Jiang T. Some algorithmic challenges in genome-wide ortholog assignment. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 25(1): 1– Jan. 2010

# Some Algorithmic Challenges in Genome-Wide Ortholog Assignment

Tao Jiang (姜 涛), *Fellow*, *ACM*

*Department of Computer Science and Engineering, University of California, Riverside, CA 92521, U.S.A.*

E-mail: jiang@cs.ucr.edu

Received September 1, 2009; revised November 21, 2009.

**Abstract** Genome-scale assignment of orthologous genes is a fundamental and challenging problem in computational biology and has a wide range of applications in comparative genomics, functional genomics, and systems biology. Many methods based on sequence similarity, phylogenetic analysis, chromosomal syntenic information, and genome rearrangement have been proposed in recent years for ortholog assignment. Although these methods produce results that largely agree with each other, their results may still contain significant differences. In this article, we consider the recently proposed parsimony approach for assigning orthologs between closely related genomes based on genome rearrangement, which essentially attempts to transform one genome into another by the smallest number of genome rearrangement events including reversal, translocation, fusion, and fission, as well as gene duplication events. We will highlight some of the challenging algorithmic problems that arise in the approach including (i) minimum common substring partition, (ii) signed reversal distance with duplicates, and (iii) signed transposition distance with duplicates. The most recent progress towards the solution of these problems will be reviewed and some open questions will be posed. We will also discuss some possible extensions of the approach to the simultaneous comparison of multiple genomes.

**Keywords** algorithm, comparative genomics, computational biology, genome rearrangement, ortholog assignment

## 1 Introduction

In this article, we review the new combinatorial approach that we recently introduced for genome-wide assignment of orthologous genes between closely related species, and highlight a few algorithmic challenges. Essentially, the method, called the *parsimony approach*, assigns the orthology relationship between the genes in two input genomes so that the overall number of genome rearrangement events (i.e., reversals, translocations, fusions, and fissions) and duplication events required to transform one genome into the other is minimized. It has been implemented as a prototype ortholog assignment system, called *MSOAR*, and tested on the human and mouse genomes with very promising results. Our ongoing research attempts to further improve MSOAR and make it an accurate, high-throughput ortholog assignment system. However, many challenges faces us. In particular, our parsimony approach could benefit from efficient algorithms for several combinatorial optimization problems, including (i) signed reversal distance with duplicates (SRDD), (ii) signed transposition distance with duplicates (STDD), and (iii) minimum common substring partition (MCSP). These problems, which will be defined formally later, are all NP-hard. Efficient and effective (approximation/heuristic) algorithms for them will not only be crucial to the success of our ortholog assignment system, but may also reveal interesting combinatorial structures and new algorithmic design techniques of interest to the general algorithms and computational biology communities, due to the elegant nature of the problems and connection to well-known problems in the literature. In addition, we will consider how to extend the system MSOAR, which currently works only for two species, to multiple species.

In the following, we first give a brief introduction to the ortholog assignment problem and the existing methods for assigning orthologs in Section 2, and then describe our new parsimony approach for assigning orthologs between closely related genomes based on genome arrangement and show some promising preliminary experimental results in Section 3. We then present some algorithmic problems that are critical to the success of the parsimony approach and discuss some possible methods for solving these problems in Section 4.

## 2 Genome-Wide Assignment of Orthologous Genes

In evolutionary biology, the term *homology* refers to

Survey

This work is supported by the NSF of USA under Grant No. IIS-0711129.

the sharing of a common ancestor. Thus, homologous genes (or simply *homologs*) are those that evolved from the same ancestral gene. *Orthologs* and *paralogs*[1] are two fundamentally different types of homologs. They differ in the way that they arose: orthologs are genes that evolved by *speciation*, while paralogs are genes that evolved by *duplication*. To better describe the evolutionary process and functional diversification of genes, paralogs are further divided into two subtypes: *outparalogs*, which evolved via ancient duplications preceding a given speciation event under consideration (i.e., by pre-speciation duplications), and *inparalogs*, which evolved more recently via duplications subsequent to the speciation event[2-3] (i.e., by post-speciation duplications). For a given set of inparalogs on a genome, there commonly exists a gene that is the direct descendant of the ancestral gene of the set, namely the one that best reflects the original location of the ancestral gene in the ancestral genome. Sankoff[4] named such a gene the *true exemplar* of the inparalogous set. Given two genomes, two sets of inparalogous genes (one from each genome) are said to be *co-orthologous* if they are descendants of the same ancestral gene at the time of speciation. For two co-orthologous sets of inparalogous genes, the *main ortholog* pair is defined as the two true exemplar genes of each set. These concepts are illustrated in Fig.1.

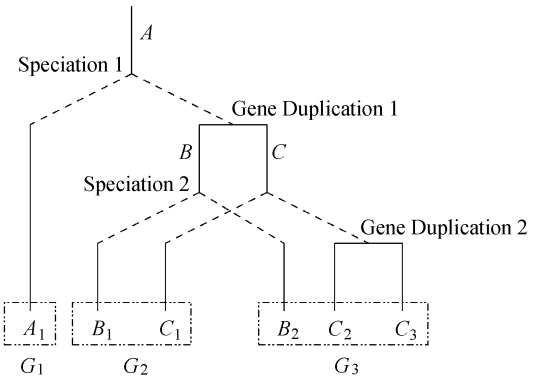


Fig.1. Illustration of orthologous and paralogous relationships. After two speciation events and two gene duplications, three present genomes, $G_1 = (A_1)$, $G_2 = (B_1, C_1)$ and $G_3 = (B_2, C_2, C_3)$ are formed. In this scenario, all genes in $G_2$ and $G_3$ are co-orthologous to gene $A_1$. Genes $B_1$ and $C_1$ are outparalogs with respect to $G_3$ (i.e., the 2nd speciation), and are inparalogs with respect to $G_1$ (i.e., the 1st speciation). Gene $C_2$ is the direct descendant (i.e., true exemplar) of the ancestral gene $C$ while $C_3$ is not, assuming that $C_3$ is duplicated from $C_2$. Genes $C_1$ and $C_2$ form a pair of main orthologs, so do $B_1$ and $B_2$.

Clearly, orthologs are evolutionary and, typically, functional counterparts in different species. Therefore, many existing computational methods for solving various biological problems, e.g., the inference of functions of new genes, analysis of phylogenetic relationship between different species, and comparative inference of biological pathways, use orthologs in a critical way. A major complication with the use of orthologs in these methods, however, is that orthology is generally not a one-to-one relationship because a single gene in one phylogenetic lineage may correspond to a whole family of inparalogs in another lineage, as illustrated in Fig.1. In practice, much caution should be taken while such one-to-many and many-to-many relationships are applied to the transfer of functional assignments between homologous genes because some inparalogs could have acquired new functions during the course of evolution. As a consequence, the identification of orthologs and inparalogs, especially the one-to-one relationship between main orthologs, is critical for evolutionary and functional genomics, and thus a fundamental problem in computational biology and genomics. Note that, main orthologs are more likely to be functional counterparts in different species, since they are both evolutionary and positional counterparts.

