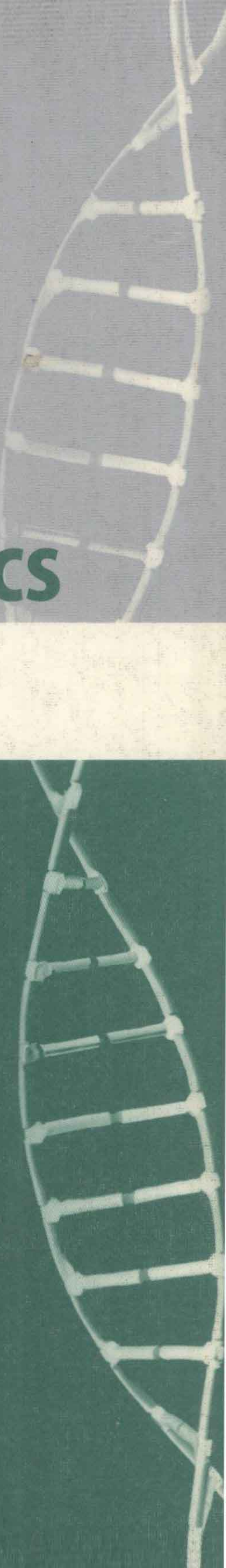
Inge Jonassen
Junhyong Kim (Eds.)

LNBI 3240

Algorithms in Bioinformatics

4th International Workshop, WABI 2004
Bergen, Norway, September 2004
Proceedings



Springer

Inge Jonassen Junhyong Kim (Eds.)

Algorithms in Bioinformatics

4th International Workshop, WABI 2004
Bergen, Norway, September 17-21, 2004
Proceedings

Springer

Series Editors

Sorin Istrail, Celera Genomics, Applied Biosystems, Rockville, MD, USA
Pavel Pevzner, University of California, San Diego, CA, USA
Michael Waterman, University of Southern California, Los Angeles, CA, USA

Volume Editors

Inge Jonassen
University of Bergen
Department of Informatics and Computational Biology Unit, HIB
5020 Bergen, Norway
E-mail: Inge.Jonassen@ii.uib.no

Junhyong Kim
University of Pennsylvania
Goddard Laboratories, Biology Dept., School of Arts and Sciences
415 S University Avenue, Philadelphia, PA 19104, USA
E-mail: junhyong@sas.upenn.edu

Library of Congress Control Number: 2004111461

CR Subject Classification (1998): F.1, F.2.2, E.1, G.1, G.2, G.3, J.3

ISSN 0302-9743
ISBN 3-540-23018-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11319306 06/3142 5 4 3 2 1 0

Lecture Notes in Bioinformatics 3240

Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Preface

It gives us great pleasure to present the proceedings of the 4th Workshop on Algorithms in Bioinformatics (WABI 2004) which took place in Bergen, Norway, September 17–21, 2004. The WABI 2004 workshop was part of a five-conference meeting, which in addition to WABI, included ESA, WAOA, IWPEC, and ATMOS, hosted by the University of Bergen, Norway. See http://www.ii.uib.no/algo2004/ for more details.

The Workshop on Algorithms in Bioinformatics covers research on all aspects of algorithmic work in bioinformatics. The emphasis is on discrete algorithms that address important problems in molecular biology. These are founded on sound models, are computationally efficient, and have been implemented and tested in simulations and on real datasets. The goal is to present recent research results, including signficant work in progress, and to identify and explore directions of future research.

Original research papers (including significant work in progress) or state-of-the-art surveys were solicited on all aspects of algorithms in bioinformatics, including, but not limited to: exact and approximate algorithms for genomics, genetics, sequence analysis, gene and signal recognition, alignment, molecular evolution, phylogenetics, structure determination or prediction, gene expression and gene networks, proteomics, functional genomics, and drug design.

We received 117 submissions in response to our call for papers, and were able to accept 39 of these. In addition, WABI hosted one invited distinguished lecture, given to the entire ALGO 2004 conference, by Dr. Marie France Sagot of the INRIA Rhône-Alpes laboratories in France.

