Probabilistic Reconstruction of Ancestral Gene Orders with Insertions and Deletions

Fei Hu, Jun Zhou, Lingxi Zhou and Jijun Tang

Abstract—Changes of gene orderings have been extensively used as a signal to reconstruct phylogenies and ancestral genomes. Inferring the gene order of an extinct species has a wide range of applications, including the potential to reveal more detailed evolutionary histories, to determine gene content and ordering, and to understand the consequences of structural changes for organismal function and species divergence. In this study, we propose a new adjacency-based method, PMAG$^+$, to infer ancestral genomes under a more general model of gene evolution involving gene insertions and deletions (indels), in addition to gene rearrangements. PMAG$^+$ improves on our previous method PMAG by developing a new approach to infer ancestral gene contents and reducing the adjacency assembly problem to an instance of TSP. We designed a series of experiments to extensively validate PMAG$^+$ and compared the results with the most recent and comparable method GapAdj. According to the results, ancestral gene contents predicted by PMAG$^+$ coincides highly with the actual contents with error rates less than 1%. Under various degrees of indels, PMAG$^+$ consistently achieves more accurate prediction of ancestral gene orders and at the same time, produces contigs very close to the actual chromosomes.

Index Terms—Ancestral Genome, Gene Order, Genome Rearrangement, Gene Insertion, Gene Deletion

1 Introduction

Gene order data has been proved to be very useful in phylogenetic reconstruction, but determining the ancestral orders and orientations of genes is still far from solved. In recent years, reconstruction the hypothetical gene orders of ancestors with or without being given the speciation history have both been studied. If the speciation history is given (in the form of a binary tree), the problem of finding ancestors at non-leaf nodes is defined as the small phylogeny problem (SPP); on the other hand, starting from a set of related species, the big phylogeny problem (BPP) searches for the phylogeny tree along with all the ancestors in the tree. Current methods to solve SPP are either event-based or adjacency-based. Event-based methods seek for a set of assignments of gene orders to each ancestor such that the number of evolutionary events is minimized. These methods are very expensive, and may not be able to find a solution even after months of computation. To overcome this problem, several adjacency-based methods were proposed, which compute the score or probability of each gene adjacency and assemble individual adjacencies into a valid permutation of gene order based on their scores or probabilities.

- *Fei Hu and Jijun Tang are affiliated with the Tianjin Key Laboratory of Cognitive Computing and Application at the Tianjin University of China, and the Department of Computer Science and Engineering at the University of South Carolina.*
  E-mail: jtang@cse.sc.edu
- *Jun Zhou and Lingxi Zhou are Ph.D. Students in the Department of Computer Science and Engineering at the University of South Carolina.*

Currently most methods are restricted to handle datasets involving only rearrangements. Under such model, species can only have equal gene content such that each gene has exactly one copy in every species. Therefore in this study we propose PMAG$^+$ as an extension to our previous method PMAG in order to efficiently handle datasets underwent a large scale of rearrangements, as well as gene deletions and insertions (indels) of a single or a segments of genes. Our experimental results on simulated datasets suggest that PMAG$^+$ can efficiently and accurately predict both ancestral gene contents and ancestral gene orders.

2 Evolution of Gene Orders

Given a set of n genes labeled as $\{1, 2, \cdots, n\}$, a genome can be represented by an *ordering* of these genes. Each gene is assigned with an orientation that is either positive, written i, or negative, written $-i$. Two genes i and j form an *adjacency* (i, j) if i is immediately followed by j, or, equivalently, $-j$ is immediately followed by $-i$. If gene k lies at one end of a linear chromosome, we let k be adjacent to an extremity e to mark the beginning or ending of the chromosome, written as (e, k) or (k, e), and called *telomere*.

Genome rearrangement operations change the ordering of genes on chromosomes. An *inversion* operation (also called *reversal*) reverses a segment of a chromosome. A *transposition* is an operation that swaps two segments of a chromosome. In case of multiple chromosomes, *translocation* breaks a chromosome and reattaches a part to another chromosome, while fusion joins two chromosomes and fission split one

chromosome into two. Yancopoulos et al. [1] proposed a universal *double-cut-and-join* (DCJ) operation that accounts for all common events. There are another set of operations which can alter the gene content in a genome. A *deletion* (also called *loss*) deletes a single or a segment of genes from the genome. Its reverse operation called *insertion* introduces one or a segment of genes that have not seen before into a chromosome at a time. *Whole genome duplication* (WGD) creates an additional copy of the entire genome of a species.