It follows from the definition of orthology and paralogy that the best way to identify orthologs is to measure the divergence time between homologous genes in two different genomes. As the divergence time could be estimated by comparing the DNA or protein sequences of genes, most of the existing algorithms for ortholog assignment, such as the well-known COG system[5-6] and INPARANOID program[3], rely mainly on sequence similarity (usually measured via BLAST scores[7]). An implicit, but often questionable, assumption behind these methods is that the evolutionary rates of all genes in a homologous family are equal. Incorrect ortholog assignments might be obtained if the real rates of evolution vary significantly between paralogs. On the other hand, we observe that molecular evolution proceeds in two different forms: local mutation and global rearrangement. Local mutations include base substitution, insertion and deletion, and global (genome) rearrangements include reversal, transposition, translocation, fusion, fission, and so on. Apparently, the sequence similarity-based methods for ortholog assignment make use of local mutations only and neglect genome rearrangement events that might contain valuable information. A more detailed account of recent work on ortholog assignment is given at the end of this section.

In our recent papers[8-11], we introduced a new approach for ortholog assignment between two closely related genomes that takes advantage of evolutionary evidence from both local mutations and global genome rearrangements. It begins by identifying homologous gene families on each genome and the correspondence

between families on both genomes using sequence similarity (i.e., BLAST) search. The homologs are then treated as copies of the same genes, and ortholog assignment is formulated as a natural combinatorial optimization problem of rearranging one genome consisting of a sequence of (possibly duplicated) genes into the other with the minimum number of rearrangement events, where the most parsimonious rearrangement process should suggest (main) orthologous gene pairs in a straightforward way. Spurious assignments of inparalog are then detected by using a post-processing procedure (called "noise" gene pair detection). A high-throughput system, called MSOAR, was implemented based on this approach. Our preliminary experiments on simulated and real data demonstrate that MSOAR outperforms or is at least comparable to other popular ortholog assignment methods such as Exemplar[4] and INPARANOID[3].

### 2.1 Existing Ortholog Assignment Methods and Related Work

In the past decade, many computational methods for ortholog assignment have been proposed, most of which are based primarily on sequence similarity. These methods include the COG system[5-6], EGO (previously called TOGA)[12], INPARANOID[3], and OrthoMCL[13], just to name a few. Some of these methods combine sequence similarity and a parsimony principle, such as the reconciled tree method[14] and the bootstrap tree method[15], or make use of synteny information, such as OrthoParaMap[16] and the recent method proposed by Zheng *et al.*[17]. However, none of these methods use genome rearrangement. See [18] for a recent review on bioinformatics tools for ortholog assignment.

On the other hand, there have been a few papers in the literature that study rearrangement between genomes with duplicated genes, which is closely related to ortholog assignment. Sankoff[4] proposed an approach to identifying the true exemplar gene of a gene family, by minimizing the breakpoint/reversal distance between two reduced genomes that consist of only true exemplar genes. El-Mabrouk[19] developed an approach to reconstructing the ancestor of a modern genome by minimizing the number of duplication transpositions and reversals. The work in [20-21] proposed methods that attempt to find one-to-one gene correspondence between gene families based on conserved segments. Very recently, Swenson *et al.*[22] presented some algorithmic results on the cycle splitting problem in a combinatorial framework similar to the one introduced in [8-10].

## 3 Parsimony Approach to Ortholog Assignment via Genome Rearrangement

Here, we give more details of the parsimony approach that we recently introduced in [8-10] for assigning orthologs between two closely related genomes. Suppose that the two genomes to be compared, denoted as $\Pi$ and $\Gamma$, have undergone a series of genome rearrangement and gene duplication events since they split from their last common ancestral genome. Clearly, we could easily identify the main orthologs and inparalogs if given such an evolutionary scenario. Based on this observation and the parsimony principle, ortholog assignment was posed as the problem of reconstructing an evolutionary scenario incurring the minimum number of rearrangement and duplication events in [8-10]. Equivalently, it can be formulated as a problem of finding a most parsimonious transformation from one genome into the other by genome rearrangements and gene duplications, without explicitly inferring their ancestral genome. Let $R(\Pi, \Gamma)$ and $D(\Pi, \Gamma)$ denote the number of rearrangement events and the number of gene duplications in a most parsimonious transformation, respectively, and $RD(\Pi, \Gamma)$ denotes the *rearrangement/duplication* ($RD$) *distance* between $\Pi$ and $\Gamma$ satisfying $RD(\Pi, \Gamma) = R(\Pi, \Gamma) + D(\Pi, \Gamma)$. The genome rearrangement events considered in [8-10] include *reversals*, *translocations*, *fissions*, and *fusions*. *Transpositions* will also be considered in this proposed research.

Fig.2 presents a simple example to illustrate the basic idea behind our parsimony approach. Consider two

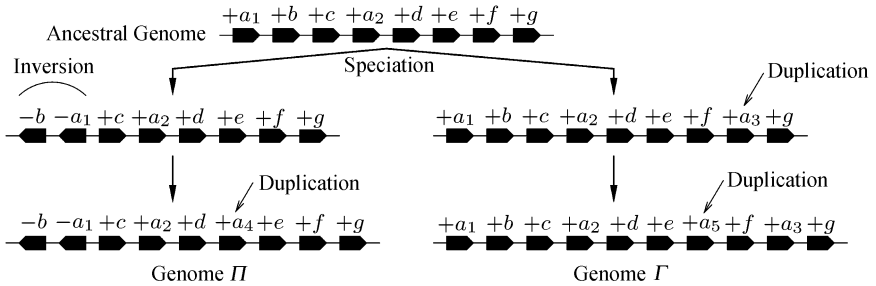


Fig.2. Evolutionary history of two genomes $\Pi$ and $\Gamma$ since the splitting from their ancestral genome. $\Pi$ evolved from the ancestor by one inversion and one gene duplication, and $\Gamma$ by two duplications.

(uni-chromosomal) genomes, $\Pi = -b - a_1 + c + a_2 + d + a_4 + e + f + g$ and $\Gamma = +a_1 + b + c + a_2 + d + e + a_5 + f + a_3 + g$, sharing a gene family $a$ with multiple copies and several "singleton" families. As shown in Fig.2, both genomes evolved from the same ancestral genome $+a+b+c+d+e+f+g$, $\Pi$ by an inversion and a gene duplication and $\Gamma$ by two gene duplications, respectively. By computing the rearrangement/duplication distance $RD(\Pi, \Gamma) = 4$, the true evolutionary scenario can be reconstructed, which then suggests that the two genes $a_1$ form a pair of main orthologs, as well as the two genes $a_2$. Meanwhile, $a_3$, $a_4$, and $a_5$ are inferred as inparalogs that were derived from duplications after the speciation event. It is interesting to see that here $a_4$ is not assigned orthology to $a_3$ or $a_5$ greedily.① This simple example illustrates that, by minimizing the rearrangement/duplication distance, our approach is able to pick correct main orthologs out of sets of inparalogs.

