scores of 64 inversions and 64 DCJ operations [1, 11, 13]. Our approach improved the DCJ tree score by finding 138 trees with a score of 63, all with a different topology from the tree with DCJ score of 64 reported in [1]. The vertebrate set has not been analyzed by anyone as whole genomes at these resolutions, so direct comparisons are not possible. Nor is comparison with sequence-based analyses fruitful at this stage, since the use of rearrangement data in phylogenetic analysis is too immature for an interpretation of the resulting trees; moreover, the decomposition of whole genomes into syntenic blocks is itself a poorly resolved problem, one aggravated by the incomplete assembly of many of these genomes. Instead, our purpose with these datasets is to demonstrate the scalability of our approach and thus pave the way for detailed studies. Our exhaustive approach completed each of the five datasets in a few minutes and returned the same tree for each (the generally accepted tree), illustrated in Figure 6.

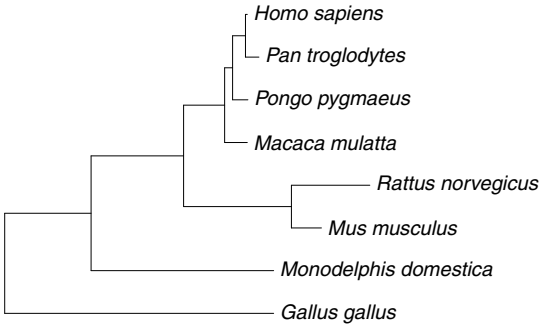


Fig. 6. Phylogeny of 8 vertebrates

9 Conclusion

We presented a new approach to phylogenetic analysis of rearrangement data that integrates past and recent work through various adaptations (such as generalized adequate subgraphs) and includes GASTS, a fast, highly accurate, and surprisingly robust scoring method for a fixed tree. We presented experimental results demonstrating that our scoring procedure scales gracefully to trees far larger than anything used to date with gene-order data, as well as to evolutionary distances that are well into saturation, all while returning tree lengths in a very narrow interval around the true tree length (generally within less than 0.1% and even in the worst cases within 0.5%). We tested our reconstruction methods under simulation and on datasets of whole genomes; the simulations indicate that the reconstruction is perfect in almost all cases, while the whole-genome datasets demonstrate the power of the method on a dataset of vertebrates with up to 10,000 markers, a scale that had been entirely out of reach until now. While our approach does not yet handle duplications and losses and so must be used with either simple genomes like organelles or large genomes represented by sequences of unique syntenic blocks (as is often done for vertebrates), it makes it possible, for the first time, to conduct nontrivial phylogenetic analyses from high-resolution genomic data.

References

1. Adam, Z., Sankoff, D.: The ABCs of MGR with DCJ. Evol. Bioinf. Online 4, 69–74 (2008)
2. Aldous, D.J.: Stochastic models and descriptive statistics for phylogenetic trees, from Yule to today. Stat. Sci. 16, 23–34 (2001)
3. Bergeron, A., Mixtacki, J., Stoye, J.: A unifying view of genome rearrangements. In: Bücher, P., Moret, B.M.E. (eds.) WABI 2006. LNCS (LNBI), vol. 4175, pp. 163–173. Springer, Heidelberg (2006)
4. Bininda-Emonds, O.R.P., Brady, S.G., Kim, J., Sanderson, M.J.: Scaling of accuracy in extremely large phylogenetic trees. In: Proc. 6th Pacific Symp. on Biocomputing (PSB 2001), pp. 547–558. World Scientific Pub., Singapore (2001)
5. Blanchette, M., Bourque, G., Sankoff, D.: Breakpoint phylogenies. In: Miyano, S., Takagi, T. (eds.) Genome Informatics, pp. 25–34. Univ. Academy Press, Tokyo (1997)
6. Bourque, G., Pevzner, P.: Genome-scale evolution: reconstructing gene orders in the ancestral species. Genome Res. 12, 26–36 (2002)
7. Day, W.H.E., Sankoff, D.: The computational complexity of inferring phylogenies from chromosome inversion data. J. Theor. Biol. 127, 213–218 (1987)
8. Earnest-DeYoung, J., Lerat, E., Moret, B.M.E.: Reversing gene erosion: reconstructing ancestral bacterial genomes from gene-content and gene-order data. In: Jonassen, I., Kim, J. (eds.) WABI 2004. LNCS (LNBI), vol. 3240, pp. 1–13. Springer, Heidelberg (2004)
9. Fertin, G., Labarre, A., Rusu, I., Tannier, E., Vialette, S.: Combinatorics of Genome Rearrangements. MIT Press, Cambridge (2009)
10. Hillis, D.M.: Approaches for assessing phylogenetic accuracy. Syst. Biol. 44, 3–16 (1995)
11. Larget, B., Kadane, J.B., Simon, D.L.: A Markov chain Monte Carlo approach to reconstructing ancestral genome arrangements. Mol. Biol. Evol. 22, 486–489 (2002)
12. Miklós, I., Mélykúti, B., Swenson, K.M.: The metropolized partial importance sampling MCMC mixes slowly on minimal reversal rearrangement paths. ACM/IEEE Trans. on Comput. Bio. & Bioinf. 7(4), 763–767 (2010)