3 Methods for Solving the Small Phylogeny Problem (SPP)

In the context of event-based methods, to find a solution for SPP, it is typical to iterate over each internal node to solve for the median genomes until the sum of all edge distances (tree score) is minimized. The median problem can be formalized as follows: give a set of m genomes with permutations $\{x_i\}_{1\leq i\leq m}$ and a distance measurement d, find another permutation x_t such that the median score defined as $\sum_{i=1}^{m} d(x_i, x_t)$ is minimized. `GRAPPA` [2] and `MGR` [3] (as well as their recently enhanced versions) are two widely-referenced methods that implement a selection of median solvers for phylogeny and ancestral gene-order inference. However solving even the simplest case of median problem when m equals to three is NP-hard for most distance measurements. Progress has been made in handling genomes with unequal gene content. *Tang and Moret* proposed a two-phase method [4] in which the best gene content for the median is computed and then a branch-and-bound approach is used to determine the best ordering of these gene contents. *Zhang et al.* later extended Caprara's inversion median solver [5] and proposed a simplified DCJ-based distance computation for unichromosomal genomes with indels.

The first adjacency-based method in probabilistic framework was introduced in `InferCarsPro` [6]. The key of this method is to estimate the posterior probability of observing an adjacency in the ancestor based on an extended Jukes-Cantor model for breakpoints. With the obtained adjacency probabilities, it then uses a greedy heuristic to find a valid gene order for each ancestor. Later *Hu et al.* proposed a faster and more accurate method `PMAG` [7]. Although `PMAG` also seeks to compute the probabilities for adjacencies and uses the same greedy heuristic to assemble gene orders, it avoids the analysis of predecessor and successor relationships, and directly calculates the probabilities for only a subset of adjacencies appeared in leaf nodes. However both methods are unable to handle datasets with indels and the greedy heuristic often returns an excessive number of contigs (fragments of chromosomes) when some adjacencies may have equally high probabilities but conflict each other. In the past few years, several methods had been proposed to accommodate datasets with unequal gene content [8], [9], [10]. Among them, the most recent method `GapAdj` [10] uses another scoring mechanism for gene adjacencies and reduces the assembly problem to an instance of TSP. To filter out less reliable adjacencies, it introduced a cutoff value to remove adjacencies with scores below it in the TSP solution. Further by considering pair of genes separated by up to a given number of genes as direct gene adjacency, contigs are iteratively combined into longer ones. Compared to `InferCars` [11], `GapAdj` produces a more correlated number of contigs to the actual number of chromosomes at the cost of accuracy. Through a natural process for the inference of ancestral gene contents described in [12], `GapAdj` also supports the analysis of unequal gene contents.

4 Algorithm Details

Given a phylogeny, our new method computes the gene content and ordering of ancestral (internal) nodes one at a time. Prior to the inference of a target ancestral node, we reroot the given phylogeny tree to the node such that it becomes the root of the new tree. The underlying rationale is that the calculation of probabilities follows a bottom-up manner and only the species in the subtree of the target node are considered, therefore rerooting can prevent loss of information. As a standard procedure, rerooting has already found use for ancestral genome reconstruction [6], [7].

After rerooting, `PMAG`$^+$ proceeds the following three steps: 1) inferring the gene content of target node to determine which genes should appear; 2) computing the probabilities of gene adjacencies; 3) forming and solving a TSP problem to place genes on chromosomes. The following subsections describe these steps in detail.

4.1 Inference of Ancestral Gene Contents

The very first step of ancestral reconstruction often involves explicitly estimating gene content in ancestral nodes, using content information from leaves. A number of approaches have been developed and most of them are similar in spirit to the Fitch-Hartigan parsimony algorithm [4], [12], [13].