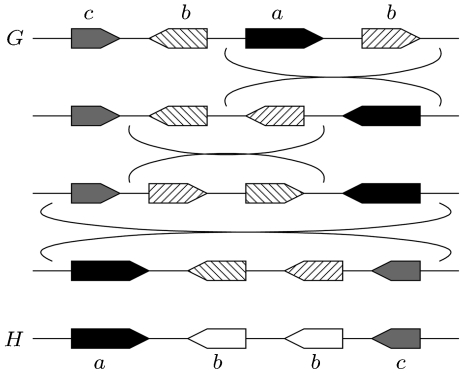


Fig.3. The most parsimonious transformation using three reversals.

When we have two unichromosomal genomes of equal gene content (i.e., each gene family has the same number of members in both genomes) and all duplication events occurred before the speciation, the evolutionary process after speciation only involves reversals and thus the above problem reduces to finding a transformation with the minimum number of reversals (namely, sorting by reversals[23]). The following example illustrates how in this (simpler) case we can assign orthologs via sorting by reversals. Consider two genomes, $G = +c - b + a + b$ and $H = +a - b - b - c$, consisting of four genes and one multi-gene family each. Fig.3 shows the parsimonious transformation from $G$ into $H$ using three reversals. In this transformation, the first (or second) copy of gene $b$ in $G$ is found to correspond to the first (or second, respectively) gene $b$ in $H$, indicating that they might be a pair of main orthologs.

The parsimony approach for ortholog assignment has been implemented as a prototype high-throughput system, called MSOAR[10-11].② An outline of MSOAR is depicted in Fig.4. After defining gene families by homology search, the system employs a 4-step heuristic algorithm to estimate the rearrangement/duplication distance between the two input genomes, which can be used to reconstruct a most (or nearly most) parsimonious evolutionary scenario. The first three steps of the heuristic try to find a transformation between the two genomes with the minimum number of rearrangements, matching as many homologous genes as possible. It then uses a post-processing step (the 4th step) to detect assigned gene pairs whose deletion (or uncoupling) would decrease the rearrangement distance by at least two.③ Such gene pairs are more likely to consist of inparalogs caused by post-speciation duplications than main orthologs and referred to as "noise" gene pairs.

We have tested MSOAR extensively on both simulated and real genomic data in [8-11]. The simulation results show that MSOAR performs equally well as an iterated version of the Exemplar algorithm of Sankoff[4] in terms of detecting inparalogs and outperforms it significantly in terms of identifying main ortholog pairs, although the way that the genomic data was simulated is a bit simplistic. So, we focus on the test results on the human and mouse genomes below. The human genome contains 20 181 protein-coding genes and the mouse contains 17 858 protein-coding genes. As shown in Table 1, before removing "noise" gene pairs, MSOAR

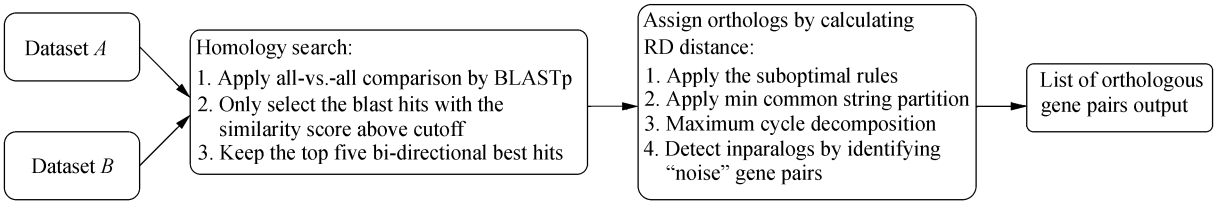


Fig.4. An outline of MSOAR.

①They are orthologs, but not main orthologs, by definition.

②The system has been recently updated with a new function to explicitly treat tandemly duplicated genes[24].

③Note that, in this case, the overall rearrangement/duplication distance will not increase since the deletion of a gene pair may only increase the number of duplications required in an optimal scenario by two.

assigned 13 395 main orthologs pairs. Then MSOAR removed 177 "noise" gene pairs and output 13 218 main orthologs pairs. The detailed result can be found at website http://msoar.cs.ucr.edu/.

We have validated MSOAR's assignment by using gene annotation, in particular, gene names extracted from UniProt[25] release 6.0 (September 2005). The total number of *assignable* pairs of orthologs, i.e., pairs of genes with identical (known) names, between human and mouse is 9891. Among the 13 218 (main) ortholog pairs that MSOAR predicted, 9214 are true positives, 2126 involve unknown genes, and 1978 are false positives, resulting in a sensitivity of 93.16% and a specificity of 83.07%.

The detailed comparison result between MSOAR and INPARANOID[3] is shown in Table 1. MSOAR was able to identify 99 more true ortholog pairs than INPARANOID, although it also reported more false positives. Fig.5 illustrates two examples where INPARANOID failed to assign main ortholog pairs correctly because the two genes in each pair are not bidirectional best hits of each other, but MSOAR successfully resolved them.

The HGNC Comparison of Orthology Predictions (HCOP)[26] is a tool maintained by HUGO Gene Nomenclature Committee that integrates and displays the human-mouse orthology assertions made by six methods including Ensembl, Homologene, INPARANOID, PhIGS, MGD and HGNC, some of which involve human expert curation. The following Fig.6 illustrates that the main ortholog pairs predicted by MSOAR are supported by most of these methods.

We have also validated MSOAR's assignment by considering protein functions as defined in the Panther database of Applied Biosystems[27]. Among the 13 218 ortholog pairs assigned by MSOAR, 10 800 pairs have both orthologous genes in the same protein subfamily. Finally, MSOAR's result is quite consistent with Jackson Lab's human-mouse ortholog database[28]; 9603 of the ortholog pairs predicted by MSOAR can be found in Jackson Lab's human-mouse ortholog database.

