Phylogenetics

webMGR: an online tool for the multiple genome rearrangement problem

Chi Ho Lin1, Hao Zhao1, Sean Harry Lowcay2, Atif Shahab1, and Guillaume Bourque1,∗

1Genome Institute of Singapore, 60 Biopolis Street, #02-01, Genome, Singapore 138672 and 2Experimental Therapeutics Centre, 31 Biopolis Way, Nanos Level 3, Singapore 138669

Received on September 18, 2009; revised on November 30, 2009; accepted on December 9, 2009

Advance Access publication December 18, 2009

Associate Editor: Alex Bateman

ABSTRACT

Summary: The algorithm MGR enables the reconstruction of rearrangement phylogenies based on gene or synteny block order in multiple genomes. Although MGR has been successfully applied to study the evolution of different sets of species, its utilization has been hampered by the prohibitive running time for some applications. In the current work, we have designed new heuristics that significantly speed up the tool without compromising its accuracy. Moreover, we have developed a web server (webMGR) that includes elaborate web output to facilitate navigation through the results.

Availability: webMGR can be accessed via http://www.gis.a-star.edu.sg/~bourque. The source code of the improved standalone version of MGR is also freely available from the web site.

Contact: bourque@gis.a-star.edu.sg

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

In recent years, the continuous flow of new genome sequences has enabled a number of rearrangement studies that have looked at the large-scale genomic organization of various species. An important and challenging problem associated with these studies is the Multiple Genome Rearrangement Problem. This problem aims to recover the phylogeny T and the most parsimonious rearrangement scenario that can explain a set of extant genomes. Formally, the problem can be described as follows: given the block orders of a set of m genomes, each represented by a signed permutation, the goal is to reconstruct an unrooted tree T and assign permutations to the internal nodes (ancestors), such that the sum of the distances d(A,B) over all the edges of T is minimized. Here, d(A,B) is the genomic distance on an edge (A,B) which is defined as the minimum number of rearrangement operations necessary to transform genome A into genome B. When m = 3, this problem is called the Median Problem. When the genomes are uni-chromosomal, the operations permitted are restricted to reversals (also called inversions). Such operations invert the order and the orientation of a set of consecutive blocks. When the genomes are multi-chromosomal, reversals, translocations, fissions/fusions are considered.

∗To whom correspondence should be addressed.

Although the genomic distance between two genomes can be computed efficiently (Bader and Moret, 2001; Tesler, 2002a), the multiple genome rearrangement problem is computationally challenging. Specifically, it was proved that the median problem with reversals is NP hard (non-deterministic polynomial-time hard; Caprara, 1999). In Bourque and Pevzner (2002), the algorithm MGR was proposed to tackle the median problem using heuristics. When m > 3, MGR handles the problem by iteratively solving a sequence of median problems. MGR has been applied to reconstruct rearrangement evolutionary scenarios for various sets of genomes (Lin et al., 2008; Murphy et al., 2005; Xia et al., 2007) and these reconstructions have shed new light on the forces shaping the genome of these species.

MGR solves the median problem by iteratively performing rearrangement events that bring the extant genomes closer to each other. In the current work, we present two complementary strategies that can greatly reduce the running time of the algorithm without compromising the quality of the solutions obtained. The first improvement involves the systematic ‘condensing’ and ‘uncondensing’ of the permutations being evaluated, whereas the second improvement involves simplifying rules in the selection of the rearrangement to be performed. Using simulations we show that these improvements greatly enhance the efficiency of MGR without sacrificing the reconstruction quality.

Finally, to facilitate user accessibility to this algorithm, we describe a web server version of MGR so that users can submit reconstruction requests directly. The web server also facilitates interactive navigation through the results.