For pure rearrangements, every gene observed in leaf species should also be present in all ancestors; however in the presence of gene indels, such correspondence does not hold anymore and a gene can be either present or absent in an ancestor. Therefore our inference of ancestral contents relies on viewing genes as independent characters (with binary states); we can then determine the state for every gene in the ancestor. The first step involves encoding the gene contents of leaf species into binary sequences. In particular, suppose a dataset G with N species is given and a set of n distinct genes $S = \{g_1, g_2, ..., g_n\}$ is identified

from G. For each leaf species G_i, its gene content $S_i = \{g_{i_1}, ..., g_{i_k}\}$ with $k \leq n$ can be equivalently represented by a sequence $\pi_i = \{\pi_{i_1}, \pi_{i_2}, ..., \pi_{i_n}\}$ in which each element has two states; if $g_j \in S_i$, $\pi_{i_j} = 1$, otherwise $\pi_{i_j} = 0$ for all j $(1 \leq j \leq n)$. For instance (table 1), a total of five distinct genes $\{a, b, c, d, e\}$ can be identified from two toy species $G1$ and $G2$ with gene orders $(+a, -c, +d)$ and $(+b, +a, -e)$ respectively.

Many methods are available to infer ancestral states from binary characters, including RAxML [14] for maximum likelihood and PAUP* [15]. In this study, we chose RAxML (version 7.2.8 was used to produce the results given in this paper) to conduct the inference of states. Once the probabilities of presence state, $P = \{p_1, p_2, ..., p_n\}$, for the root node are computed, the gene i belongs to the gene content of root S_{root} if $p_i \geq 0.5$, otherwise, gene i is not in S_{root}. Following this paradigm, gene contents for all ancestral nodes can be separately inferred from leaf species. Our simulation shows that this approach can estimate gene contents with less than 1% error even for very difficult datasets.

4.2 Inference the Probabilities of Ancestral Gene Adjacencies

In [7], we have presented an adjacency-based method in probabilistic framework called PMAG to calculate the probability of observing an adjacency in the target ancestral node. The method proceeds in the following three main steps.

Step 1 Each species in the dataset is screened to identify all unique gene adjacencies and telomeres. By viewing each adjacency and telomere as an independent character with binary states—presence or absence, gene orders of species can be rigorously encoded into aligned sequences of binary characters.

Step 2 The phylogeny tree is rerooted to the target ancestral node in order to take all leaf species into consideration. At the same time, the $2n$ ratio for base compositions is setup such that the rate of presence to absence transitions is roughly $2n$ times as high as the rate of transitions in the other direction under the same evolutionary distance, where n is equal to the number of genes. Such model has been successfully used for phylogeny reconstruction [16].

Step 3 The probabilities of characters states for all gene adjacencies and telomeres at the root node are computed . The marginal ancestral reconstruction approach suggested by Yang [17] for molecular data was adopted and extended to compute for t

PMAG+ reuses the three steps as described to calculate probabilities for adjacencies and telomeres. Once these probabilities are obtained, it then uses the following step to connect gene adjacencies and telomeres into contigs, from which the ancestral gene ordering can be identified.

4.3 Assembling Ancestral Adjacencies into Ancestral Gene Orders

The last step is to assemble gene adjacencies and telomere into a valid gene order, with respect to the gene content inferred from the first step. In general, higher probability of presence state implies an adjacency or telomere should be more likely to be included in the ancestor; however the decision on choosing an adjacency or telomere cannot be solely made upon its own probability as each gene can only be selected once. In PMAG, ancestral adjacencies are assembled by the greedy heuristic based on the adjacency graph proposed by *Ma et al.*. This greedy method starts from a contig with the first gene and picks its neighbor by using the adjacency with the highest probability; it then continues adding new genes until there is no more valid connection, in which case the current contig is closed and a new one will be formed. There are two issues with this approach that motivated us to replace the greedy assembler with an exact solver. First, the greedy heuristic can achieve good approximation only when the dataset is closely related in which case most vertices in the graph have only one outgoing edge. Second, the greedy heuristic tends to return an excessive number of contigs as it frequently leads itself into dead ends.