**Table 1.** Comparison of Ortholog Assignments Between MSOAR and INPARANOID

| | Assignable Orthologs | Assigned | True Positive | Unknown Pairs |
|---|---|---|---|---|
| MSOAR (before removing "noise" pairs) | 9 891 | 13 395 | 9 263 | 2 177 |
| MSOAR (after removing "noise" pairs) | 9 891 | 13 218 | 9 214 | 2 126 |
| INPARANOID | 9 891 | 12 758 | 9 115 | 2 034 |

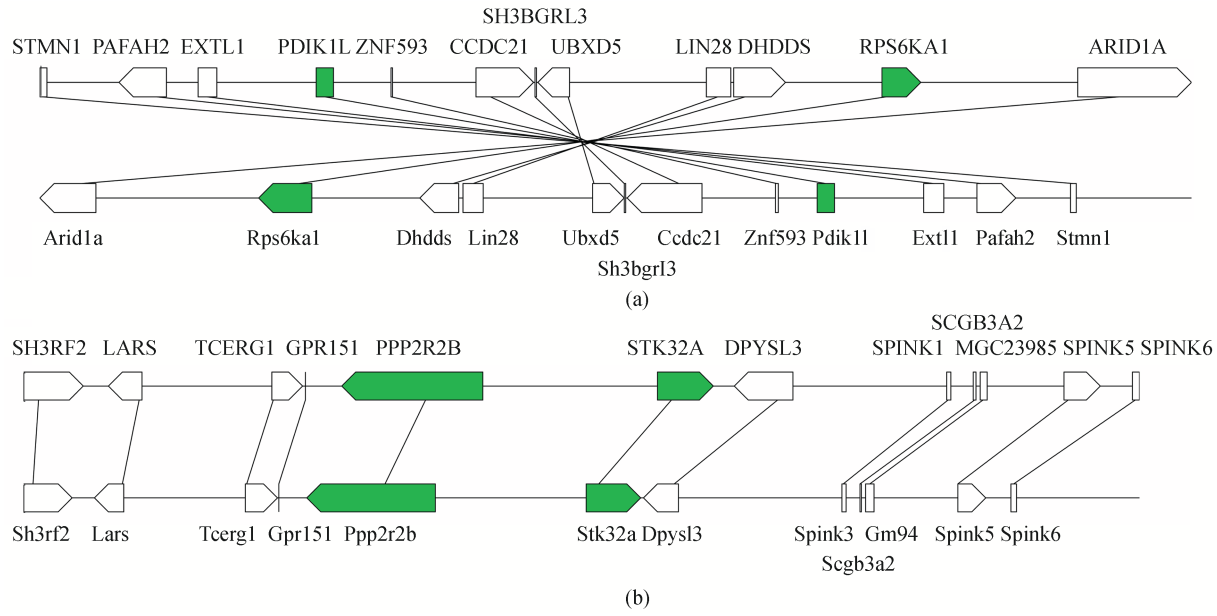


Fig.5. (a) Assigned orthologs on segment (26 099 194 bp ∼ 26 981 180 bp) of human chromosome 1 and the corresponding segment of mouse chromosome 4 (132 951 085 bp ∼ 133 745 914 bp), where MSOAR correctly identified mouse orthologs of genes PDIK1L and RPS6KA1 which were missed by INPARANOID. (b) Segment of the human chromosome 5 (145 296 334 bp ∼ 147 574 892 bp) and the corresponding segment in mouse chromosome 18 (42 179 639 bp ∼ 44 208 941 bp), where MSOAR identified orthologs of PPP2R2B and STK32A which were missed by INPARANOID.

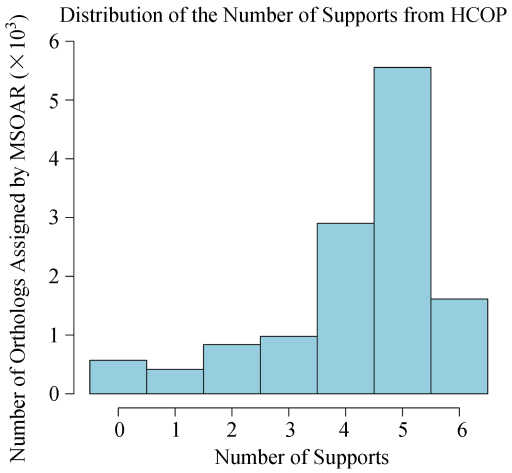


Fig.6. Consistency with the six methods on HCOP.

## 4 Some Key Algorithmic Problems

The experimental and comparison results in the last section demonstrate that:

1) The parsimony approach for ortholog assignment is very promising because it takes into account not only the sequence information of the genes, but also their positional information on the genomes. This advantage will become more apparent when more complete genomes are sequenced and annotated. The program MSOAR is very competitive to the existing methods for ortholog assignment, including those that involve human expert curation.

2) Genome-wide ortholog assignment still remains as a very challenging problem. For example, none of the methods listed on the HCOP website seem to yield dominant assignment results.

Hence, there is need and a large room to improve the performance of MSOAR.

In this section, we present a few key combinatorial optimization problems that are critical to the success of MSOAR and discuss possible ways of solving them. The problems include (i) *signed reversal distance with duplicates* (*SRDD*), (ii) *signed transposition distance with duplicates* (*STDD*), and (iii) *minimum common substring partition* (*MCSP*). Recall that MSOAR is based on the estimation of the RD (i.e., rearrangement/duplication) distance between the two input genomes, which is done in the 2nd stage of the program as shown in Fig.4, and the crux of this estimation is an efficient algorithm for calculating the rearrangement distance between two genomes with equal gene content, as shown in the first 3 steps in this stage. By using the technique introduced in [29-30], the chromosomes in two multichromosomal genomes can be appropriately concatenated so that the rearrangement events (i.e., reversal, translocation, fission, and fusion) in an optimal transformation between the genomes can all be thought of as reversals in the concatenated (unichromosomal) genomes. In other words, solving the rearrangement distance between two multichromosomal genomes is essentially equivalent to solving the reversal distance between two unichromosomal genomes, namely, the problem SRDD. The problem STDD considers the event transposition (including transreversal) in addition to reversal and is a straightforward extension of SRDD. It will be useful when we try to incorporate transposition into the parsimony approach. In the current version of MSOAR, the event transposition is simply ignored since it occurs less frequently than the other rearrangement events in evolution and is harder to deal with than reversal algorithmically. The problem MCSP is used in MSOAR as a preprocessing step for solving the SRDD problem and expected to help reduce the multiplicity in the instance (i.e., the maximum number of duplicates of any gene in each genome). Such a reduction is very important because it could help the next step of MSOAR (i.e., the 3rd step of stage 2) tremendously which uses the algorithmic framework given in [23, 29-31] for computing the reversal distance between permutations without duplicates.

These problems are all NP-hard, so we will mostly be interested in efficient approximation and heuristic algorithms for them, although certain special cases and variants implied by practical constraints will be considered too. The problems will be defined formally in separate subsections below. In addition, we will discuss how to extend MSOAR to deal with multiple species.

### 4.1 Signed Reversal Distance with Duplicates

From now on, a (unichromosomal) genome is represented as a string of signed symbols from a finite alphabet $\mathcal{A}$, where each sign ($+$ or $-$) represents a transcriptional orientation and a symbol denotes a gene. All the occurrences of a symbol in a genome constitute a *gene family*. A gene is called a *singleton* if it is the only member of its family; otherwise, it is a *duplicated* gene. Two genomes $G$ and $H$ are *related* if they have the same gene content, i.e., an equal number of gene families and an equal size of each family.