2 IMPROVEMENTS ON MGR

We first describe the original heuristics that were implemented in MGR to solve the median problem. Given three uni-chromosomal genomes G1, G2 and G3, MGR searches and iteratively performs reversals in the starting genomes until they all converge to a common ancestor. At every step, the reversal that is performed is selected from a list of good reversals where a reversal r acting on Gk is said to be good if performing it on Gk reduces the distance to both G2 and G3. The idea behind this is that a reversal that brings G1 closer to both G2 and G3 is also likely to bring it closer to the unknown ancestor A. At any given step, there is often a variety of good reversals to choose from. For such cases, the original heuristic that was implemented involved calculating, for every reversal r, the
instance, if in the exact same order in all the genomes being considered. For strips. A
reversal to evaluate is correlated with the size of the permutations. slow for large genomes because the number of candidate good
reversals to evaluate is correlated with the size of the permutations. The first improvement we proposed is based on the concept of
strips. A strip is a maximal segment of two or more blocks that are in the exact same order in all the genomes being considered. For instance, if $G_1 = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$, $G_2 = -3\ -2\ -1\ 4\ 5\ 6\ 7\ 8\ 9$ and $G_3 = 1\ 2\ 3\ 7\ 8\ 9\ -6\ -5\ -4$, there will be three strips: $1\ 2\ 3\ 4\ 5\ 6$ and $7\ 8\ 9$. These strips display coherent gene orders and it has been shown that there exists an optimal phylogeny tree $T$ of
the genomes such that every strip appears in the ancestral nodes of $T$ (Tesler, 2002b). Intuitively, any good reversal will not break these
strips, thus allowing us to replace the strips by single blocks in a new alphabet. In the example above, $G_1$, $G_2$ and $G_3$ will be ‘condensed’ and transformed into $G'_1 = 1\ 2\ ' 3\ ' 4\ ' 5\ ' 6\ ' 7\ ' 8\ ' 9$ and $G'_2 = 1\ 3\ ' 2\ ' 4\ ' 5\ ' 6\ ' 7\ ' 8\ ' 9$. This will have a significant impact on the running
time of MGR because the size of the genomes being tracked will
be greatly reduced. The strips will also be uncondensed before the
output is generated, so the whole process will actually be seamless.

In the second improvement, we propose two heuristics $H_1$ and $H_2$
to simplify the rule of selecting a good reversal at a given step when we have a list of such reversals: ($H_1$) pick the first good reversal, or ($H_2$) based on a simple assumption that short reversals represent a more common evolutionary event as compared with translocation, fusions and fissions (Sankoff, 2002), we select the ‘shortest’ good reversal. Note that in both of these heuristics we no longer need to
calculate $n_r$ for all good reversals $r$.

When the input genomes are multi-chromosomal, the good
rearrangements that are selected and performed also include translocations and fissions/fusions. Based on an observation that reversals are more common than the other operations in the evolution, at each step we always select and perform good reversals first, and then good translocations and fusions/fissions if we run out of
good reversals. This is done as in Bourque and Pevzner (2002).

We use simulated trees with 7 genomes and 100 genes to test the
impact of these ideas on the running time and the reconstruction
accuracy. The evolutionary rate $m$ (average number of events on
an edge) for the trees varies from 2.5 to 12.5 and the trees are generated as in Zhao and Bourque (2009) (see Supplementary
Materials). Figure 1A shows that the condensing strategy and the
two heuristics significantly reduce the running time. Since we seek
to reconstruct a most parsimonious tree, we use the total number of
predicted events to measure the reconstruction quality. Figure 1B shows that MGR recovers trees with sometimes even fewer events as compared with the real trees (especially when $m > 2.5$) and that the
improved MGR returns trees with nearly identical number of events
as before (see Supplementary Materials). Thus, our improvements on
MGR greatly enhance the efficiency without negatively affecting
the reconstruction quality.

3 MGR WEB TOOL
We have implemented these changes in MGR and built a web server
called webMGR using the Ruby on Rails technology. The interface
offers a number of customizable options to determine how MGR
will be executed. The web page is partitioned in three sections:
(1) the input genomes can be either uni- or multi-chromosomal;
(2) options to condense the genomes, to use either of the two
heuristics ($H_1$ and $H_2$) and to decide if the reconstruction
should be restricted to a fixed phylogenetic tree;
(3) output options.

The output from webMGR is provided in html format and includes:
(1) ASCII representation of the unrooted tree recovered with
clickable edges to display the possible optimal rearrangement
events via the GRIMM web server (Tesler, 2002a).
(2) Pairwise distance matrix of the input genome and offers
clickable pairwise scenarios.
(3) Signed permutations of the input genome as well as the
ancestors recovered.

To illustrate the use of webMGR, we have applied it to the 2435 gene
orthologs from Lin et al. (2008) (see Supplementary
Materials).

ACKNOWLEDGEMENTS
We thank Glenn Tesler for his help with both GRIMM and MGR.

Funding: Agency for Science, Technology and Research of Singapore.

Conflict of Interest: none declared.

REFERENCES
between signed permutations with an experimental study. In Proceedings of the 7th
International Workshop on Algorithms and Data Structures (WADS2001), Springer
Berlin/Heidelberg, Providence, RI, pp. 365–376.