Obtaining gene orders from (conflict) adjacencies can be transformed into an instance of symmetric Traveling Salesman Problem (TSP), as shown in [10], [18]. In this case, we can transform genes into cities and adjacency probabilities into edge weights in the TSP graph. In particular, suppose for the target ancestral node I, we have identified a set of m adjacencies $A = \{a_1, a_2, ..., a_m\}$ and n telomeres $T = \{t_1, t_2, ..., t_n\}$ from leaf species. If the gene content of I has been inferred as $S_I = \{g_1, g_2, ..., g_k\}$ and the probabilities $P = \{p_{a_1}, ..., p_{a_m}, p_{t_1}, ..., p_{t_n}\}$ for each adjacency and telomere are known, we can create the TSP graph G as follows:

1) Each gene $g \in S_I$ is represented by two vertices—its head and tail, denoted as g^h and g^t respectively. Every extremity in the telomere $t \in T$ is represented by a unique vertex e_i, where $1 \leq i \leq n$. In this way, the total number of vertices in the graph is equal to $2 \times m + n$.
2) Edges between all pairs of head and tail of the same gene (g^h, g^t) are added with $-\inf$ to guarantee this connection is present in the solution. Edges are also established with $-\inf$ for all pairs of extremities (e_i, e_j) where $i \neq j$ and $1 \leq i, j \leq n$.
3) For every adjacency $(f, g) \in A$, the corresponding edge is added to G connecting f^t and g^h.

TABLE 1: Example of binary encoding on gene content.

	a	b	c	d	e
G_1	1	0	1	1	0
G_2	1	1	0	0	1

Similarly for other combination of orientations $(-f, g)$, $(f, -g)$ and $(-f, -g)$, we can add (f^h, g^h), (f^t, g^t) and (f^h, g^t) respectively.

4) For every telomere $(e_i, g) \in T$, we add an edge to G between e_i and g^h. In case of (g, e_i), an edge between g^t and e_i are added.
5) For the rest of the edges in G, we set the edge weights to $\inf$ to exclude them from the solution.

As the inferred probabilities range from 0 to 1, using them directly as edge weights may introduce undesirable impact associated with handling small float points. It is critical for TSP to have a more precise and fine-grained set of edge weights to assure the quality of its solution. The most straightforward way is to linearly correlate the edge weight with its probability, however in such case, differences of weights between adjacencies are too strong and adjacencies with smaller probabilities can hardly be considered. Therefore we decide to use the following equation to curve the probabilities into edge weights:

$$w_{(f,g)}(m) \quad = \quad \log_2(10^m \times (1 - p_{(f,g)})) \qquad (1)$$

where $(f, g) \in \{A \cup T\}$ and $p_{(f,g)}$ is the probabilities of observing (f, g). m is the sole parameter determining the shape of the curve and according to our experiments, TSP yields good results when $m = 6$.

We then utilize the power of one of the most used TSP solver `Concorde` [19] to find the optimal path which traverses every vertex once with the minimum total score. In the solution path, multiple contiguous extremities are shrank to a single one and a gene segment between two extremities is taken as a contig. Our construction of TSP topology is in spirit similar to `GapAdj`, however `GapAdj` requires additional procedures and parameters to adjust the contig number. Instead our inference of ancestral genome is uniform and directly from the solution of TSP, minimizing the risk of introducing artifacts.

5 Results

5.1 Experimental Design

To evaluate the performance of `PMAG`$^+$, we ran a series of experiments on simulated datasets under a wide variety of settings. We generated model topologies from the uniformly distributed binary trees, each with s species. An initial gene order of n distinct genes and m chromosomes was assigned at the root so it can evolve down to the leaves following the tree topology mimicking the natural process of evolution, by carrying out a set of predefined evolutionary events. We used different evolutionary rates r with 50% relative fluctuation, thus the actual number of events per edge is in the interval $\lfloor \frac{r \times n}{2}, r \times n \rfloor$. Several evolutionary events were considered—inversions, translocations and indels and each kind of event was assigned a probability to be selected during the simulation process. In this paper, we only present results with 20 genomes, each with 1000 genes and 5 chromosomes, to closely mimic bacterial genomes. The evolutionary rates r were set from 50 to 200 events, the later representing highly disturbed datasets. For each combination of evolutionary events, we simulated 10 datasets and reported averages and standard deviations.