We would like to sincerely thank all the authors of sumbitted papers, and the participants of the workshop. We also thank the program committee for their hard work in reviewing and selecting the papers for the workshop. We were fortunate to have on the program committee the following distinguished group of researchers:

Pankaj Kumar Agarwal (Duke University)
Amihood Amir (Bar-Ilan University)
Alberto Apostolico (Purdue University)
Gary Benson (MSSN, New York)
Alvis Brazma (EMBL-EBI, UK)
Olivier Gascuel (LIRMM, Montpelier)
Raffaele Giancarlo (University of Palermo)
David Gilbert (University of Glasgow)
Jan Gorodkin (KVL, Denmark)
Roderic Guigo (University of Pompeu Fabra)
Jacques van Helden (Université Libre de Bruxelles)
Daniel Huson (University of Tubingen)
Gregory Kucherov (Loria, France)
Nadia El-Mabrouk (University of Montreal)
Inge Jonassen (University of Bergen)

Junhyong Kim (University of Pennsylvania)
Jens Lagergren (KTH, Sweden)
Gad M. Landau (University of Haifa)
Thierry Lecroq (Université de Rouen)
Bernard Moret (University of New Mexico)
Vincent Moulton (University of Uppsala)
Roderic D.M. Page (University of Glasgow)
David Sankoff (University of Ottawa)
Joao Carlos Setubal (Virginia Polytechnic Institute)
Jens Stoye (University of Bielefeld)
Esko Ukkonen (University of Helsinki)
Lisa Vawter (Aventis Inc., USA)
Jaak Vilo (Egeen Inc., Estonia)
Alfonso Valencia (CNB-CSIC, Spain)
Martin Vingron (Max Planck Inst., Berlin)
Tandy Warnow (University of Texas)
Peter S. White (University of Pennsylvania)
Louxin Zhang (National University of Singapore)

We would also like to thank the co-reviewers who assisted the program committee members in their work: Ali Al-Shahib, Lars Arvestad, Vineet Bafna, Vikas Bansal, Ugo Bastolla, Ann-Charlotte Berglund, Allister Bernard, Laurent Brehelin, Dave Bryant, Trond Hellem Bø, Sergio Carvalho Jr., Robert Castel, Sergi Castellano, Benny Chor, Richard Christen, Matteo Comin, Richard Coulson, Eivind Coward, Miklós Csürös, Tobias Dezulian, Bjarte Dysvik, Ingvar Eidhammer, Isaac Elias, Eduardo Eyras, Pierre Flener, Kristian Flikka, Eva Freyhult, Ganesh Ganapathy, Clemens Groepl, Stefan Grunewald, Yann Guermeur, Michael Hallett, Sylvie Hamel, Chao He, Danny Hermelin, Pawel Herzyk, Matthias Höchsmann, Katharina Huber, Torgeir Hvidsten, Johan Kåhrström, Hans-Michael Kaltenbach, Michael Kaufmann, Carmel Kent, Mikko Koivisto, Tsvi Kopelowitz, Arnaud Lefebvre, Alice Lesser, Zsuzsanna Liptak, Olivier Martin, Gregor Obernosterer, Sebastian Oehm, Kimmo Palin, Kjell Petersen, Cinzia Pizzi, Mathieux Raffinot, Sven Rahmann, Pasi Rastas, Knut Reinert, Eric Rivals, Kim Roland Rasmussen, Mikhail Roytberg, Gabriella Rustici, Anastasia Samsonova, Erik Sandelin, Stefanie Scheid, Alexander Schliep, Beng Sennblad, Rileen Sinha, Örjan Svensson, Jinsong Tan, Gilleain Torrance, Aurora Torrente, Dekel Tsur, Juris Viksna, Alexey Vitreschak, Li-San Wang, Oren Weimann, R. Scott Winters, Peng Yin, Tomasz Zemojtel, Zhonglin Zhou, Michal Ziv-Ukelson.

We also would like to thank the WABI steering committee, Gary Benson, Olivier Gascuel, Raffaele Giancarlo, Roderic Guigo, Dan Gusfield, Bernard Moret, and Roderic Page, for inviting us to co-chair this program committee, and for their help in carrying out that task.

Finally we are grateful to Ole Arntzen at the Dept. of Informatics, University of Bergen, who helped with technical issues.