A reversal operation $\rho(i,j)$ transforms a genome $G = (g_1 \cdots g_{i-1} g_i g_{i+1} \cdots g_{j-1} g_j g_{j+1} \cdots g_n)$ into another genome $G \cdot \rho(i,j) = (g_1 \cdots g_{i-1} - g_j - g_{j-1} \cdots - g_{i+1} - g_i g_{j+1} \cdots g_n)$, where $-g_i$ means the gene $g_i$ with an opposite orientation. Given two related genomes $G$ and $H$, the *signed reversal distance with duplicates* (*SRDD*) problem is to find the smallest number of reversals $\rho_1, \rho_2, \cdots, \rho_r$ such that $G \cdot \rho_1 \cdot \rho_2 \cdots \rho_r = H$. The signed reversal distance between $G$ and $H$ is thus $d(G,H) = r$. We note in passing that the *unsigned*

version of SRDD has been studied in [32], where some complexity results and upper/lower bounds are obtained.

If all the genes in $G$ and $H$ are singletons, the signed reversal distance problem is usually referred to as the problem of *sorting signed permutations by reversals* or simply, *sorting by reversals* (*SBR*), and the distance can be calculated by the well-known Hannenhalli-Pevzner formula[23]:

$$d(G,H) = b(G,H) - c(G,H) + h(G,H) + f(G,H)$$

where $b(G,H)$ is the number of black edges (each representing a *breakpoint*) in the *breakpoint graph* for $G$ and $H$, $c(G,H)$ the number of cycles in a maximum cycle decomposition, $h(G,H)$ the number of *hurdles*, and $f(G,H)$ the number of *fortresses*, respectively. (The reader is referred to [23] for the definitions of breakpoint graph, black edges, hurdles, and fortresses.) The basic idea behind this seminal work is to view one of the input permutations as the identity permutation and represent the other permutation as a breakpoint graph whose edges are bi-colored. The reversal distance between the two permutations can then be calculated by decomposing the breakpoint graph into a maximum number of edge-disjoint cycles with alternating colors and counting the numbers of breakpoints, cycles, and hurdles, as well as checking the existence of a fortress. The best running time for transforming a permutation into the other with the minimum number of reversals is quadratic[33], although the signed reversal distance can be computed in linear time[34].

When the genomes $G$ and $H$ contain duplicated genes, however, the signed reversal distance problem, i.e., SRDD, cannot be directly solved by the Hannenhalli-Pevzner method anymore. In fact, SRDD has been shown NP-hard in [9] (also, independently in [35]) by a reduction from the NP-hardness of sorting *unsigned* permutations by reversals[36-37]. The NP-hardness holds even if the maximum size of a gene family is limited to two. Nevertheless, as mentioned above, it will be of tremendous interest to design an efficient and effective algorithm for SRDD. Below, we describe a lower bound on SRDD using a graphical structure similar to the breakpoint graph, which will be useful for constructing efficient approximation algorithms for SRDD.

Following the notations in [19, 32], we convert a (signed) genome $G = (g_1 g_2 \cdots g_n)$ to an unsigned one by replacing each gene $g_i$ with a string $g_i^h g_i^t$ if $g_i$ is positive or $g_i^t g_i^h$ if $g_i$ is negative, as it is done in the breakpoint graph[23,38]. A *partial graph*[19] associated with a genome $G = (g_1 g_2 \cdots g_n)$ is the graph $\mathcal{G}(V,E)$, where $V = \{g_i^s | 1 \leqslant i \leqslant n,\ s \in \{h,t\}\}$, and each (undirected) edge in $E$ links two nodes in $V$ that correspond to adjacent symbols in the genome $G$ except pairs of $g_i^h$ and $g_i^t$ from the same gene $g_i$. Let $\tilde{V}$ be the set of distinct symbols in $V$, where $g_i^h$ and $g_j^h$ are viewed as the same symbol if $g_i$ and $g_j$ are from the same family. Clearly, the partial graphs of a pair of related genomes have an identical vertex set $V$ and set $\tilde{V}$.

For each pair of elements $\{\tilde{v}_1, \tilde{v}_2\} \in \tilde{V}$, let $f_G(\tilde{v}_1, \tilde{v}_2)$ denote the number of edges in $E$ that link two nodes in the partial graph $\mathcal{G}(V,E)$ of $G$ with symbols $\tilde{v}_1$ and $\tilde{v}_2$, respectively. The number of *breakpoints* between two related genomes $G$ and $H$ is defined as:

$$b_r(G,H) = \sum_{\{\tilde{v}_1,\tilde{v}_2\}\in\tilde{V}} \delta(f_H(\tilde{v}_1,\tilde{v}_2) - f_G(\tilde{v}_1,\tilde{v}_2))$$

where $\delta(x) = x$ if $x > 0$ and 0 otherwise[32]. It is easy to see that $b_r(G,H) = b_r(H,G)$, although the above definition is not explicitly symmetric with respect to the two genomes. This new definition of breakpoints is a natural extension to the concept of breakpoints employed in the Hannenhalli-Pevzner theory for SBR on signed permutations with no duplicates. By observing that a reversal operation reduces the number of breakpoints by at most two, we have a lower bound on the reversal distance in terms of the number of breakpoints (which can easily be calculated from the partial graphs)[9]:

$$d(G,H) \geqslant \lceil b_r(G,H)/2 \rceil.$$

Although the lower bound resembles that in [39], which was used to obtain a 2-approximation algorithm for SBR (i.e., an algorithm that sorts the input permutation with at most twice the minimum number of reversals), at the moment we do not know of any matching upper bound on SRDD in terms of the number of breakpoints and probably some other relevant parameters that can be computed efficiently. Such an upper bound, if exists, could potentially lead to an efficient approximation algorithm for SRDD. However, we know that the number of breakpoints alone cannot provide an upper bound on the reversal distance, since we can prove that for any $\alpha > 0$, there exist genomes $G, H$ such that

$$d(G,H) > \alpha \cdot b_r(G,H).$$

One approach to deriving feasible upper bounds is to replace breakpoints by something more "global", e.g., the number of pairwise inversions of the elements on genome $G$ with respect to their positions on genome $H$. A greedy-style algorithm can potentially be designed to reduce the number of pairwise inversions by performing reversals. However, it is unclear how to lowerbound the reversal distance or upper bound the performance

of the greedy algorithm in terms of the number of pairwise inversions. It would be interesting to know if this approach could lead to a good approximation algorithm for SRDD.