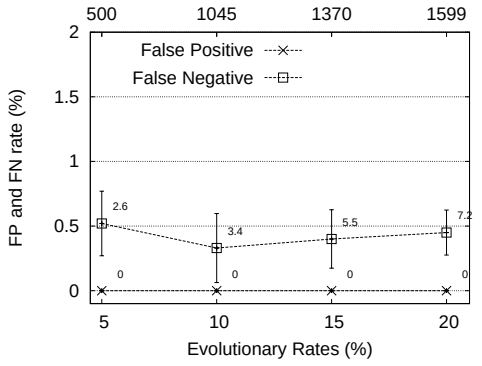
Our predicted ancestral genomes are evaluated by the ratio of correct adjacencies and telomeres recovered. In specific, we used the following equation to compute the error rate of reconstruction.

$$E = (1 - \frac{|D \cap D'|}{|D \cup D'|}) \times 100\%$$

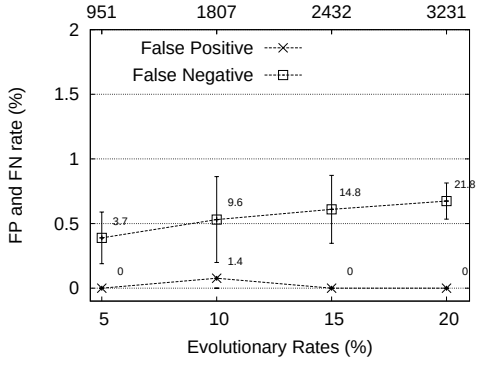
where D represents the set of gene adjacencies and telomeres in the real genome and D' the predicted genomes. We further refer an element that is contained in inferred set S' but not in true set S as a `false positive` (FP) and `false negative` (FN) is defined similarly, by swapping S and S'.

5.2 Assessing the Accuracy of Ancestral Gene Contents

We first ran simulations to test `PMAG`$^+$ on the inference of ancestral gene contents. Our gene orders, derived from its direct ancestor through a number of events, underwent random indels and inversions (two boundaries of each inversion are uniformly distributed). Two different probabilities (5% and 10%) of occurrences for indels were used. We compared our inferred gene content with its corresponding true content and counted the number of *FPs* and *FNs*. For each dataset, we summed the number of *FPs* and *FNs* in all internal nodes and divided it by the total number of genes in all ancestral nodes that are missing or inserted. Figure 1 shows our results. From this figure, the *FP* rates are always extremely low (only one dataset produced *FPs*), indicating that our inference can prevent introducing erroneous gene content and the inferred contents are reliable. *FN* rates increase slightly when more indel operations were performed, but even in the worst case the error rate stays below 1%. At the same time, we ran `GapAdj` without specifying any WGD node and set the cut-off value and maximal iterations to 0.6 and 25 as suggested. According to the results, `GapAdj` failed to



(a) 5% Gene Insertion and Deletion



(b) 10% Gene Insertion and Deletion

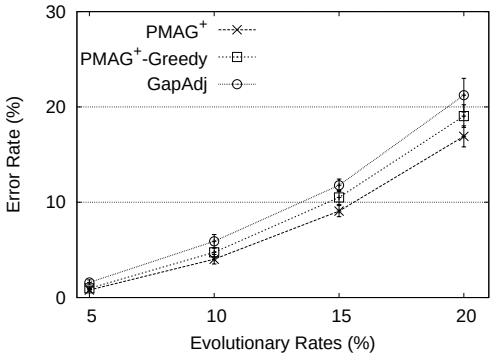
Fig. 1: *FP* and *FN* rates (divided by the numbers on upper x-axis) with standard deviations under various evolutionary rates and indel rates. Labels on upper x-axis represent the total number of genes that are inserted or deleted over all internal nodes due to indel operations. Numbers above points indicate the actual amount of errors in average.

infer a large portion of inserted genes, making the *FPs* rates in all cases higher than 60%.