July 2004

Inge Jonassen and Junhyong Kim
WABI 2004 Program Co-chairs

Lecture Notes in Bioinformatics

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004.

Vol. 2812: G. Benson, R.D.M. Page (Eds.), Algorithms in Bioinformatics. X, 528 pages. 2003.

Vol. 2666: C. Guerra, S. Istrail (Eds.), Mathematical Methods for Protein Structure Analysis and Design. XI, 157 pages. 2003.

Table of Contents

Papers

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]

[1] Dept. of Computer Science, Univ. of New Mexico, Albuquerque, NM 87131, USA
{joeled,moret}@cs.unm.edu
[2] Dept. of Ecology and Evolutionary Biology, Univ. of Arizona, Tucson, AZ 85721, USA
lerat@email.arizona.edu

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; these new data have proved useful in shedding light on these relationships [1–4]. The order and the orientation in which genes lie on a chromosome change very slowly, in evolutionary terms, and thus together provide a rich source of information for reconstructing phylogenies. Until recently, however, algorithms using such data required that all genomes have identical gene content with no duplications, restricting applications to very simple genomes (such as organelles) or forcing researchers to reduce their data by equalizing the gene content (deleting all genes not present in every genome and all "copies" of each gene, e.g., using the *exemplar* strategy [5]). 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 [6].

Our group recently developed a method to compute the distance between two nearly arbitrary genomes [7, 8] and another to reconstruct phylogenies based on gene-content and gene-order in the presence of mildly unequal gene content [6]. In this paper, we bring together these methods in a framework that enables us to reconstruct the genomes

I. Jonassen and J. Kim (Eds.): WABI 2004, LNBI 3240, pp. 1–13, 2004.

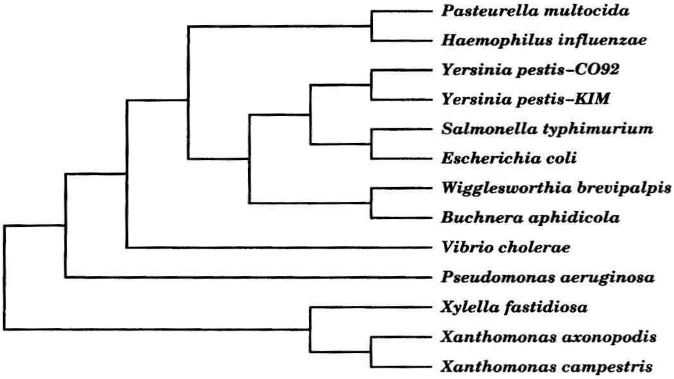


Fig. 1. The 13 gamma-proteobacteria and their reference phylogeny [9].

of the common ancestors of the 13 modern bacteria shown in Fig. 1 (from [9]). Gamma-proteobacteria are an ancient group of bacteria, at least 500 million years old [10]; the group includes endosymbiotic, commensal, and pathogenic species, with many species playing an important medical or economic role. The evolutionary history of the group is quite complex, including high levels of horizontal gene transfer [11–13] and, in the case of *B. aphidicola* and *W. brevipalpis*, massive levels of gene loss. These factors make a phylogenetic analysis of this group both interesting and challenging.

The rest of this paper is organized as follows. Section 2 presents the problem. 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.

2 The Problem

We phrase the reconstruction problem in terms of a parsimony criterion:

> 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 in terms of the number of evolutionary events (permissible operations) needed to transform the genome at one end of the edge into the genome at the other end. The permissible operations in our case are inversions, insertions (and duplications), and deletions; all operations are given the same cost in computing edge lengths. Restricting rearrangements to inversions follows from findings that the inversion phylogeny is robust even when other rearrangements, such as transpositions, are used in creating the data [14]. Our assignment of unit costs to all operations simply reflects 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 – this is the main optimization performed by our software suite GRAPPA [15]. However, even the most recent version of GRAPPA [6] is limited to relatively small genomes (typically of organellar size, with fewer than 200 genes), with modestly unequal content and just a few duplications. 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 3% to 30% of the genome. Thus, in our model, most pairwise genomic distances are very large: a simple pairwise comparison along the tree of Fig. 1 indicates that some edges of the tree must represent at least 300 events. Such lengths are at least an order of magnitude larger than GRAPPA can handle. The large genome size, vastly unequal gene content, large number of duplications, and large edge lengths all combine to make this dataset orders of magnitude more difficult to analyze than previously analyzed genome sets.