Another method is to define a graphical structure similar to the breakpoint graph, called the *complete-breakpoint graph*[19], and then analyze its cyclic structure as was done in [23, 38] for SBR. Following the constructions in [23, 38], we could modify the definition of a cycle decomposition (by introducing some constraints) so that it corresponds to a feasible solution of SRDD[9-10]. It would be interesting to consider strategies to find a large feasible cycle decomposition under the new definition and see if they could lead to a promising approximation algorithm. (Currently, MSOAR adopts a simple greedy strategy.) Besides the well-known methods in [23, 33-34, 38], the new decomposition strategy in [40] will also be consulted.

The third approach to approximating SRDD is based on the problem of MCSP (to be defined formally in Subsection 4.3). Intuitively, given an instance of SRDD, an algorithm for MCSP could be applied to divide the genomes into corresponding segments. Since the segments are likely to be unique in practice, this preprocessing step (as used in MSOAR) could result in a pair of permutations without any duplicates (i.e., an instance of SBR) or an instance with few duplicates. Because SBR has very efficient algorithms, this could be a very effective solution for SRDD in practice, if the algorithm for MCSP performs well. In Subsection 4.3, we will show that in fact, an $\alpha$-approximation algorithm for MCSP④ implies a $2\alpha$-approximation algorithm for SRDD with essentially the same running time.

For any constant $k > 0$, let $k$-SRDD denote SRDD restricted to instances with multiplicity bounded by $k$ (i.e., each gene family has the size of at most $k$). It would be interesting to study $k$-SRDD taking advantage of the bounded multiplicity. Efficient algorithms for $k$-SRDD could still be very useful for ortholog assignment since there are very few large gene families in practice. Note that the results in [32, 35] concern SRDD with a bounded number of gene families (or alphabet size) instead and are only of theoretical interest.

### 4.2 Signed Transposition Distance with Duplicates

A transposition operation $\rho(i,j)$ transforms a genome $G = (g_1 \cdots g_{i-1} g_i \cdots g_j g_{j+1} \cdots g_k g_{k+1} \cdots g_n)$ into another genome $G \cdot \rho(i,j) = (g_1 \cdots g_{i-1} g_{j+1} \cdots g_k g_i \cdots g_j g_{k+1} \cdots g_n)$ or $G \cdot \rho(i,j) = (g_1 \cdots g_{i-1} g_{j+1} \cdots g_k - g_j \cdots - g_i g_{k+1} \cdots g_n)$. Note that, the second case is sometimes referred to as *transreversals*[41-42] and it includes reversals. Here, we will call it a transposition for simplicity. Given two related genomes $G$ and $H$, the *signed transposition distance with duplicates* (*STDD*) problem is to find the smallest number of transpositions $\rho_1, \rho_2, \cdots, \rho_t$ such that $G \cdot \rho_1 \cdot \rho_1 \cdots \rho_t = H$. The signed transposition distance between $G$ and $H$ is defined as $d_t(G,H) = t$. Similar to $k$-SRDD, the restricted version $k$-STDD can be defined for instances with bounded multiplicity. As mentioned before, an efficient solution to STDD or $k$-STDD will be crucial for our ortholog assignment approach if we want to incorporate transpositions into our evolutionary model.

When no duplicates are present, STDD becomes the problem of *sorting signed permutations by transpositions* or *sorting by transpositions* (*SBT*). Although SBT has been extensively studied in the literature, its complexity remains open. In fact, the complexity of its unsigned version is also open. Several approximation algorithms for SBT have been given (e.g., [41-42]), with the best approximation ratio being 1.5. These algorithms are all based on the breakpoint graph structure originally introduced in [43].

We can show that STDD is NP-hard by modifying the reduction from 3-partition given in [35] for proving the NP-hardness of the unsigned version of STDD. This reduction has to assume unbounded multiplicity and thus does not work for $k$-STDD. An interesting open question is the complexity of $k$-STDD (for any $k$) and if the breakpoint graph structures in [41-43] can be extended to produce efficient approximation algorithms STDD and $k$-STDD.

### 4.3 Minimum Common Substring Partition

Given a genome $G = g_1 g_2 \cdots g_n$, a *segment* $s_i$ is a substring (or its signed reversal) of $G$. A *partition* of $G$ is a list $\{s_1, s_2, \cdots, s_m\}$ of segments of $G$ such that the concatenation of the segments (or their signed reversals) in some order results in $G$. The list can be viewed as a *contracted representation* of $G$ if we consider each segment $s_i$ as a symbol. A list of segments is called a *common partition* of two (related) genomes $G$ and $H$ if it is a partition of $G$ and a partition of $H$ as well. Note that the contracted representations of two related genomes induced by a common partition are still related to each other. Furthermore, a *minimum common partition* is a partition with the minimum cardinality (denoted as $L(G,H)$) over all possible common partitions of $G$ and $H$. For example, for genomes $G = +c + a + b - a + d$ and $H = +c + a - d + a - b$, a minimum common partition is $\{+c + a, +b - a, +d\}$

④That is, an algorithm that returns a common substring partition with cardinality at most $\alpha$ times the optimum. The parameter $\alpha$ is also called the *approximation ratio* of the algorithm.

with $L(G,H) = 3$. The *minimum common substring partition (MCSP)* problem is defined as the problem of finding the minimum common partition between two given genomes.[⑤]

MCSP is used in MSOAR as a preprocessing step before solving SRDD. In fact, MCSP provides very good upper and lower bounds on SRDD. It is shown in [9] that for any two genomes $G$ and $H$,

$$\lceil (L(G,H)-1)/2 \rceil \leqslant d(G,H) \leqslant L(G,H) - 1.$$

(Actually, in order for these bounds to hold precisely, we need assume that the first genes of $G$ and $H$, as well as the two last genes, are identical and positive singletons, which can be easily satisfied in the general case by padding the strings $G$ and $H$ with dummy symbols.) This relationship suggests a method to approximate SRDD by MCSP. Unfortunately, MCSP is also NP-hard[44]. The NP-hardness holds even if the instance is restricted to have multiplicity at most $k$, i.e., an instance of $k$-MCSP, for any $k \geqslant 2$.

Approximation algorithms for MCSP have been recently investigated in [9, 44-46]. In particular, [9] presents an approximation algorithm based on a new graphical structure called *pair-match graphs*. Given two related genomes $G$ and $H$, a *single-match* is a pair of identical genes $g_i$ and $h_j$ from $G$ and $H$ that may have different signs. A *pair-match* is a pair of adjacent gene pairs $g_ig_{i+1}$ and $h_jh_{j+1}$ that are identical or the (signed) reversal of each other. Clearly, a pair-match consists of two single-matches. Observe that two pair-matches may not co-exist in a common partition. Such pair-matches are said to be *incompatible*. For two related genomes $G$ and $H$, we can construct a pair-match graph $\mathcal{P}(V,E)$, where $V$ consists of all possible pair-matches between $G$ and $H$, and $E$ includes edges connecting incompatible pair-matches. Then it is not hard to see that MCSP on genomes $G$ and $H$ is equivalent to the maximum independent set problem on $\mathcal{P}(V,E)$. Since the complement of an independent set of $\mathcal{P}(V,E)$ is a vertex cover of $\mathcal{P}(V,E)$, the algorithm in [9] approximates MCSP by using an efficient approximation algorithm for the vertex cover problem, and outputs a common partition of size at most $(r-1)(|V|-n) + r \cdot L(G,H)$, where $r$ is the approximation ratio for vertex cover, $|V|$ the number of vertices in the pair-match graph, and $n$ the size of genome $G$ (or $H$). In particular, for 2-MCSP, the algorithm achieves an approximation ratio of 1.5, since the pair-match graph $\mathcal{P}(V,E)$ is 6-claw-free, $|V| \leqslant n$, and there exists a 1.5-approximation algorithm for vertex cover on 6-claw-free graphs[47]. This algorithm is currently employed in MSOAR.

