UNIVERSITY OF CALIFORNIA
RIVERSIDE

Combinatorial Approaches to Accurate Identification of Orthologous Genes

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Guanqun Shi

August 2011

Dissertation Committee:

Dr. Tao Jiang, Chairperson
Dr. Stefano Lonardi
Dr. Neal E. Young

The Dissertation of Guanqun Shi is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would not have been able to finish my dissertation without the guidance of my committee members, the help of my friends and the support of my family members.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Tao Jiang, for everything I learned from him in the past few years. As my advisor, he gave me numerous invaluable advise on my study, research work, as well as life in the United States. I also appreciate his excellent guidance, persistent encouragement and consistent support in my research work.

I would like to thank Dr. Stefano Lonardi and Dr. Neal E. Young for providing advice in my research work and helping me to develop my knowledge in algorithm design and analysis, approximation algorithms, computational theory, and bioinformatics. I am also grateful to Dr. Eamonn Keogh and Dr. Jun Li (Statistics Department, UCR) for being on my oral qualifying exam committee.

My thanks also goes to my collaborators, Dr. Liqing Zhang (Department of Computer Science, Virginia Tech) and Meng-Chih Peng, for their valuable comments and discussions in our collaboration. I would like to thank my group members: Zheng Fu (now with Google), Lan Liu (now with Google), Bob Wang, Wei Li, Minzhu Xie and Olga Tanaseichuk, my lab mates: Claire Huang, Li Yan, Elena Strzheletska, Monik Khare and Alman Yousefi, my previous lab mates: Yonghui Wu (now with Google), Christos Koufogiannakis, Serdar Bozdag, Vladimir Vacic, Elena Harris, Zhaocheng Fan, Yang Yang, Dandan Song, Jiayuan Zhao, Jianxin Feng, for their help both in work and life at UCR.

Last but not least, I would like to thank my mom and dad for their unconditional

love and consistent support of my study with their best wishes. Special thanks goes to my girlfriend Xiaoxiao, who is always there cheering me up and accompanying me through the good times and bad.

ABSTRACT OF THE DISSERTATION

Combinatorial Approaches to Accurate Identification of Orthologous Genes

by

Guanqun Shi

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, August 2011
Dr. Tao Jiang, Chairperson

The accurate identification of orthologous genes across different species is a critical and challenging problem in comparative genomics and has a wide spectrum of biological applications including gene function inference, evolutionary studies and systems biology. During the past several years, many methods have been proposed for ortholog assignment based on sequence similarity, phylogenetic approaches, synteny information, and genome rearrangement. Although these methods share many commonly assigned orthologs, each method tends to produce an ortholog assignment significantly different from the others.

In this dissertation, we study the problem of assigning orthologous genes among closely related genomes on a genome scale. We first give a brief review of the existing methods for ortholog assignment in the literature, followed by a comprehensive comparison of each method. We then propose a new combinatorial approach for assigning ortholog pairs between a pair of closely related genomes by addressing the limitations of the existing methods. Our approach is based on the parsimony principle to transform one genome to another by minimizing the number of genome rearrangement events, including reversal,

transposition, fusion, fission and gene duplications. By explicitly incorporating tandem gene duplication model and combining phylogenetic approaches, we develop an improved system MSOAR 2.0. Our experimental results on both simulated data and real data show that MSOAR 2.0 achieves the highest overall prediction accuracy among different programs in comparison.

Based on pairwise genome comparison results, we extend our ortholog assignment method to multiple genome comparison and develop a new system MultiMSOAR 2.0 to identify ortholog groups among multiple genomes. In MultiMSOAR 2.0, pairwise orthology information produced by MSOAR 2.0 is used to construct multipartite graphs for each gene family. In order to partition each gene family into a set of disjoint sets of orthologous genes, a multidimensional matching problem is formulated and a heuristic maximum weight matching algorithm is proposed. The partition results are then used to label the species tree. Considering some biological constraints, we formulate the tree labeling problem in the combinatorial optimization framework and develop two dynamic programming algorithms to solve the problem. Our experimental results show that MultiMSOAR 2.0 achieves much higher prediction accuracy than the existing ortholog assignment systems for multiple genomes. Moreover, MultiMSOAR 2.0 also provides information about gene births, duplications and losses in evolution, which may be of independent biological interest.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Basic Concepts

The ever-increasing number of completely sequenced genomes brings great opportunities as well as challenges to the study of comparative genomics. It makes the study of the evolutionary history of closely related species at the genome level possible. It also enhances our ability to perform gene functional analysis across different species. For these purposes as well as many other applications, the identification of orthologous genes across different species often serves as a starting point.

Homologous genes (*i.e., homologs*) are genes that evolved from a common ancestral gene. There are two types of homologous genes, namely, orthologous genes and paralogous genes. Orthologous genes (*i.e., orthologs*) are homologous genes in different species that are separated by speciation events, while paralogous genes (*i.e., paralogs*) refer to genes in the same genome generated by duplication events [1].

According to the definition, orthologs originate by vertical descent from a single gene of the last common ancestor, so they are more likely to preserve the original gene function. Paralogs, on the other hand, may diverge in function since the duplicated copies are free to mutate due to lack of the original selective pressure. For instance, the four known classes of hemoglobins (hemoglobin A, hemoglobin A2, hemoglobin B, and hemoglobin F) are paralogs of each other. While each of these proteins serves the same basic function of oxygen transport, they have already diverged slightly in function: fetal hemoglobin (hemoglobin F) has a higher affinity for oxygen than adult hemoglobin [2]. As a result, orthologs, rather than paralogs, are often used as universal and unique landmarks within each genome as well as links across different genomes [3].

To better understand the evolutionary process, paralogs are further divided into two subtypes: *outparalogs* and *inparalogs* [4]. With respect to a given speciation event, outparalogs are genes duplicated before the speciation while inparalogs are genes duplicated after the speciation. These concepts as well as the relationship among orthologs, outparalogs and inparalogs are illustrated in Figure 1.1, which depicts the evolution of globin genes in human, mouse and rat.

Clearly, it is easy to identify orthologs across different species if the duplication history of the genes on the concerned genomes is given (relative to their speciation events). Unfortunately, this evolutionary process is unknown. What we know is all the genes in the contemporary genomes. In order to find the most probable ortholog assignment among different genomes, we need to reconstruct the true evolutionary history.

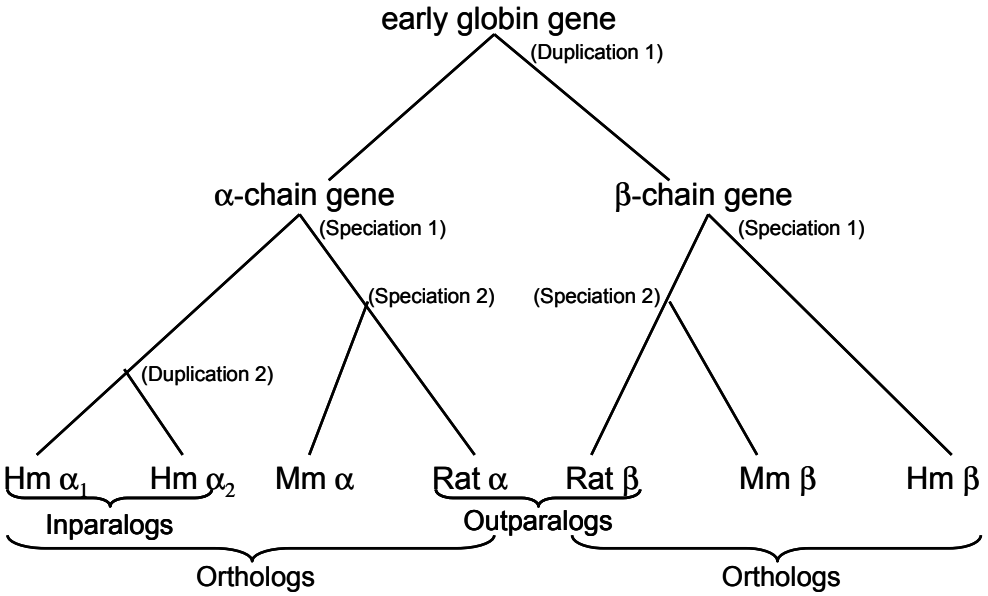


Figure 1.1: An illustration of orthology and paralogy relationships. The α globin genes on different species are orthologous to each other, and they form an ortholog group. So do the β globin genes. The rat α and β globin genes are outparalogs with respect to the speciation of mouse and rat (*i.e.*, speciation 2) as well as with respect to speciation 1. The human α_1 and α_2 globin genes are inparalogs with respect to the speciation of human and rat (*i.e.*, speciation 1).