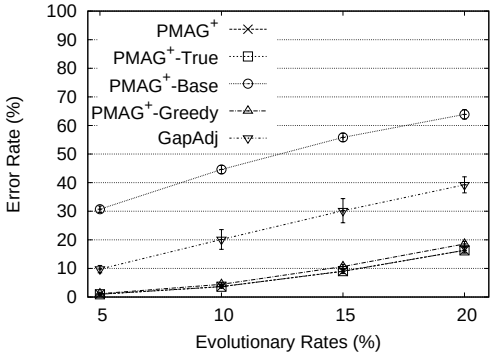
5.3 Assessing the Accuracy of Ancestral Gene Orders

We conducted several tests to evaluate the accuracy of `PMAG+` under different degrees of indels. Our first test is to compare `PMAG+` with current standard approach that reduces the dataset into equal content by elimating genes that are not present in every genome, which forms the baseline method (named `PMAG+-Base`). Our second test is to give `PMAG+` the "ground true" content (named `PMAG+-True`) to eliminate all impacts from gene contents. To compare the greedy heuristic to the TSP solution, we switched back to the greedy heuristic and redid the tests (named `PMAG+-Greedy`). Finally the results of `GapAdj` (which is the most recent method to our knowledge) were reported. To have a fair comparison, we also compared `PMAG+` with `GapAdj` using datasets without indel operations.

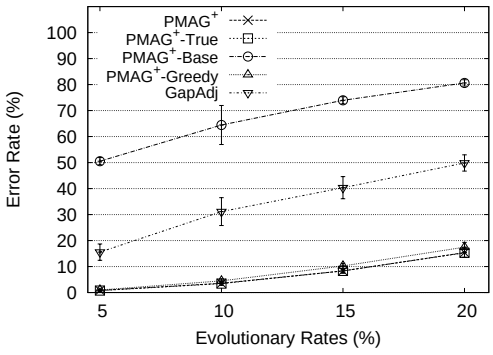
Evaluation of designed experiments in terms of error rates is shown in figure 2. From the figure, the



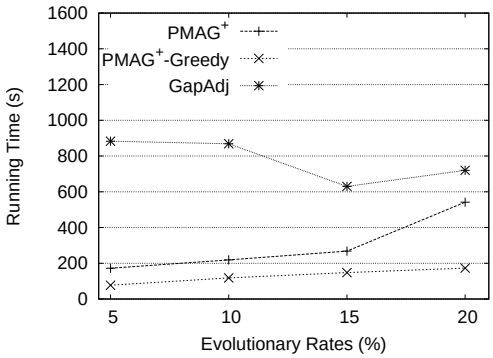
(a) 90% Inv and 10% Tsl



(b) 5% Ins and Del, 80% Inv and 10% Tsl



(c) 10% Ins and Del, 70% Inv and 10% Tsl



(d) Running time of tests in (a)

Fig. 2: (a), (b) and (c) summarize the error rates under various evolutionary rates and combinations of evolutionary events (Ins for insertion, Del for deletion, Inv for inversion and Tsl for translocation). (d) shows the running time for methods in (a). Error bars indicate the standard deviations

error rates for both PMAG$^+$ and PMAG$^+$-True are the lowest in all cases and the difference between the two approaches is almost indistinguishable, indicating that errors introduced by a very limited amount of false contents are not significant.

As expected, PMAG$^+$-Base recovered the least amount of adjacencies due to the loss of contents. GapAdj, due to its failure in gene content inference, achieved much higher error rates in the presence of indels. Even in the test of equal gene content, PMAG$^+$ can still outperform GapAdj with around 5% higher accuracy.

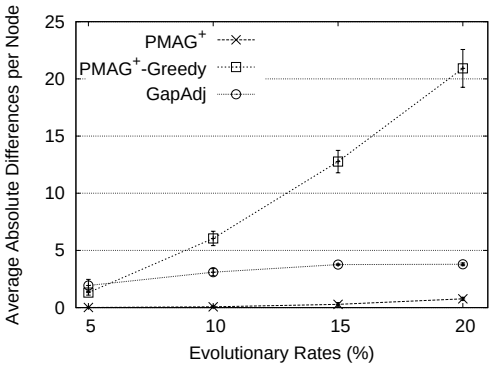
PMAG$^+$-Greedy came very close to PMAG$^+$, however in all test, PMAG$^+$ can always return more accurate reconstruction than PMAG$^+$-Greedy, suggesting the usefulness of our TSP assembler.