More recently, Kolman presented an efficient $O(k)$-approximation algorithm for $k$-MCSP by using the concept of hitting sets and efficient algorithms for suffix trees and the disjoint set union problem[46,48]. Note that, this implies an efficient $O(k)$-approximation algorithm for $k$-SRDD as well. However, the approximation ratio of $O(k)$ is clearly unsatisfactory since it does not even imply a constant ratio approximation for MCSP (or SRDD, respectively). Moreover, the algorithm seems to perform worse than the above algorithm based on pair-match graphs and vertex cover when $k$ is very small, and it may not be very useful in the real ortholog assignment practice since the parameter $k$ could be as large as 100 for Eukaryotic genomes such as human and mouse.

It would be interesting to know if there is an efficient approximation algorithm for MCSP with a constant approximation ratio (our conjecture is on the positive side). A plausible starting point would be trying to improve the analyses in [9, 48] and considering a hybrid combination of both algorithms (since both try to preserve common pairs of adjacent genes), but perhaps some new ideas (e.g., insights into the pair-match graph) and techniques will be necessary too.

### 4.4 Comparison of Multiple Genomes

It is also interesting to study the orthology relationship among genes from several species. The simultaneous comparison of multiple genomes may help boost our confidence about true ortholog pairs and reduce the number of false positives[13]. A natural extension of the parsimony approach would be, given a set of genomes (for each species) and a species tree depicting the evolutionary history of the species, to reconstruct a genome for each ancestor in the tree so that the total RD distance between each species and its parent is minimized. However, this is obviously an NP-hard problem, and moreover, the existing methods[49-50] for dealing with multiple genome rearrangement (without duplicated genes) do not seem to work well here, because of the presence of duplicates.

One idea is to run MSOAR on each pair of input genomes and then combine the results consistently into an ortholog assignment for all input genomes[51]. Another idea is to apply the "lifting" method introduced in [52] for computing multiple sequence alignment under a fixed phylogeny to the multiple genome ortholog assignment problem. The basic idea is that we will try to "lift" some input genome to each ancestral species and then build the whole orthology relationship by

[⑤]The problem of MCSP was also independently introduced recently in [21], under the name *sequence cover*.

computing pairwise ortholog assignment between each species and its ancestor using MSOAR. As shown in [52], the question of which genome should be lifted to each ancestral species can be solved by simple dynamic programming. Of course, the success of the approach will rely on the performance and efficiency of MSOAR, since it generally requires the computation of many pairwise ortholog assignments.

## References


[1] Fitch W M. Distinguishing homologous from analogous proteins. *Syst. Zool.*, 1970, 19(2): 99-113.

[2] Koonin E V. Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, 2005, 39: 309-338.

[3] Remm M, Storm C, Sonnhammer E. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, 2001, 314(5): 1041-1052.

[4] Sankoff D. Genome rearrangement with gene families. *Bioinformatics*, 1999, 15(11): 909-917.

[5] Tatusov R L, Galperin M Y, Natale D A, Koonin E V. The COG database: A tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.*, 2000, 28(1): 33-36.

[6] Tatusov R L, Koonin E V, Lipman D J. A genomic perspective on protein families. *Science*, 1997, 278: 631-637.

[7] Altschul S, Madden T, Schaffer A, Zhang J, Zhang Z, Miller W, Lipman D. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 1997, 25(17): 3389-3402.

[8] Chen X, Zheng J, Fu Z, Nan P, Zhong Y, Lonardi S, Jiang T. Computing the assignment of orthologous genes via genome rearrangement. In *Proc. the 3rd Asia Pacific Bioinformatics Conf.* (*APBC 2005*), Singapore, Jan. 17-21, 2005, pp.363-378.

[9] Chen X, Zheng J, Fu Z, Nan P, Zhong Y, Lonardi S, Jiang T. The assignment of orthologous genes via genome rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2005, 2(4): 302-315.

[10] Fu Z, Chen X, Vacic V, Nan P, Zhong Y, Jiang T. A parsimony approach to genome-wide ortholog assignment. In *Proc. the 10th Annual International Conference on Research in Computational Molecular Biology* (*RECOMB*), Venice, Italy, April 2-5, 2006, pp.578-594.

[11] Fu Z, Chen X, Vacic V, Nan P, Zhong Y, Jiang T. MSOAR: A high-throughput ortholog assignment system based on genome rearrangement. *Journal of Computational Biology*, 2007, 14(9): 1160-1175.

[12] Lee Y, Sultana R, Pertea G, Cho J, Karamycheva S, Tsai J, Parvizi B, Cheung F, Antonescu V, White J, Holt I, Liang F, Quackenbush J. Cross-referencing eukaryotic genomes: TIGR orthologous gene alignments (TOGA). *Genome Res.*, 2002, 12(3): 493-502.

[13] Li L, Stoeckert C, Roos D. OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Res.*, 2003, 13(9): 2178-2189.

[14] Yuan Y P, Eulenstein O, Vingron M, Bork P. Towards detection of orthologues in sequence databases. *Bioinformatics*, 1998, 14(3): 285-289.

[15] Storm C, Sonnhammer E. Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics*, 2002, 18(1): 92-99.

[16] Cannon S B, Young N D. OrthoParaMap: Distinguishing orthologs from paralogs by integrating comparative genome data and gene phylogenies. *BMC Bioinformatics*, 2003, 4(1): 35.

[17] Zheng X H, Lu F, Wang Z, Zhong F, Hoover J, Mural R. Using shared genomic synteny and shared protein functions to enhance the identification of orthologous gene pairs. *Bioinformatics*, 2005, 21(6): 703-710.

[18] Kuzniar A, van Ham R, Pongor S, Leunissen J. The quest for orthologs: Finding the corresponding gene across genomes. *Trends in Genetics*, 2008, 24(11): 539-550.

[19] El-Mabrouk N. Reconstructing an ancestral genome using minimum segments duplications and reversals. *Journal of Computer and System Sciences*, 2002, 65(3): 442-464.

[20] Marron M, Swenson K, Moret B. Genomic distances under deletions and insertions. *Theoretical Computer Science*, 2004, 325(3): 347-360.