1.2 Background

There are lots of ortholog databases publicly available nowadays, such as COG/KOG [5], Roundup [6], TreeFam [7], EnsemblCompara [8], eggNOG [9,10], OrthoDB [11,12], PhylomeDB [13], OMA [14,15]. However, most of these databases are restricted to include model organisms only and each database has a limitation of the number of species included. As a result, if a species of interest is not included in any of the current ortholog databases, we cannot hope to find its orthologs to other species based on database search. Besides, ortholog databases usually cannot provide orthology information at various levels of resolution. How to precisely extract the orthology information for only a small subset of genomes

of interest from the databases is not trivial.

Given the limitations of the current ortholog databases, some stand-alone ortholog assignment programs and tools are developed, such as InParanoid/MultiParanoid [4,16,17], CCCPart [18, 19], OrthoFocus [20], LOFT [21]. Applying these tools, users can find orthologs for a set of self defined genomes. However, most of these programs only play emphasis on one part of the available information, such as sequence similarity, phylogenetic information or genomic context. The prediction accuracies of these programs are far from satisfactory. Hardly can a current program take all the available information into consideration, and combine them in an integrated framework to assign orthologs.

1.3 Goals of the Dissertation

The goals of this dissertation are to propose new combinatorial approaches to assign orthologs by considering all available information, including sequence similarity, phylogenetic information, synteny information as well as genome rearrangement. By applying phylogenetic approach to remove duplicated inparalogs in tandem array on each genome, we first develop an improved system MSOAR 2.0 to assign ortholog pairs between two closely related genomes. Then we extend the system to multiple genome comparison and develop a new system MultiMSOAR 2.0 to identify ortholog groups among multiple genomes.

Both MSOAR 2.0 and MultiMSOAR 2.0 are high-throughput ortholog assignment systems that can be applied to identify orthologs on a genome scale. Experiments on both simulated data and real data have shown that both programs achieve much better prediction accuracy than the other programs in comparison.

Chapter 2

Literature Review

2.1 Sequence Similarity Based Methods

Most of the traditional ortholog identification methods are based on sequence similarity search, such as COG/KOG [5], Roundup [6], InParanoid/MultiParanoid [4, 16], OrthoMCL [22], and HomoloGene [23]. Generally speaking, these methods first calculate some pairwise similarity scores and then use some clustering algorithms to identify ortholog pairs or groups.

Take the InParanoid program for example. It assigns a gene pair with the bidirectional best hit (*i.e., BBH*) as a main ortholog pair and uses it as the "seed" to cluster similar genes from both genomes into an ortholog group. As its extension to multiple genomes, the MultiParanoid program basically clusters the pairwise orthology results of InParanoid to generate ortholog groups for multiple genomes. However, the BBH requirement for a main ortholog pair is often too stringent, especially when comparing multiple genomes. As

a result, the InParanoid and MultiParanoid program may miss a lot of true ortholog pairs and ortholog groups when some of the ortholog pairs are not BBHs.

OrthoMCL is an ortholog assignment program similar to InParanoid, but uses a different clustering algorithm (the Markov Clustering algorithm, or *MCL*) to find ortholog groups for multiple genomes. The algorithm simulates random walks on a graph using Markov matrices to determine the transition probabilities among nodes in the graph. Although the algorithm is more complicated than the clustering algorithm used in InParanoid, however, it cannot resolve the many-to-many orthology relationship among multiple genomes effectively. As a result, the ortholog groups found by OrthoMCL may include lots of "recent" inparalogs from each genome [22].

Instead of using BBH as the criteria to identify orthologs, some recent databases, such as the Roundup ortholog database, choose to use the reciprocal smallest distance algorithm (*i.e., RSD*) to assign orthologs. Since the RSD algorithm uses global rather than local sequence alignments and evolutionary estimates of distance between sequences rather than blast probability scores, it has been shown to improve upon the BBH approach [6].

2.2 Phylogenetic Tree Based Methods

Another popular method to identify orthologs is based on phylogenetic trees, such as TreeFam [7], EnsemblCompara GeneTrees [8], PhylomeDB [13], LOFT [21], and PhyOP [24]. Since a phylogeny can be used conveniently to represent the evolution of a gene family, ortholog can be assigned in a straightforward way.

Take the EnsemblCompara GeneTrees for example. It is a computational pipeline

to generate maximum likelihood phylogenetic gene trees and reconcile with the given species tree. It then predicts orthologs and paralogs by distinguishing duplication or speciation events on the reconciled tree. Since Ensembl gene trees are calculated using a new method, TreeBeST, which integrates multiple tree topologies, this method shows the best performance among many phylogenetic approaches in comparison [8]. However, this method, as well as many other phylogenetic methods, do not allow the identification of orthologs at different levels of resolution.

LOFT (Levels of Orthology From Trees) is a program developed to describe the multi-level nature of orthology relations. By introducing the concept of "levels of orthology", LOFT aims at assigning hierarchical orthology numbers to genes based on a phylogenetic tree, which can be used to make high-resolution ortholog assignment.

However, tree-based methods generally present orthology as a many-to-many relationship. Most of them can never tell the "parent-daughter" relationships among duplicated genes [25]. As a result, most tree-based methods cannot differentiate orthologs that are direct descendants of an ancestral gene and those inparalogs that are products of recent duplications. Consequently, each ortholog group found by these methods tends to include lots of lineage-specific duplicated inparalogs.

2.3 Synteny Based Methods

Neither sequence based nor phylogenetic based methods take gene order information into consideration. However, studies have shown that orthologous genes are most likely to be conserved in the same genomic context across different species [26, 27]. Based

on conserved synteny, lots of methods have been proposed in recent years.

By using local synteny information alone, Jun *et al.* have shown that their methods can achieve pretty high prediction accuracy in mammalian genomes, which is a robust substitute to coding sequence for identifying orthologs. Besides, they claim that local synteny can be used to identify non-orthologous gene displacement by retroduplicated paralogs [28].

CCCPart is a synteny-based approach to find orthologs based on the assumption that isofunctional genes are well preserved both in common genomic context as well as in sequence similarity between two or more species [18, 19]. By treating gene neighborhood as an edge connecting two vertices in a multiple graph, the program tries to find the common connected components in the graph.

EGM (Encapsulated Gene-by-Gene Matching) is another recently developed synteny-based program to identify orthologs between two genomes. It takes into account gene context and family information, and tries to find an optimal global gene matching between two genomes.

Synteny based methods are usually simple and accurate for very closely related genomes where the synteny information is well reserved. For comparison of genomes where lots of genome rearrangement events are involved, synteny information alone may not be sufficient to achieve good performance.

2.4 Genome Rearrangement Based Methods

Although most gene order may be conserved within syntenic blocks, it can be shuffled between different syntenic blocks. It is known that genome rearrangement is very

common between closely related genomes [29–32]. In fact, there might be many microrearrangments even within the same synteny block [31].

Based on genome rearrangement, a high-throughput one-to-one ortholog assignment system called MSOAR [33,34] has recently been developed. It is based on the assumption that orthologs should correspond to each other on the evolutionary path that minimizes the number of rearrangements and post-speciation duplications. The system attempts to reconstruct the evolutionary history of the genes in the input genomes in terms of genome rearrangement and gene duplication events, and tries to minimize the *RD* (rearrangement and duplication) distance under the parsimony principle.

2.5 Comparison of Different Ortholog Assignment Methods

Generally speaking, sequence similarity based methods are simple and efficient, and they are usually capable of achieving high specificity due to their stringent requirements. However, on the other hand, their sensitivity are usually low since they may miss quite a lot of true orthologs which may not be bidirectional best hits.

Phylogenetic approaches generally reconstruct the evolutionary history for a gene family, then assign orthologs in a straightforward way. However, the tree reconstruction and reconciliation process usually involves sophisticated algorithms and is time consuming, especially for large gene families. Moreover, the specificity of these methods are usually low due to inclusion of lots of lineage specific duplicated paralogs.

Synteny based methods usually performs well in prokaryotic genomes, where gene order is well conserved. For comparison in eukaryotic genomes, such as vertebrates, where

gene order is much more variable due to genome rearrangement, gene order information alone may not be able to provide high prediction accuracy [28].

Genome rearrangement based method (here we refer to MSOAR), combines sequence similarity based method as well as gene order information, and has shown to achieve better prediction accuracy than many sequence based methods [34]. However, it fails to incorporate phylogenetic information which may help to purify the prediction result. Besides, the random gene duplication model assumed in MSOAR is too simple to reflect the actual scenarios of gene duplications in reality.

There are some other methods for ortholog assignment proposed in recent years. Recent comprehensive reviews on ortholog assignment programs in the public domain can be found in [14, 35].

Chapter 3

Accurate Identification of Ortholog Pairs between a Pair of Genomes

3.1 Definition of One-to-One Ortholog Pairs

According to Fitch's definition, orthology between two species is in general a many-to-many relationship [1]. In other words, for a pair of genomes, an ortholog set consists of a pair of sets of inparalogs, one from each genome. The inparalogs in one set are co-orthologous to all the inparalogs in the other. For each set of inparalogs on a genome, there usually exists a gene that is the direct descendant of the ancestral gene of such a set, which is referred to as the "true examplar" by Sankoff [36], while the other inparalogs in the set are duplicated from the true examplar gene. Therefore, for each ortholog set, we may select a representative from each set of inparalogs (*e.g.*, the exemplar gene) and define a one-to-one ortholog pair consisting of the two representatives. Such an ortholog pair may contain the

two genes, one from each set, that correspond the best in terms of their positions on the genomes [34] or sequence similarity [4]. This allows us to think of orthology as a one-to-one relationship.