3 Prior Work

A thorough recent review of the current work in phylogenetic reconstruction based on gene content and gene order appears in [16]; we review only the relevant points here.

The GRAPPA software package [17] computes internal labels in two phases. First, it initializes internal labels of the tree by some method. Then it iteratively refines labels until convergence: each newly labeled (or relabeled) node is pushed on a queue and, while the queue is not empty, the node at the head of the queue is removed, a new label computed for it (by computing the median of its three neighbors), and, if the new label reduces the total distance to the three neighbors, the existing label is replaced with the improved label and the three neighbors are placed on the queue. Thus GRAPPA relies on the computation of the *median* of three genomes, that is, a fourth genome which minimizes the sum of the number of operations needed to convert it into each of the three given genomes. GRAPPA finds optimal inversion medians with an algorithm that runs in worst-case exponential time, but finishes quickly when the edge lengths are small (10 to 40 operations per edge) [6, 18]. GRAPPA treats groups of genes that occur in the same order and orientation in all genomes as a single genetic unit; this *condensation* step reduces computational costs and does not affect the final result [19].

Our group developed a method to find the distance between two genomes with arbitrary gene content [7, 8]; this method relies on 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. Thus arbitrary genomes are converted into duplication-free genomes. We proved that, given two genomes with unequal gene content and no duplications, any optimal sorting sequence can be rearranged to contain first all insertions, then all inversions, and finally all deletions – a type of *normal form* for edit sequences [7]. (Deletions here are genes unique to the source genome, while insertions are genes found only in the target genome.) Using the genomes produced by the duplication-renaming method, an optimal inversion sequence can be calculated in time quadratic in the size of the consensus genomes [20, 21]. The number of deletions is calculated by counting the number of Hannenhalli-Pevzner cycles that contain deletions, as described in [22]. 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 ones, 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 [23]. In gene-order data, an operon appears as a cluster of gene numbers with the same sign, with content, but not order, preserved across genomes. Heber and Stoye developed a linear-time *cluster-finding* algorithm to identify these operon-like clusters within equal-content genomes [24].

McLysaght *et al.* [4] reconstructed ancestral genomes for a group of poxviruses; she determined gene content by assuming that the phylogenetic tree contained a single point of origin for each gene family in the modern genomes. Each point of origin was assigned to that internal node which minimized the number of loss events necessary to achieve the gene content of the leaf genomes.

4 Designing an Algorithmic Framework

To address the problem of reconstructing ancestral genomes at the level of complexity of gamma-proteobacteria, we use condensation of gene clusters in order to reduce the size of the genomes, describe a procedure similar to that of McLysaght *et al.* to determine the gene content of every internal node, and present a new heuristic to compute the median of three very different genomes.

4.1 Medians

We use the queue-based tree-labeling heuristic described in Section 3. Since leaves contain the only labels known to be correct, we update the nodes in order of their distance from the leaves, as shown in Fig. 2. The heart of the top-level heuristic is the median computation. Exact median-finding algorithms are limited to small genomes, small edge lengths in the tree, and few changes in content – and none of these properties holds in our problem. We therefore pursue a simple heuristic inspired by geometry. The median of a triangle in the plane can be found by drawing a line from one vertex to the middle of the opposite segment, then moving two thirds of the way along this line. By analogy, we generate a sorting sequence from one genome to another (an edge of the triangle),

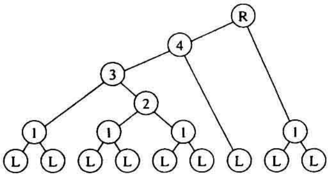


Fig. 2. Internal nodes ordered by their distance from the leaves. Nodes with lower indices will be labeled first; no label is generated for the root.

then choose a genome halfway along this sorting sequence and generate a new sorting sequence from it to the third genome, stopping one-third along the way.