[21] Swenson K, Marron M, Earnest-DeYoung J. Moret B. Approximating the true evolutionary distance between two genomes. In *Proc. the 7th SIA Workshop on Algorithm Engineering & Experiments*, Vancouver, Canada, Jan. 22, 2005, pp.121-125.

[22] Swenson K, Pattengale N, Moret B. A framework for orthology assignment from gene rearrangement data. In *Proc. the 3rd RECOMB Workshop on Comparative Genomics* (*RECOMB-CG 2005*), Dublin, Ireland, Sept. 18-20, 2005, *LNCS* 3678, Springer, pp.153-166.

[23] Hannenhalli S, Pevzner P. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *J. ACM*, 1999, 46(1): 1-27; extended abstract in *Proc. ACM STOC*, Las Vegas, USA, May 23-June 1, 1995, pp.178-189.

[24] Shi G, Zhang L, Jiang T. MSOAR 2.0: Incorporating tandem duplications into ortholog assignment based on genome rearrangement. In *Proc. the 8th LSS Computational Systems Bioinformatics Conference*, Stanford, USA, August 10-12, 2009, pp.12-24.

[25] Bairoch A, Apweiler R, Wu C H, Barker W C, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin M J, Natale D A, O'Donovan C, Redaschi N, Yeh L S. The Universal Protein Resource (UniProt). *Nucleic Acids Res.*, 2005, 33(Database Issue): D154-D159.

[26] http://www.gene.ucl.ac.uk/cgi-bin/nomenclature/hcop.pl.

[27] ftp://ftp.pantherdb.org/sequence_classifications/.

[28] http://www.jax.org.

[29] M Ozery-Flato, Ron Shamir. Two notes on genome rearragnements. *Journal of Bioinformatics and Computational Biology*, 2003, 1(1): 71-94.

[30] Tesler G. Efficient algorithms for multichromosomal genome rearrangements. *Journal of Computer and System Sciences*, 2002, 65(3): 587-609.

[31] Hannenhalli S, Pevzner P A. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proc. IEEE 36th Ann. Symp. Foundations of Comp. Sci.* Milwaukee, USA, Oct. 23-25, 1995, pp.581-592.

[32] Christie D, Irving R. Sorting strings by reversals and by transpositions. *SIAM J. Discrete Math.*, 2001, 14(2): 193-206.

[33] Kaplan H, Shamir R, Tarjan R. Faster and simpler algorithm for sorting signed permutations by reversals. In *Proc. the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, USA, Jan. 5-7, 1997, pp.344-351.

[34] Bader D, Moret B, Yan M. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 2001, 8(5): 483-491.

[35] Radcliffe A, Scott A, Wilmer E. Reversals and transpositions over finite alphabets. *SIAM J. Discrete Math.*, 2005, 19(1): 224-244.

[36] Caprara A. Sorting by reversals is difficult. In *Proc. the First Annual International Conference on Computational Molecular Biology*, Santa Fe, USA, Jan. 20-23, 1997, pp.75-83.

[37] Caprara A. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM J. Discrete Math.*, 1999, 12(1): 91-110.

[38] Bafna V, Pevzner P. Genome rearrangements and sorting by reversals. *SIAM J. Comput.*, 1996, 25(2): 272-289; extended abstract appeared in *Proc. IEEE FOCS 1993*, Palo Alto, USA, Nov. 3-5, 1993, pp.148-157.

[39] Kececioglu J, Sankoff D. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 1995, 13(1/2): 180-210.

[40] Lin G, Jiang T. A further improved approximation algorithm for breakpoint graph decomposition. *Journal of Combinatorial Optimization*, 2004, 8(2): 183-194.

[41] Gu Q, Peng S, Sudborough H. A 2-approximation algorithm for genome rearrangements by reversals and transpositions. *Theoret. Comput. Sci.*, 1999, 210(2): 327-339.

[42] Hartman T, Sharon R. A 1.5-approximation algorithm for sorting by transpositions and transreversals. *Journal of Computer and System Sciences*, 2005, 70(3): 300-320.

[43] Bafna V, Pevzner P. Sorting by transpositions. *SIAM J. Discrete Math.*, 1998, 11(2): 224-240.

[44] Goldstein A, Kolman P, Zheng J. Minimum common string partition problem: Hardness and approximations. In *Proc. the 15th International Symposium on Algorithms and Computation*, Hong Kong, China, Dec. 20-22, 2004, pp.484-495.

[45] Chrobak M, Kolman P, Sgall J. The greedy algorithm for the minimum common string partition problem. In *Proc. the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, Cambridge, USA, Aug. 22-24, 2004, *LNCS* 3122, Springer, pp.84-95.

[46] Kolman P. Approximating aeversal distance for strings with bounded number of duplicates. In *Proc. the 30th International Symposium on Mathematical Foundations of Computer Science*, Gdansk, Poland, Aug. 29-Sept. 2, 2005, pp.580-590.

[47] Halldorsson M M. Approximating discrete collections via local improvements. In *Proc. the Sixth Annual ACM-SIAM Symp. Discrete Algorithms*, San Francisco, USA, Jan. 22-24, 1995, pp.160-169.

[48] Kolman P, Walen T. Reversal distance for strings with duplicates: Linear time approximation using hitting set. In *Proc. the 4th Workshop on Approximation and Online Algorithms*, Zurich, Switzerland, Sept. 14-15, 2006, pp.279-289.

[49] Bourque G, Pevzner P. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Research*, 2002, 12(1): 26-36.

[50] Sankoff D, Blanchette M. Multiple genome rearrangement and breakpoint phylogeny. *Journal of Computational Biology*, 1998, 5(3): 555-570.

[51] Fu Z, Jiang T. Clustering of main orthologs for multiple genomes. *Journal of Bioinformatics and Computational Biology*, 2008, 6(3): 573-584.

[52] Wang L, Jiang T, Lawler E. Approximation algorithms for tree alignment with a given phylogeny. *Algorithmica*, 1996, 16(3): 302-315.

**Tao Jiang** received the B.S. degree in computer science and technology from the University of Science and Technology of China, Hefei, in July 1984 and Ph.D. degree in computer science from University of Minnesota in Nov. 1988. He was a faculty member at McMaster University, Hamilton, Ontario, Canada during Jan. 1989∼July 2001 and is now professor of computer science and engineering at University of California — Riverside (UCR). He is also a member of the UCR Institute for Integrative Genome Biology, a member of the Center for Plant Cell Biology, a principal scientist at Shanghai Center for Bioinformation Technology, and Changjiang Visiting Professor at Tsinghua University. Tao Jiang's recent research interest includes combinatorial algorithms, computational molecular biology, bioinformatics, and computational aspects of information retrieval. He is a fellow of ACM and of AAAS. More information about his work can be found at http://www1.cs.ucr.edu/~jiang.