In fact, the one-to-one orthology relationship is critically used in many comparative genomics studies, such as the reconstruction of accurate gene trees [37], alignment of protein-protein interaction (PPI) networks across multiple species [38], identification of functional orthologs [39], evolutionary, comparative and systematic studies in plants [40], and mapping of biological pathways [41]. (One-to-one orthologs are called "true orthologs" in [39] and "single copy orthologous genes" in [40].) Note that once a one-to-one ortholog pair or ortholog group is specified for a set of orthologous genes, all other pairs or groups of genes from the set will be regarded as false positives (with respect to the one-to-one orthology relationship). In this chapter, we are interested in assigning orthologs as a one-to-one relationship. To avoid ambiguity, we will add the prefix "one-to-one" in front of such orthologs.

3.2 Introduction to MSOAR

MSOAR is a high-throughput one-to-one ortholog assignment system based on genome rearrangement. It tries to transform one genome to the other with the minimum number of genome rearrangement and gene duplication events.

3.2.1 Genome Rearrangement

The analysis and study of genome rearrangements in molecular evolution involves solving a combinational puzzle of finding a series of genome rearrangement events to transform one genome into another [29]. There are five common genome rearrangement events considered in the literature, which are illustrated in Figure 3.1.

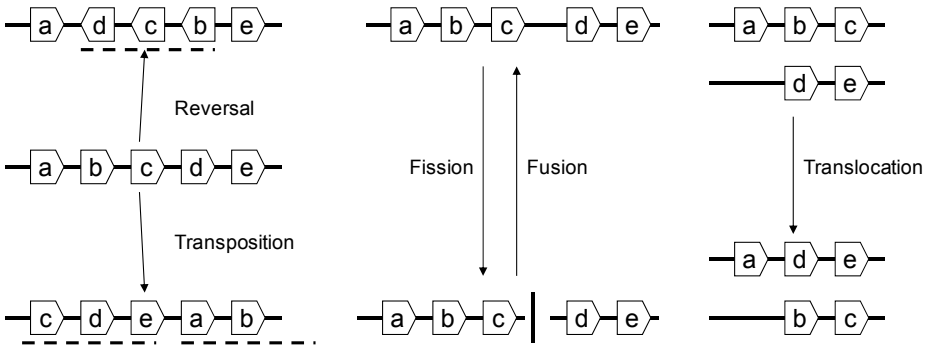
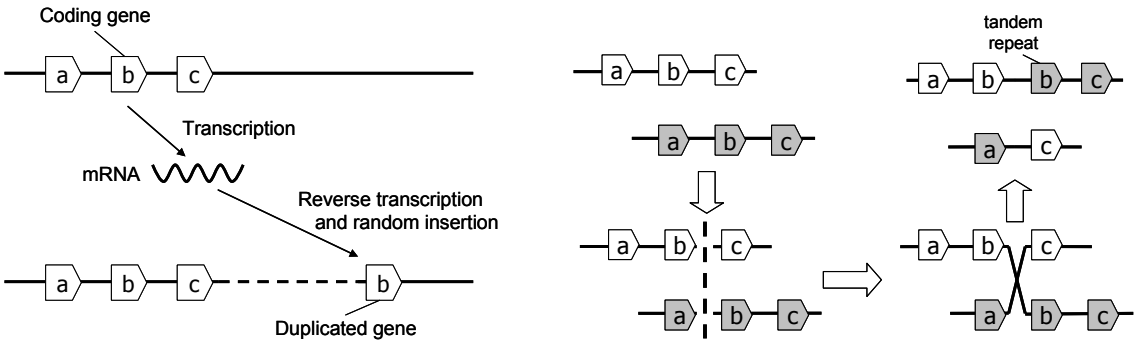


Figure 3.1: Five common rearrangement events.

3.2.2 Gene Duplication Models

The importance of gene duplication in molecular evolution is well established [42, 43]. However, the biological mechanism behind gene duplication has been unknown for quite many years. Recently, biologists proposed three different mechanisms for gene duplication based on the size of the duplication and whether they involve an RNA intermediate [44, 45]: retrotransposition, tandem duplication, and genome duplication.

Retrotransposition describes the integration of a reverse transcribed mRNA into the genome in a random manner (see Figure 3.2(a)), and is the cause of random duplications.



(a) Retrotransposition.

(b) Unequal crossover.

Figure 3.2: Gene duplication mechanisms.

Tandem duplication is one of the possible outcomes of "unequal crossover", which results from the homologous recombination between paralogous sequences (see Figure 3.2(b)). As a result, genes are duplicated next to their original copies in tandem arrays on the genome, which are known as *TAGs* (*i.e.*, *tandemly arrayed genes*) [46]. Genome duplication is probably due to the lack of disjunction between daughter chromosomes after DNA replication, and occurs more in plants than in animals. Recent studies show that there is another type of large-scale duplications, segmental duplication, which involves 1kb∼400kb nucleotides, though the molecular mechanism of segmental duplication is still unclear [44].

3.2.3 Drawbacks of MSOAR

MSOAR considers four genome rearrangement events including reversal (*i.e.*, inversion), translocation, fusion, and fission (the transposition can be mimicked by three reversals). For gene duplications, MSOAR assumes that a gene duplication event inserts a duplicated gene into the concerned genome at a random location (*i.e.*, the random dupli-

cation model). However, biologists believe that genes are more likely to be duplicated in tandem arrays in reality [46, 47].

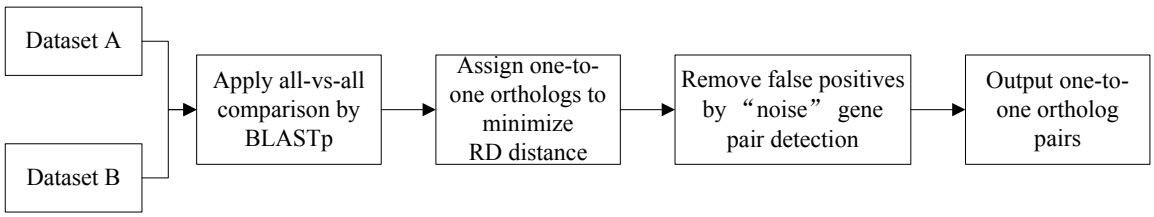
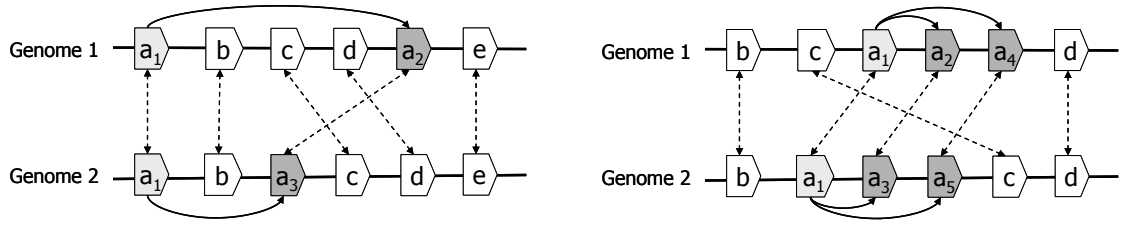


Figure 3.3: An outline of MSOAR.

For the convenience of the reader, an outline of the major algorithmic steps in MSOAR is sketched in Figure 3.3. In particular, MSOAR attempts to remove false one-to-one ortholog pairs that involve genes randomly duplicated after the speciation in the "noise" gene pair detection step. Such a (false) ortholog pair usually incurs a great cost in the rearrangement distance between the genomes, and thus we would be able to reduce the RD distance by "uncoupling" (*i.e.*, removing) the pair. However, in reality, randomly duplicated genes only account for a part of all duplicated genes. Recent studies have shown that at least 30% of duplicated genes are found next to their original copies (*i.e.*, in tandem positions) [46, 47].

Although MSOAR is able to identify most randomly duplicated inparalogs in the "noise" gene pair detection step, it is incapable of catching inparalogs that are produced by tandem duplications, which prevents MSOAR from identifying false one-to-one ortholog pairs that involve two duplicated inparalogs in TAGs from both genomes. See the examples in Figures 3.4.



(a) Genes a_2 and a_3 are randomly duplicated from gene a_1.

(b) Genes a_2, a_3, a_4 and a_5 are tandemly duplicated from gene a_1.

Figure 3.4: An example showing MSOAR's incapability of catching tandemly duplicated inparalogs.