We extend the method of Marron *et al.* [7] to enumerate all possible positions, orientations, and orderings of genes after each operation. 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 lie in the target genome. We can thus generate the genomes produced by "running" a portion of the sorting sequence, then use these intermediate genomes for the median heuristic just described, all in polynomial time.

This handling of inserted genes leads to an *overestimate* of the edit distance, which Marron *et al.* showed at most doubles the number of operations [7]. Their original method calculates all possible positions in the source genome to which the inversion sequence could move insertions and chooses the final position (for each insertion) to minimize the number of groups of inserted genes; it may *underestimate* the edit distance because the grouping of inserted genes may require an inversion to join inserted genes and simultaneously split deleted genes, which is not possible. We compared pairwise distances produced by our method and by theirs to get an upper bound on the overestimation: 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 before computing any median: once the gene content of an internal node is assigned, it remains unchanged. Since the tree is rooted, we know the direction of time flow on each tree edge; we also assume that deletions are far more likely than insertions, The number of copies of each gene g is decided independently of all other genes; at internal node i, it 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 that contain g; or (ii) there are leaves containing 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. 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. We use a second depth-first traversal to set the actual number of copies of each gene at each internal node. If

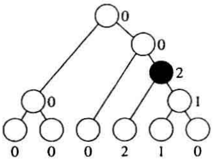


Fig. 3. The number of copies of a gene in internal nodes.

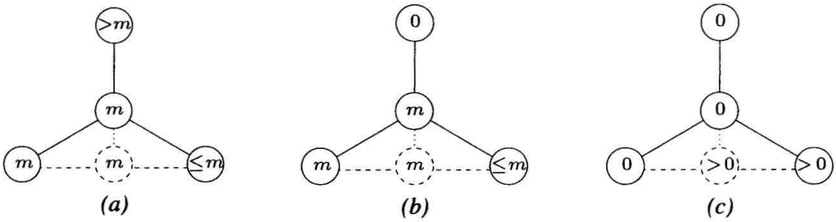


Fig. 4. Cases where the median and its neighbors have different numbers of copies of a gene. Solid lines are tree edges; dashed and dotted lines are fractions ($\frac{1}{2}$ and $\frac{1}{3}$, resp.) of sorting sequences.

either of the root's children has a subtree maximum 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 maximum 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.

Internal nodes will thus 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 do. (We consider the two children of the root to be neighbors.) Moreover, if a node is the nearest common ancestor of all genomes possessing the gene, it may have more copies of the gene than its parent and one of its children, as in the case of the black node in Fig. 3. The gene content of intermediate genomes along sorting sequences is a union of the gene contents 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, we face three cases in which the number of copies of a gene differ between the intermediate genome, the median genome, and the median's parent – see Fig. 4. In Fig. 4a, the intermediate genome has the same number of copies as the median, but fewer than the parent, as with the black node's right child in Fig. 3. Each copy in the parent that is not matched by the duplication-renaming algorithm will be excluded from the median genome. The case of 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 Fig. 3. Genomes along the intermediate sequence have the same number of copies as the median, while the parent of the median contains no copy at all. Finally, the case of Fig. 4c can only arise when the right child of the median is the nearest common ancestor of all genomes containing the gene, as with the parent of the black node in Fig. 3.

Biologically, this process of finding which duplicates to include in the median corresponds to matching orthologous duplicates of each gene between genomes and to discarding 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 extract information from larger and more complex biological datasets, we need fast algorithms with fast implementations; faster processing also enables a more thorough analysis and thus produces results of higher quality. The key factor here is the size of the genomes – their number is a much smaller issue. We thus developed a technique to identify and condense gene clusters in order to reduce the size of the genomes. Our approach generalizes that used in genomes with equal content [24]; in contrast, GRAPPA only condenses identical subsequences of genes, because it aims to preserve the identity of edit sequences. Our method allows the condensation of clusters based only on content (not order, at least as long as genes stay on the same strand) and also handles the difficult cases that arise out of unequal gene content (such as an insertion within 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 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 [24] 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 one of 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 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 replaced with the cluster's median. At this point, if any extra genes moved to the side of the cluster are still beside it, they are moved back inside the cluster to a position similar to their original one.

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 the inversion portion of 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 Fig. 4 and