Using different degrees of indels has little impact on the performances of PMAG$^+$. From the perspective of adjacency evolution, an inversion operation always breaks two extant adjacencies and creates two new adjacencies, the disturbances on adjacencies introduced by an indel operation are essentially much similar to an inversion. In particular, a deletion breaks two adjacencies and creates a new one, while a insertion breaks one adjacency and introduces two new adjacencies. Therefore, as long as ancestral gene contents can be accurately predicted, PMAG$^+$ returns comparable results with all combinations of evolutionary events.

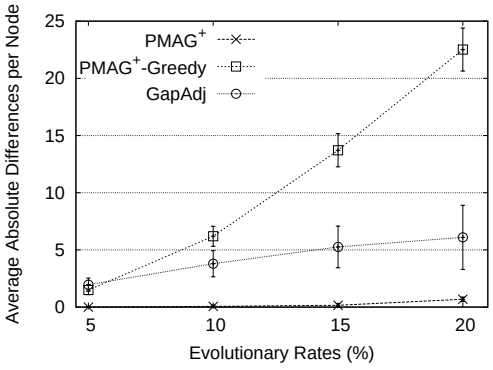
The last figure summaries the running time of all methods. From the figure, PMAG$^+$-Greedy benefits from the greedy heuristic is indeed slightly faster than PMAG$^+$, while GapAdj which solves the TSP problem heuristically took a longer time to finish than PMAG$^+$ using an exact solver.

5.4 Assessing the Number of Inferred Contigs

In [7], PMAG was tested with only unichromosomal genomes, but the inferred ancestral genomes were always composed of a large number of contigs. GapAdj designed a series of algorithms with two arguments to reconnect contigs into chromosomes with restriction of local and small evolutionary operations. Our method PMAG$^+$, on the other hand, by treating telomeres as a special type of adjacencies, simultaneously finds the best set of adjacencies and telomeres in one step. As translocation operations account for interchromosomal rearrangements which can be equivalently viewed as a fission followed by a fusion, thus all ancestors should also have the same amount of chromosomes to the root node, which is 5 in our test cases. For each dataset with N ancestors, the number of contigs c_i $(1 \leq i \leq N)$ in each ancestor was counted and the average absolute differences per ancestral node $\frac{\sum_{i=1}^{N} |c_i - 5|}{N}$ was computed to assess the accuracy of chromosomal assembly. Figure 3 summaries our findings. As predicted, the amount of contigs produced by PMAG was totally irrelevant to the true number of chromosomes, while GapAdj can indeed reduced a large portion of redundant contigs. In comparison, the number of contigs returned by PMAG$^+$ can precisely reflect the actual number of chromosomes in the true genomes.



(a) 0% Gene Insertion and Deletion



(b) 10% Gene Insertion and Deletion

Fig. 3: The average of absolute differences per ancestral node produced by various methods. Error bars indicate the standard deviations

6 Conclusions

In this study, we proposed a new adjacency-based method called PMAG$^+$ to infer the ancestral gene orders under a more general model of gene evolution, including intra-chromosomal and inter-chromosomal rearrangements as well as gene insertions and deletions. As real ancestors are unknown, we tested our method through a series of simulation studies. According to the results, PMAG$^+$ can accurately deduce the ancestral gene contents with error rates less than 1%. In the subsequent inference of ancestral gene orders, PMAG$^+$ can outperform all existing methods. Also by adopting a TSP solution for adjacency assembly, PMAG$^+$ not only overcame the issue on producing excessive contigs, but also achieved better performance than PMAG.

7 Acknowledgment

FH, JZ, LZ and JT are supported by grants US NSF #0904179 and #1161586.

REFERENCES

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

[2] B. Moret, S. Wyman, D. Bader, T. Warnow, and M. Yan: A new implementation and detailed study of breakpoint analysis. In *Proc. 6th Pacific Symp. Biocomputing (PSB'01)*, 583–594, 2001.

[3] G. Bourque and P. Pevzner: Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Research*, 12, 26–36, 2002.

[4] J. Tang, B. Moret, L. Cui, and C. dePamphilis: Phylogenetic reconstruction from arbitrary gene-order data. In *Proc. 4th IEEE Symp. on Bioinformatics and Bioengineering (BIBE'04)*, 592–599, 2004.