In Figures 3.4, we assume that the genes with the same letter from the two genomes represent true one-to-one orthologs, and all duplications happened after the speciation in both genomes. For example, in Figure 3.4(b), (a_1, a_1) is a true one-to-one ortholog pair while (a_2, a_3) and (a_4, a_5) are not. The genes a_2 and a_3 in Figure 3.4(a) and genes a_2, a_3, a_4 and a_5 in Figure 3.4(b) are all duplicated from gene a_1 after the speciation, and thus are inparalogs of a_1. In both cases, MSOAR first tries to assign one-to-one orthology between all pairs of genes and calculates the RD distance between the two genomes. However, in the "noise" gene pair detection step, MSOAR is able to identify the false one-to-one ortholog pair (a_2, a_3) in Figure 3.4(a) since the RD distance between the two genomes will decrease by 1 (*i.e.*, 3 fewer reversals and 2 more duplications) if this pair is removed. However, if the duplicated genes are in TAGs, as shown in Figure 3.4(b), removing any of the pairs (a_2, a_3) and (a_4, a_5) will not affect the number of reversals but will increase the number of duplications by 2, thus increasing the RD distance between the two genomes. Since MSOAR tries to find an assignment to minimize the RD distance between the two genomes, it will correctly identify the false one-to-one ortholog pair (a_2, a_3) in Figure 3.4(a) while

incorrectly keep both false one-to-one ortholog pairs (a_2, a_3) and (a_4, a_5) in Figure 3.4(b) in the assignment.

3.3 Improved System MSOAR 2.0

3.3.1 Motivation

By addressing the drawbacks of MSOAR, we explicitly incorporate the tandem duplication model into MSOAR, and develop an improved system, simply called MSOAR 2.0, to assign one-to-one ortholog pairs between two genomes. The idea is to consider tandemly duplicated genes first and try to identify the inparalogy relationship among them using a simple phylogenetic tree reconciliation method. For each set of inparalogs (on the same genome), all but one gene will be deleted from the concerned genome before MSOAR is invoked. Our experimental results demonstrate that this pre-processing step could indeed remove many false positives correctly and thus greatly improve the specificity of MSOAR.

3.3.2 An Outline of MSOAR 2.0

The system MSOAR 2.0 has been implemented as a C++ application on a standard Linux system. Its main steps, as outlined in Figure 3.5, include: (i) the construction of gene families using a clustering approach, (ii) the identification of inparalogs in TAGs using a simple phylogenetic analysis, (iii) the invocation of MSOAR after removing inparalogs in TAGs, and (iv) the identification of additional one-to-one ortholog pairs in a post-processing step. The detailed description of each of the main steps is given below. The software is available to the public for free and can be downloaded from http://msoar.cs.ucr.edu/MSOAR2.0/.

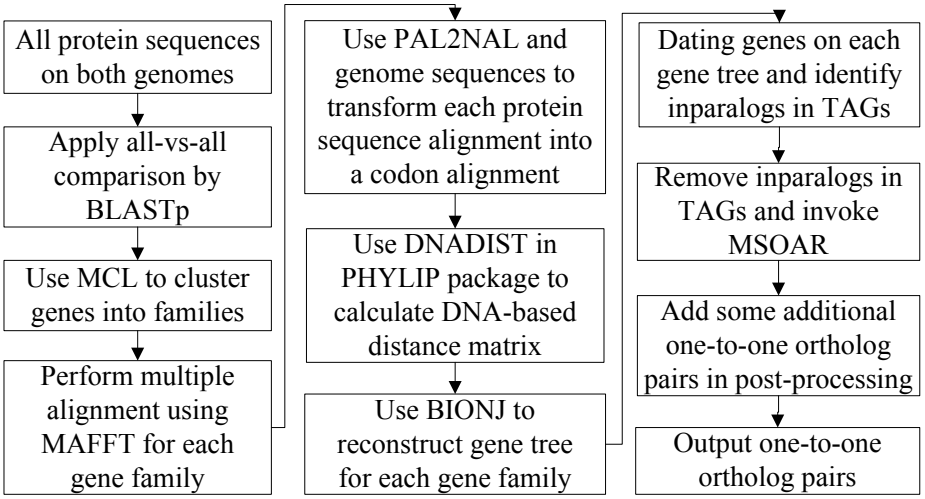


Figure 3.5: An outline of MSOAR 2.0.

3.3.3 Gene Family Definition and Construction

A gene family is defined to be the set of genes that are all descended from a common ancestral gene [7,34]. Given two input genomes, our improved system starts by constructing gene families for all the genes on both genomes. We mix all protein sequences on both genomes and calculate the pairwise similarity scores by applying an all-versus-all BLASTp comparison [48]. By analyzing the results of BLASTp, we obtain a square similarity matrix, whose elements contain sequence similarity measurements for each pair of proteins in the dataset. Gene families can be calculated using the MCL (Markov clustering) algorithm [49] with default parameters.

Based on probability and graph flow theory, MCL simulates random walks on a graph using Markov matrices to determine the transition probabilities among the vertices of the graph. Unlike many other protein sequence clustering algorithms, MCL is able to deal

with the presence of multi-domain proteins, promiscuous domains and fragmented proteins, making it one of the most widely used clustering algorithms in bioinformatics [49,50]. Some papers use MCL directly to identify ortholog groups such as OrthoMCL [22], while some others use TribeMCL (an extension of MCL) as a tool to find paralogs within a genome [46]. In our system, we apply MCL to cluster all homologous genes on both genomes (including all possible orthologs and paralogs) into gene families.

3.3.4 DNA-based Gene Tree Reconstruction

For each gene family, we perform multiple sequence alignment using MAFFT [51, 52] on the amino acid sequences of the genes and then calculate a DNA-based distance matrix. MAFFT is a rapid multiple sequence alignment tool based on fast Fourier transform, which has shown to be more accurate than other available tools including TCoffee [53] and ClustalW [54]. Moreover, MAFFT (with the fast mode) is able to align a large number (*e.g.*, several hundred) of sequences on a standard desktop PC in a few minutes.

Since DNA-based distance measure is shown to be more accurate than either protein-based distance or dS-based distance (*i.e.*, synonymous substitution rate) [37], we calculate the DNA-based distance for each gene family using the PHYLIP's DNADIST program [55] with the F84 nucleotide substitution model [56, 57]. To obtain DNA sequence alignments, we reverse translate the amino acid sequence of each gene into its corresponding codon sequence using the program PAL2NAL [58] and the given genome sequences, and then map the codon sequence onto its respective protein sequence alignment.

After getting the DNA-based distance matrix, we use the algorithm BIONJ [59,60]

to reconstruct a gene tree for each family. Not only is BIONJ the best neighbor-joining algorithm for phylogenetic reconstruction, it was found to have a competitive (if not better) accuracy as many other popular phylogenetic reconstruction methods including PHYML [61], MrBayes [62] and PAML [63] in genome-wide reconstruction of gene trees according to a recent study [37]. Although maximum-likelihood methods are known to be more accurate than distance-based methods in general phylogenetic reconstruction, we chose a distance-based method here mostly because of its efficiency since MSOAR 2.0 has to deal with many large gene families consisting of very long sequences on real data. In order to produce a rooted gene tree for each family, we introduce before BIONJ is run an artificial outgroup gene whose distance to each of the other genes in the family is twice the maximum distance in the original distance matrix. This can be achieved by simply adding a new row and a new column in the original distance matrix. Running BIONJ on this expanded distance matrix is equivalent to mid-point rooting [64].

3.3.5 Gene Duplication Dating on the Gene Tree

Once a gene tree is reconstructed, we need to label each of its internal nodes as either a duplication event or a speciation event. This process is a special case of the *gene duplication dating* problem, or the problem of reconciling a gene tree with a species tree. The phylogenetic tree reconciliation problem has been studied extensively in the literature, and many exact and heuristic algorithms have been proposed (see, *e.g.*, [65]). In our case, since only two species are involved, we propose a straightforward algorithm to date the duplication events in linear time.

To avoid postulating unnecessary gene losses, every internal node with descendant genes from the same species is labeled as a duplication event. Then, the lowest internal nodes with descendant genes from both species are labeled as speciation events. All ancestral nodes of the speciation nodes must be labeled as duplication events since there are only two species. An example of such a gene duplication dating algorithm is shown in Figure 3.6.

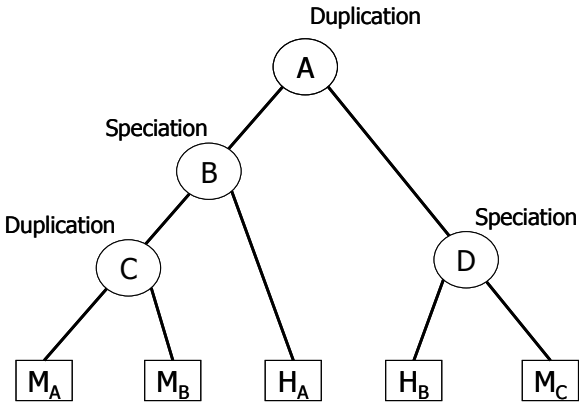


Figure 3.6: An example of the gene duplication dating algorithm. Node C is a duplication event since M_A and M_B are both from the same species. Node B and D correspond to speciation events since they have descendant genes from two species. Node A is a duplication event since it is the ancestral node of speciation nodes B and D.