13. Moret, B.M.E., Siepel, A.C., Tang, J., Liu, T.: Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. In: Guigó, R., Gusfield, D. (eds.) WABI 2002. LNCS, vol. 2452, pp. 521–536. Springer, Heidelberg (2002)
14. Moret, B.M.E., Wyman, S.K., Bader, D.A., Warnow, T., Yan, M.: A new implementation and detailed study of breakpoint analysis. In: Proc. 6th Pacific Symp. on Biocomputing (PSB 2001), pp. 583–594. World Scientific Pub., Singapore (2001)
15. Rajan, V., Xu, A.W., Lin, Y., Swenson, K.M., Moret, B.M.E.: Heuristics for the inversion median problem. In: Proc. 8th Asia Pacific Bioinf. Conf. (APBC 2010). BMC Bioinformatics, vol. 11(suppl. 1), p. S30 (2010)
16. Robinson, D.R., Foulds, L.R.: Comparison of phylogenetic trees. Math. Biosci. 53, 131–147 (1981)
17. Rokas, A., Holland, P.W.H.: Rare genomic changes as a tool for phylogenetics. Trends in Ecol. and Evol. 15, 454–459 (2000)
18. Siepel, A.C., Moret, B.M.E.: Finding an optimal inversion median: Experimental results. In: Gascuel, O., Moret, B.M.E. (eds.) WABI 2001. LNCS, vol. 2149, pp. 189–203. Springer, Heidelberg (2001)
19. Strimmer, K., von Haeseler, A.: Quartet puzzling: A quartet maximum likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13, 964–969 (1996)
20. Sturtevant, A.H.: A crossover reducer in Drosophila melanogaster due to inversion of a section of the third chromosome. Biol. Zent. Bl. 46, 697–702 (1926)
21. Sturtevant, A.H., Dobzhansky, T.: Inversions in the third chromosome of wild races of D. pseudoobscura and their use in the study of the history of the species. Proc. Nat'l Acad. Sci., USA 22, 448–450 (1936)
22. Tang, J., Moret, B.M.E.: Scaling up accurate phylogenetic reconstruction from gene-order data. In: Proc. 11th Int'l Conf. on Intelligent Systems for Mol. Biol (ISMB 2003). Bioinformatics, vol. 19, pp. i305–i312 (2003)
23. Xu, A.W.: On exploring genome rearrangement phylogenetic patterns. In: Tannier, E. (ed.) RECOMB-CG 2010. LNCS, vol. 6398, pp. 121–136. Springer, Heidelberg (2010)
24. Xu, A.W.: DCJ median problems on linear multichromosomal genomes: Graph representation and fast exact solutions. In: Ciccarelli, F.D., Miklós, I. (eds.) RECOMB-CG 2009. LNCS, vol. 5817, pp. 70–83. Springer, Heidelberg (2009)
25. Xu, A.W.: A fast and exact algorithm for the median of three problem—a graph decomposition approach. J. Comput. Biol. 16(10), 1369–1381 (2009)
26. Xu, A.W., Sankoff, D.: Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 25–37. Springer, Heidelberg (2008)