[5] Y. Zhang, F. Hu and J. Tang: Phylogenetic reconstruction with gene rearrangements and gene losses. 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM'10), 35–38, 2010.

[6] J. Ma A probabilistic framework for inferring ancestral genomic orders 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM'10), 179–184, 2010.

[7] F. Hu, L. Zhou and J. Tang: Reconstructing Ancestral Genomic Orders Using Binary Encoding and Probabilistic Models 9th International Symposium on Bioinformatics Research and Applications (ISBRA), 17–27, 2013.

[8] J. Ma, A. Ratan, B. Raney, B. Suh, W. Miller and D. Haussler: The infinite sites model of genome evolution. Proceedings of the National Academy of Sciences 105 (38): 14254–14261, 2008.

[9] S. Berard, C. Gallien, B. Boussau, G. Szollosi, V. Daubin and E. Tannier: Evolution of gene neighborhoods within reconciled phylogenies. Bioinformatics 28 (18): i382-i388, 2012.

[10] Y. Gagnon, M. Blanchette and N. El-Mabrouk: A flexible ancestral genome reconstruction method based on gapped adjacencies. BMC bioinformatics, 13 (Suppl 19): S4, 2012.

[11] J. Ma, L. Zhang, B. Suh, B. Raney, R. Burhans, W. Kent, M. Blanchette, D. Haussler and W. Miller: Reconstructing contiguous regions of an ancestral genome. Genome Research 16 (12): 1557-1565, 2006.

[12] J. Gordon, K. Byrne, and K. Wolfe: Additions, losses, and rearrangements on the evolutionary route from a reconstructed ancestor to the modern Saccharomyces cerevisiae genome. PLoS Genetics 5 (5): e1000485, 2009.

[13] V. Kunin and C. Ouzounis: GeneTRACE: reconstruction of gene content of ancestral species. Bioinformatics 19 (11): 1412-1416, 2003.

[14] A. Stamatakis: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics, 22 (21):2688-2690, 2006.

[15] D. Swofford David: PAUP*. Phylogenetic Analysis Using Parsimony (* and Other Methods). Version 4. (2003).

[16] Y. Lin, F. Hu, J. Tang and B. Moret: Maximum Likelihood Phylogenetic Reconstruction From High-resolution Whole-genome Data And A Tree Of 68 Eukaryotes Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing (PSB'13) 285–296, 2013.

[17] Z. Yang, K. Sudhir K and N. Masatoshi: A new method of inference of ancestral nucleotide and amino acid sequences. Genetics 1995, 141(4):1641-1650.

[18] J. Tang and L.S. Wang: Improving Genome Rearrangement Phylogeny Using Sequence-Style Parsimony. Proc. 5th IEEE Symp. on Bioinformatics and Bioengineering (BIBE'05), 137–144, 2005.

[19] D. Applegate, R. Bixby, V. Chvatal and W. Cook: Concorde TSP solver. URL: http://www.math.uwaterloo.ca/tsp/concorde/ (2011).

Fei Hu received his bachelor degree in biomedical engineering at the HuaZhong University of Science and Technology. His research interests is mainly on the phylogenetic reconstruction and inference of ancestral genomes using gene-order data.

Jun Zhou completed his bachelor degre in Biotechnology in 2008, at NanJing University, China. He had his first contact with bioinformatics in 2012, when he started working in computer science department on ancestral genome information referring project. He is currently a Ph.D. student at the computer science department, University of South Carolina, studying the small phylogeny problem.

Lingxi Zhou is a Ph.D. candidate in computer science and engineering, supervised by Dr. Jijun Tang at the bioinformatics lab of the University of South Carolina. Before that, he got his B.S. degree at the college of computer science and technology of Jilin University in July, 2011.

Jijun Tang obtained his Ph.D. from University of New Mexico in 2004. He is now an associate professor in Computer Science and Engineering, University of South Carolina, USA. He is also an adjunct professor in School of Computer Science and Technology, Tianjin University, China. His main research area is computational biology, with focus on algorithm development in phylogenetic reconstruction from genome rearrangement data.