3.3.6 Identification of Inparalogs in TAGs

After dating duplications in a gene tree, we may deem each set of genes duplicated after the speciation event as a potential set of inparalogs (*e.g.*, M_A and M_B in Figure 3.6). In order to confirm a potential set of inparalogs, we need to consider the positions of the genes on the concerned genome. If the potential inparalogs are adjacent to each other on the genome, *i.e.*, they appear in the same TAG, then we define them as inparalogs. For

each such set of inparalogs, at most one gene can be included in a one-to-one ortholog pair. Since these genes appear in tandem, it would make no difference to the RD distance (which is the objective function of MSOAR) which of them is chosen to represent the set in the one-to-one ortholog pair. Thus, we will keep the gene that has the highest similarity score against any gene in the other genome and remove the other inparalogs in the same set so they will not be considered by MSOAR later on. If some potential inparalogs are separated by other genes on the genome, they will all be kept at this step and dealt with by MSOAR later on.

3.3.7 Invocation of MSOAR and Post-Processing

After removing duplicated inparalogs in TAGs on each genome, MSOAR is now invoked on the remaining genes. To further improve the performance of MSOAR, we use a synteny-based post-processing step. If we consider the positions of the one-to-one orthologs assigned by MSOAR on each genome, we find that in many cases a large consecutive block (*i.e.*, synteny block) of assigned genes on one genome are orthologous to a consecutive block of assigned genes on the other genome with the same or reverse orientation. However, in some cases, there is a single unassigned gene (called a "gap") in each of the blocks forming an orthologous pair, and the gap appears at the same relative location in both blocks (see Figure 3.8 for an illustration). If the sequences of the two genes in the corresponding gaps are sufficiently similar (*e.g.*, at least one of the genes is the best hit of the other), then we deem the two genes as a one-to-one ortholog pair and add the pair to the output list.

3.4 Experimental Results

In order to test the performance of MSOAR 2.0, we apply it to both simulated and real data, and compare our results with MSOAR [34], the popular ortholog assignment tool InParanoid [16], Ensembl ortholog database [8] and the orthologs extracted from the whole-genome multiple alignment program MultiZ [66].

3.4.1 Simulation Results

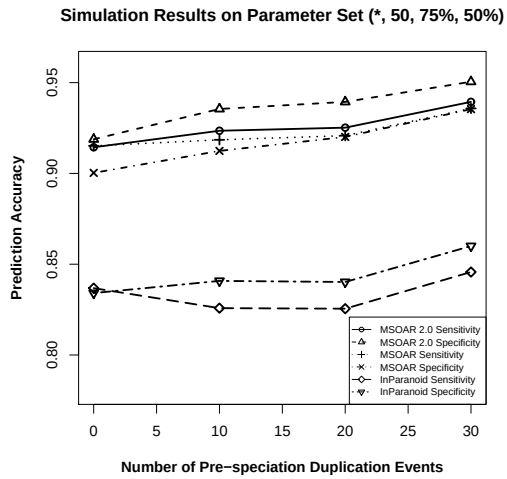
To assess the accuracy of one-to-one ortholog assignment, we simulate two input (single-chromosomal) genomes by using duplications, reversals, and point mutations. The simulation is controlled by a set of 4 parameters (k, p, α, β), where k denotes the number of duplications in the ancestral genome before the speciation, p is the total number of genome-level evolutionary events (*i.e.*, duplications and reversals) on each genome after the speciation, α is the percentage of duplications among the p events, and β is the percentage of tandem duplications among all duplications.

The simulation is controlled by a set of 4 parameters (k, p, α, β) which are defined in the Simulation Results section. The simulation is performed as follows. We first generate an ancestral genome G with 100 genes, each of which is a random sequence of 3,000 nucleotides (*i.e.*, 1,000 codons). We randomly perform k duplications in G to obtain another genome H. Then, a speciation happens and the genome H evolves into two contemporary genomes H_1 and H_2. The evolution from genome H to each of the contemporary genomes involves p evolutionary events, including $p \cdot \alpha$ duplications and $p \cdot (1 - \alpha)$ reversals. Among all duplications, β of them are tandem (*i.e.*, we randomly choose a gene and insert its copy

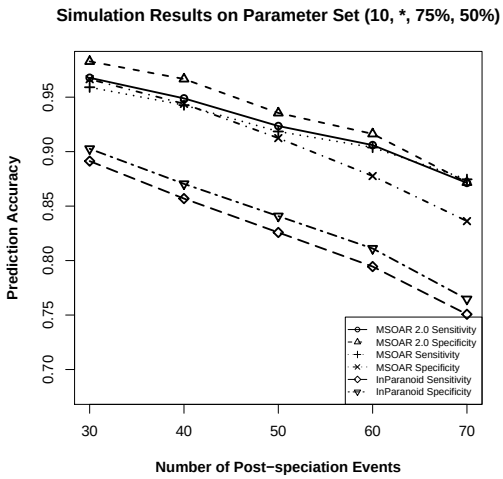
next to it) while the others are random (*i.e.*, we randomly choose a gene and insert its copy randomly into the genome). In order to simulate the sequence change of each gene along the evolutionary process, we set a constant mutation rate $\mu = 1\%$ to allow each gene on the genomes to have up to $3000\mu = 30$ random mutations of its nucleotides between every two evolutionary events (*i.e.*, 15 random nucleotide mutations would be performed on the average).

After generating two input genomes, we run MSOAR 2.0, MSOAR, and InParanoid separately. From the outputs of the three programs, we can easily compare their prediction accuracies in terms of sensitivity (*i.e.*, the number of true positive pairs assigned divided by the total number of assignable true positive pairs) and specificity (*i.e.*, the number of true positive pairs assigned divided by the total number of assigned pairs). Note that InParanoid actually outputs ortholog groups. For each ortholog group, we take the first pair of genes in the group as the one-to-one ortholog pair (which is referred to as the *main ortholog pair* in [4]).

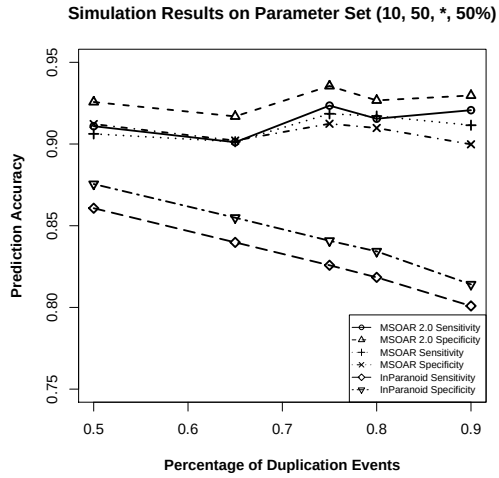
Since different parameters produce different input genomes, which may affect the prediction accuracies of the three programs, the parameters are varied as follows. We use a default parameter set and change the value of one parameter at one time. Based on recent studies on the relative ratios of various genome-level evolutionary events [46, 67], we choose to use $(10, 50, 75\%, 50\%)$ as our default parameter set. For each parameter set, 50 random datasets are simulated and the average prediction accuracies of the three programs are calculated. The performance of the three programs on various parameter sets are shown in Figure 3.7.



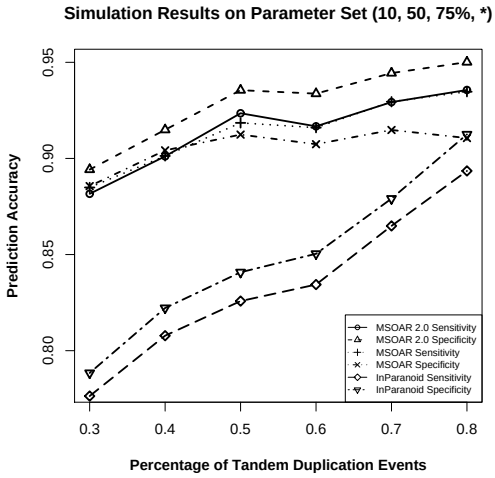
(a) Simulation results on the parameter set $(*, 50, 75\%, 50\%)$ where the parameter k is varied.



(b) Simulation results on the parameter set $(10, *, 75\%, 50\%)$ where the parameter p is varied.



(c) Simulation results on the parameter set $(10, 50, *, 50\%)$ where the parameter α is varied.



(d) Simulation results on the parameter set $(10, 50, 75\%, *)$ where the parameter β is varied.

Figure 3.7: Comparison of MSOAR 2.0, MSOAR and InParanoid on simulated data.

From Figure 3.7, we can see that parameter k has little effect on the prediction accuracies of the three programs as it only defines the number of outparalogs. Parameter p, on the other hand, has a great impact on the performance of all the programs. With the increase of p, the prediction accuracies of all the three programs sharply decrease. This is because when the number of evolutionary events increases, it is more difficult for MSOAR and MSOAR 2.0 to correctly reconstruct the evolutionary history based on the parsimony principle. Also orthologous genes may become less similar to each other for InParanoid to correctly identify them based on sequence similarity. Parameter α defines the ratio between duplications and reversals. As α goes up, the number of duplications increases while the number of reversals decreases. It becomes easier for MSOAR and MSOAR 2.0 to correctly identify reversals and assign one-to-one orthologs while it becomes harder for InParanoid to differentiate main orthologs from their duplicated inparalogs due to the large number of duplications. Parameter β defines the ratio between tandem duplications and random duplications.

As the ratio of tandem duplications goes up, the sensitivities of all three programs increase. This is due to the definition of true positives (TPs) in the simulation test. For each pair of orthologous TAGs from the two genomes, any pair of genes consisting of one gene from each TAG could be counted as a TP since these pairs are indistinguishable. However, at most one pair in the orthologous TAGs is counted as a TP. So, when the number of tandem duplications increases, all three programs output more TPs. As for specificity, since MSOAR 2.0 removes most of the inparalogs in TAGs based on the phylogenetic analysis and only the main ortholog pairs found by InParanoid are considered, the two programs do

not introduce more false positives (FPs) when the number of tandem duplications increases. Thus, the specificities of these two programs both increase. On the other hand, MSOAR may tend to assign more than one one-to-one ortholog pairs between two orthologous TAGs. This results in more FPs for MSOAR and an almost unchanged specificity.

The simulation results show that, in general, MSOAR 2.0 and MSOAR are more accurate than InParanoid in terms of both sensitivity and specificity on randomly simulated data. The sensitivity of MSOAR 2.0 is slightly better than that of MSOAR while its specificity is significantly ($2\% \sim 5\%$) higher than that of MSOAR. Note that the design of our simulation study was rather simplistic and the genomes simulated were not of real genome sizes. Hence, the above simulation results might not faithfully reflect the relative performance of InParanoid, MSOAR and MSOAR 2.0 on real data.

3.4.2 Real Data Results

In order to evaluate the performance of MSOAR 2.0 on real data, we apply MSOAR 2.0 to several real datasets. Since the human genome is the best annotated genome and has been used as the reference genome to assign gene symbols for other species, we use it as the "center" in our pairwise comparisons and compare it with four other mammalian genomes, mouse, rat, chimpanzee, and macaque that have been completely sequenced. The detailed procedure for downloading and pre-processing these genomes is described in the Methods section.

Protein sequences, transcripts, and gene locations for all five species, human (*Homo sapiens*), mouse (*Mus musculus*), rat (*Rattus norvegicus*), chimpanzee (*Pan troglodytes*) and

macaque (*Macaca mulatta*) (version 52, December 2008) were downloaded from Ensembl genome browser (http://www.ensembl.org/). Genes annotated as novel, supercontig, or mitochondrial are removed, and only protein-coding genes with known chromosome locations are kept. For genes with alternative splicing variants, we use their longest transcripts. Similar methods have been used in the previous studies [46, 68]. After such pre-processing, we obtained 21,164, 23,228, 22,490, 18,572, and 21,023 genes for human, mouse, rat, chimpanzee, and macaque, respectively.

Results on Human, Mouse and Rat

For the one-to-one ortholog assignments between human and mouse and between human and rat, Table 3.1 shows the contributions of each major step in MSOAR 2.0. The phylogenetic analysis step is able to identify more than 1,000 duplicated inparalogs in TAGs in each species (1,232/2,675 for human-mouse and 1,354/2,216 for human-rat), and remove most of them before MSOAR is invoked. Then one-to-one orthology is assigned by MSOAR on the remaining genes on each genome. Finally, in the post-processing step, MSOAR 2.0 is able to catch a few hundred one-to-one ortholog pairs (113 for human-mouse and 112 for human-rat) from the "gaps" between consecutive orthologous blocks on each genome.

In order to validate the prediction results of MSOAR 2.0, we choose to use gene symbols. Gene symbols are used by researchers to refer to a specific gene of interest across species. Each symbol for a species should be unique and each gene within a genome should be given only one approved gene symbol [69]. The nomenclature of a gene is done by the nomenclature committees for each species. At present, there are only three offi-

Table 3.1: Contributions of the major steps in MSOAR 2.0.

Pair of Species	Inparalogs in TAGs Identified by Phylogenetic Analysis	Orthologs Assigned by MSOAR	Orthologs Assigned after Post-Processing
human vs mouse	1,232 / 2,675	16,661	16,774
human vs rat	1,354 / 2,216	15,830	15,942

cial nomenclature committees in the world, for human, mouse, and rat respectively. So only these three species have official gene symbols. To obtain the most accurate gene symbol lists, we download the most recent gene symbols for human, mouse, rat from HGNC (http://www.genenames.org/), MGI (http://www.informatics.jax.org/), and RGD (http://rgd.mcw.edu/) respectively, all of which are the official nomenclature committees for the involved species. Note that since some gene symbols were assigned using information from some orthology databases, we should take the validation results based on gene symbols with a grain of salt. However, everything considered, gene symbols may still be the best available benchmark for validating genome-wide one-to-one ortholog assignment results.

To compare the performance of MSOAR 2.0 with that of MSOAR, InParanoid, the Ensembl ortholog database, and MultiZ, we consider the gene symbols of each output ortholog pair. Some genes may not have official gene symbols. Some symbols may not be meaningful, *e.g.*, when they are composed of "LOC" and gene ID, or when the gene functions have not yet been validated. In the latter case, the genes only have transcript identifiers (*e.g.*, gene symbols with the prefix "OTTMUSG" or the suffix "RIK" in the mouse genome).

Table 3.2: Comparison of the performance of five programs using gene symbol validation. In order to assess the accuracy of InParanoid, we take the first pair of genes in each ortholog group (*i.e.*, the main ortholog pair of the group) as a one-to-one ortholog pair. For the Ensembl ortholog database, we directly download all the ortholog pairs from Ensembl Biomart Browser, which includes one-to-one, one-to-many, and many-to-many orthology relationships. In order to extract the orthology information from MultiZ, we download the whole-genome multiple alignment for human, mouse and rat from UCSC genome browser, and map the annotated genes to the alignment based on their coordinates on each genome.

Pair of Species	Program	Assignable	Total Assigned	True Positives	Unknowns	Sensitivity	Specificity
	InParanoid	14,341	16,058	13,216	1,394	92.16%	90.13%
human	Ensembl	14,341	20,670	13,619	2,850	94.97%	76.43%
vs	MultiZ	14,341	16,543	13,136	1,433	91.60%	86.94%
mouse	MSOAR	14,341	16,769	13,528	1,554	94.33%	88.91%
	MSOAR 2.0	14,341	16,774	13,625	1,551	95.01%	89.50%
	InParanoid	12,688	15,197	11,750	1,529	92.61%	85.97%
human	Ensembl	12,688	18,814	12,004	2,490	94.61%	73.54%
vs	MultiZ	12,688	16,102	11,600	1,570	91.42%	79.82%
rat	MSOAR	12,688	15,883	11,970	1,723	94.34%	84.53%
	MSOAR 2.0	12,688	15,942	12,085	1,765	95.25%	85.24%

For each pair of orthologs, if both genes have identical official gene symbols, we count it as a true positive pair (*i.e.*, *TP*). If the genes have different official gene symbols, we count it as a false positive pair (*i.e.*, *FP*). If only one gene in the pair has an official gene symbol and another gene on the other genome (which is not in the pair) has the same gene symbol, then this pair is also considered as a false positive pair. For all other cases, we deem the pair as an unknown pair and ignore it in the accuracy assessment. We also calculate the assignable true one-to-one ortholog pairs between two species by counting the number of identical gene symbols. The performance of the five methods validated using gene symbols is shown in Table 3.2. The actual one-to-one ortholog assignment results of MSOAR 2.0 as well as the raw data and the MSOAR 2.0 software source code can be downloaded from the MSOAR website (http://msoar.cs.ucr.edu/MSOAR2.0/).

Table 3.2 suggests that MSOAR 2.0 achieves the best sensitivity among the five programs although its specificity is slightly worse than that of InParanoid. A detailed analysis on the differences among the ortholog assignment results by these programs is given in Table 3.3.

Since InParanoid is a sequence similarity based method, it produces ortholog groups solely based on sequence similarity. In order to compare the performance of InParanoid with MSOAR 2.0 properly, we take the first pair of each ortholog group output by InParanoid, *i.e.*, the main ortholog pair [4], as the one-to-one ortholog pairs assigned by InParanoid. As a result, all of the main ortholog pairs assigned by InParanoid are BBHs. Although many of the true one-to-one ortholog pairs may be indeed BBHs, some of them are not. In fact, more than 80% of the true one-to-one ortholog pairs assigned by MSOAR

Table 3.3: Differences between the ortholog pairs assigned by MSOAR 2.0 and those by the other programs. (a) This column lists the number of TPs found by MSOAR 2.0 but missed by InParanoid. (b) This column lists the number of TPs in the previous column that are not BBHs. (c) This column lists the number of FPs found by Ensembl but not by MSOAR 2.0. (d) This column lists the number of FPs in the previous column that are inparalogs occurring in TAGs. (e) This column lists the number of FPs found by MSOAR but not by MSOAR 2.0. (f) This column lists the number of FPs in the previous column that are inparalogs occurring in TAGs.

Pair of Species	MSOAR 2.0 vs InParanoid		MSOAR 2.0 vs Ensembl		MSOAR 2.0 vs MSOAR	
	TPs in MSOAR 2.0 but not in InParanoid[a]	Not BBHs[b]	FPs in Ensembl but not in MSOAR 2.0[c]	Inparalogs in TAGs[d]	FPs in MSOAR but not in MSOAR 2.0[e]	Inparalogs in TAGs[f]
human vs mouse	487	408	2,997	2,664	330	312
human vs rat	429	400	2,681	2,366	311	299

2.0 but missed by InParanoid in the human-mouse and human-rat comparisons (408/487 for human-mouse and 400/429 for human-rat) are not BBHs as shown in Table 3.3 (the first two columns). An example from the human-mouse comparison can be seen in Figure 3.8. Here, the true one-to-one ortholog pair (ITIH2, Itih2) is missed by InParanoid since ITIH2 and Itih2 are not BBHs. But MSOAR 2.0 was able to catch this pair correctly.

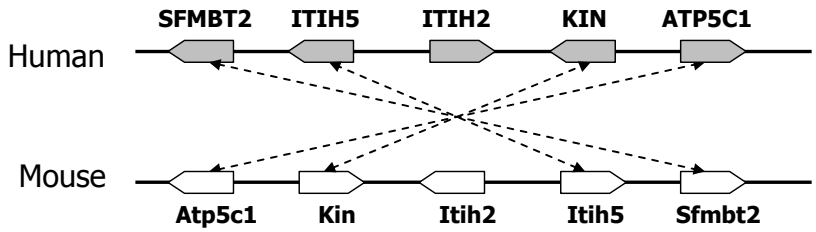


Figure 3.8: A real example of non-BBH true one-to-one ortholog pairs in the human-mouse comparison caught by MSOAR 2.0 in the post-processing step. Four one-to-one ortholog pairs were assigned by MSOAR between two corresponding orthologous blocks on human chromosome 10 (7,244,255bp-7,900,507bp) and mouse chromosome 2 (9,977,663bp-10,636,794bp). The genes ITIH2 and Itih2 were not assigned orthology by MSOAR, since ITIH2 is not among the top hits of Itih2. However, because Itih2 is the best hit of ITIH2 and the genes are located in corresponding "gaps", MSOAR 2.0 outputs them as an additional one-to-one ortholog pair.

While we mainly focus on finding the one-to-one orthology relationship between two genomes, the Ensembl ortholog database presents orthology in general as a many-to-many relationship. Thus, for each ortholog group, it outputs all pairs of genes consisting of one gene from one genome and another from the other. As a result, the specificity of the Ensembl ortholog database is quite low because each large ortholog group may result in many false positives. [1] What is interesting is that even though it outputs a large number of ortholog pairs, its sensitivity is still a little bit worse than that of MSOAR 2.0

[1]Hence, our measure of specificity is unfair to Ensembl since it treats orthology as a one-to-one relationship.

in both human-mouse and human-rat comparisons as shown in Table 3.2. It is interesting to observe that most of the false positive pairs output by Ensembl but not by MSOAR 2.0 (*i.e.*, 2,664/2,997 for the human-mouse comparison and 2,366/2,681 for the human-rat comparison) were actually found by MSOAR 2.0 to be inparalogs that appear in some TAGs, as shown in Table 3.3 (the two middle columns). See Figure 3.9 for an example of inparalogs in TAGs caught by MSOAR 2.0.

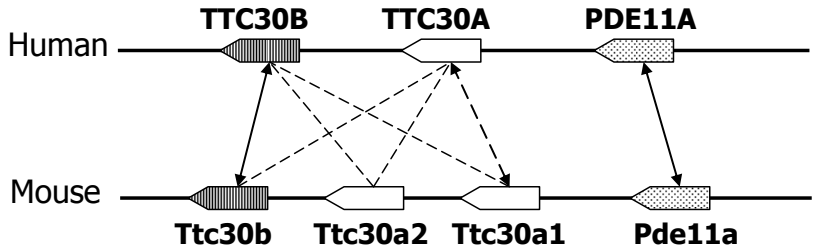


Figure 3.9: Comparison of ortholog assignments made by Ensembl, MSOAR and MSOAR 2.0 for the two segments of human chromosome 2 (178,123,219bp-178,685,428bp) and mouse chromosome 2 (75,773,906bp-76,192,000bp). Among the 7 pairs of genes illustrated in the figure, only (TTC30B, Ttc30b) and (PDE11A, Pde11a) are known one-to-one ortholog pairs according to gene symbols, as indicated by solid lines. Since the Ensembl ortholog database includes many-to-many relationship, it outputs 7 ortholog pairs, *i.e.*, (TTC30B, Ttc30b), (TTC30B, Ttc30a2), (TTC30B, Ttc30a1), (TTC30A, Ttc30b), (TTC30A, Ttc30a2), (TTC30A, Ttc30a1), and (PDE11A, Pde11a), introducing 5 false ortholog pairs, as indicated by dashed lines. MSOAR assigns three one-to-one ortholog pairs as indicated by the arrows in the figure, *i.e.*, (TTC30B, Ttc30b), (TTC30A, Ttc30a1), and (PDE11A, Pde11a), including one false one-to-one ortholog pair. MSOAR 2.0, however, identifies TTC30A as an inparalog of TTC30B on the human genome and Ttc30a2 and Ttc30a1 as inparalogs of Ttc30b on the mouse genome during the phylogenetic analysis of TAGs, and removes them before invoking MSOAR. Thus, MSOAR 2.0 only outputs two one-to-one ortholog pairs, *i.e.*, (TTC30B, Ttc30b) and (PDE11A, Pde11a), both of which are true positives.

The last two columns of Table 3.3 clearly demonstrate that MSOAR 2.0 achieves a better specificity than MSOAR because of its treatment of TAGs, since most of the false

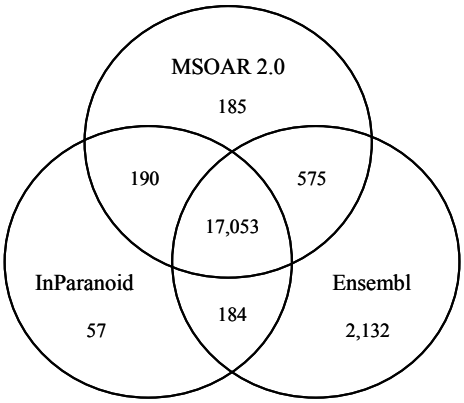
positives output by MSOAR but not by MSOAR 2.0 (312/330 and 299/311 for the human-mouse and human-rat comparisons, respectively) were identified as inparalogs in TAGs by MSOAR 2.0. For a detailed example where MSOAR 2.0 is able to catch false positives output by MSOAR, see again Figure 3.9.

MultiZ is generally viewed as a whole-genome multiple alignment program, but we can easily extract orthology information from the multiple alignment produced by MultiZ. To compare with the performance of MultiZ in one-to-one ortholog assignment, we download the human, mouse and rat genome alignment by MultiZ from UCSC genome browser, and map the annotated genes to the alignment according to their coordinates on each genome. If a gene contains several regions which are aligned to different locations belonging to different genes on another genome, then it forms a one-to-many orthology relationship and all pairs are counted in the same way as we dealt with the Ensembl ortholog database. Table 3.2 shows that MultiZ is worse than InParanoid in both sensitivity and specificity. Since both methods are based on sequence similarity, we will not include MultiZ in further comparative studies.

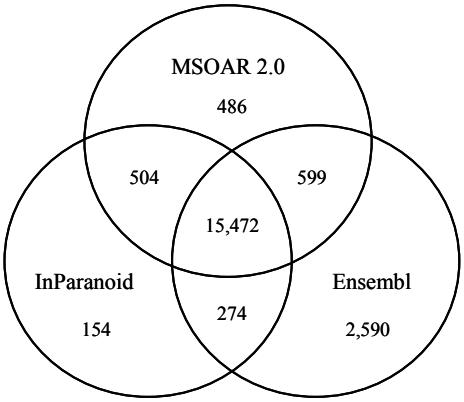
Results on Human, Chimpanzee and Macaque

Since chimpanzee and macaque do not have official gene symbols, we only compare our assignment results with those of InParanoid and the Ensembl ortholog database. Figures 3.10 uses Venn diagrams to show the commonality and difference among the ortholog pairs assigned by MSOAR 2.0, InParanoid, and the Ensembl ortholog database. We see that the three programs share more than 75% of the ortholog pairs. InParanoid outputs the

least number of unique ortholog pairs while Ensembl has the most. More than 70% of the ortholog pairs unique to Ensembl are found to be inparalogs in TAGs (results not shown).



(a) Orthologs assigned between human and chimpanzee.



(b) Orthologs assigned between human and macaque.

Figure 3.10: Orthologs assigned by MSOAR 2.0, InParanoid and Ensembl on human, chimpanzee and macaque.

Table 3.4 shows the number of ortholog pairs output by MSOAR 2.0 that are shared by at least one of the other two programs. We observe that the closer the compared species is to human, the more support the result of MSOAR 2.0 receives from the other programs. For a pair of very closely related species, such as human and chimpanzee, the one-to-one ortholog pairs assigned by MSOAR 2.0 have nearly 99% support from at least one of the other two programs, which is consistent with our expectation, and confirms that MSOAR 2.0 is a highly accurate tool for one-to-one ortholog assignment between closely related species.

Finally, we also observe that the number of inparalogs found in human by MSOAR

Table 3.4: Support of the MSOAR 2.0 one-to-one ortholog pairs by the other two programs.

Support	human vs chimpanzee	human vs macaque	human vs mouse	human vs rat
By both programs	94.72%	90.69%	89.93%	87.71%
By at least one program	98.97%	97.15%	96.98%	96.48%

Table 3.5: Inparalogs found in human and the other species by MSOAR 2.0.

Inparalogs found by MSOAR 2.0	human vs chimpanzee	human vs macaque	human vs mouse	human vs rat
Inparalogs in human	3,161	4,103	4,390	5,222
Inparalogs in the other species	569	3,962	6,454	6,548

2.0 increases with the increase of evolutionary distance between human and the other species, as shown in Table 3.5. This is consistent with the definition of inparalogs.

3.5 Conclusion and Discussion

In this chapter, we have incorporated a new gene duplication model, the tandem duplication model, into MSOAR, and developed an improved system of one-to-one ortholog assignment by combining gene phylogeny and genome rearrangement. By comparison with MSOAR, InParanoid, the Ensembl ortholog database, and MultiZ on both simulated and real data, we showed that MSOAR 2.0 achieves the best sensitivity while maintaining a high specificity. Although MSOAR 2.0 has a slightly lower specificity as compared to InParanoid on real data using gene symbols as the benchmark (*e.g.*, in the human-mouse

comparison, 90.13% for InParanoid vs. 89.50% for MSOAR 2.0), it nevertheless identified several hundred of true one-to-one ortholog pairs that were missed by InParanoid. Because the majority of the "missed" one-to-one orthologs are not BBHs, which are what the InParanoid assignment is based on, MSOAR 2.0 clearly addresses a weakness of InParanoid. Moreover, MSOAR 2.0 shows a better specificity in the simulation tests. Note that MSOAR 2.0 also reconstructs the evolutionary history in terms of gene duplication and genome rearrangement, which could be of independent interest. Although Ensembl tends to assign a higher number of ortholog pairs than both InParanoid and MSOAR 2.0, MSOAR 2.0 outperforms it in terms of not only specificity but also sensitivity.

We evaluated the performance of the programs by computer simulations and gene symbols. However, simulations could be limited because the real evolutionary processes are much more complicated than what we can simulate. Furthermore, the use of gene symbols is not always feasible as many species do not have standard gene symbol assignment. We need to develop additional validation methods such as incorporating other available information, *e.g.*, gene functions. In addition, with the discovery of more mechanisms of gene evolution, new models of gene duplication (*e.g.*, segmental duplications) and genome operations (*e.g.*, *double cut and join* or DCJ), have been proposed. How to incorporate these new gene duplication models and operations into MSOAR 2.0 is our next challenge.

Chapter 4

Accurate Identification of Ortholog Groups among Multiple Genomes

4.1 Definition of Ortholog Groups

In the previous chapter, we described an improved system to identify ortholog pairs between two genomes. When we consider multiple genomes, however, orthology relationship becomes much more complicated because of the interleaving between speciation and gene duplication events. In this chapter, we extend the one-to-one orthology relationship between a pair of genomes defined in the previous chapter to multiple genomes in a straightforward way and define an *ortholog group* for a given set of genomes as a maximal set of genes (from different genomes) that are the direct descendants of the same ancestral gene.

Note that the genes in such an ortholog group are not separated by any gene duplication. Hence, this definition, although a bit stringent, is faithful to the original

definition of orthology in Ref. [1]. For example, according to this definition, there are 4 ortholog groups in Figure 4.1(b): $(\alpha_{4,1}, \alpha_{5,1}, \alpha_{7,1})$, $(\alpha_{4,2}, \alpha_{5,2})$, $(\alpha_{4,3})$, $(\beta_{6,1}, \beta_{7,1})$. We note in passing that other more general definitions of ortholog groups have been considered in the literature and used in popular orthology databases such as COG [5] and EnsemblCompara [8]. In these definitions, orthology is considered as a many-to-many relationship and thus paralogs (*i.e.*, genes that are separated by duplications) are often allowed in an ortholog group. We prefer treating orthology as a one-to-one relationship because it makes the presentation of the chapter simpler and validation of our results cleaner. Moreover, the one-to-one orthology relationship can be thought of as a refinement of the more general many-to-many relationship.

4.2 MultiMSOAR 2.0

4.2.1 Motivation

In order to find the one-to-one ortholog groups among multiple genomes, we develop a system called MultiMSOAR 2.0, which is an extension of MSOAR 2.0 introduced in the previous chapter.

There is a previous program called MultiMSOAR [70], which is an extension of MSOAR. MultiMSOAR tries to assign orthologs among multiple genomes by using a simple clustering method based on the pairwise results of MSOAR [34]. However, the MultiMSOAR program can actually handle only three genomes well. When more genomes are involved, MultiMSOAR may not find ortholog groups accurately because it does not take into ac-

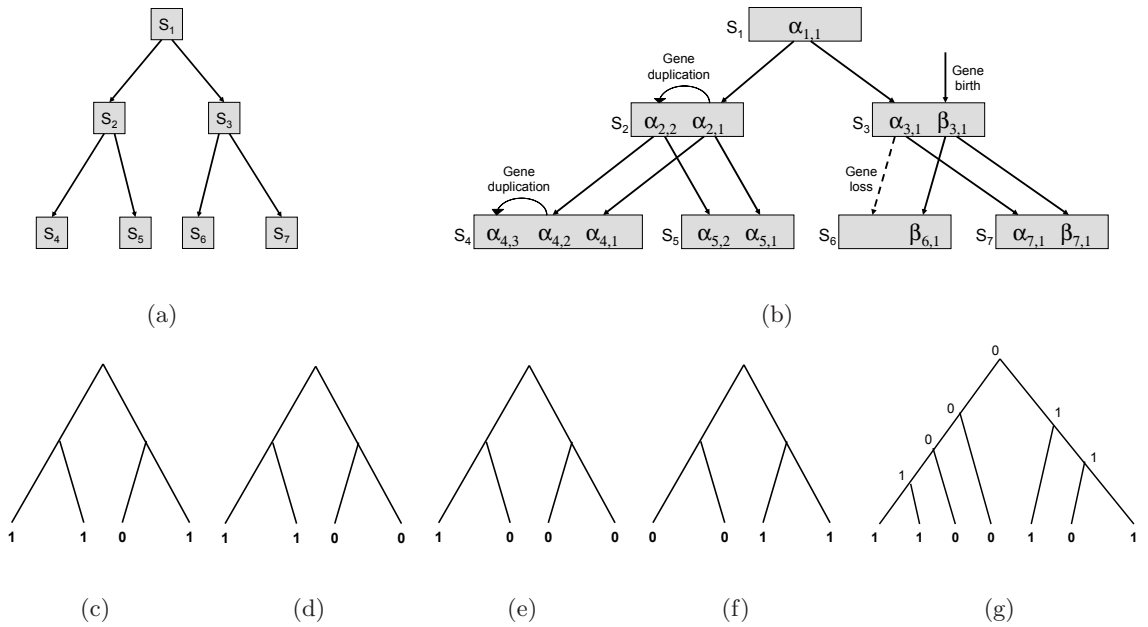


Figure 4.1: An illustration of genome evolution and corresponding TOGs. (a) The species tree for four species: S_4, S_5, S_6, S_7. (b) An example of genome evolution for the four species in (a). (c) The TOG for genes $\alpha_{4,1}, \alpha_{5,1}, \alpha_{7,1}$ in (b). (d) The TOG for genes $\alpha_{4,2}, \alpha_{5,2}$ in (b). (e) The TOG for gene $\alpha_{4,3}$ in (b). Note that, in this chapter, we will only be interested in ortholog groups containing at least two genes, and singleton ortholog groups will be ignored since they consist of only inparalogs from individual genomes. (f) The TOG for genes $\beta_{6,1}, \beta_{7,1}$ in (b). (g) An example of a TOG labeling. The labeling suggests two ortholog groups in the TOG, one consisting of two genes from the two leftmost species and the other two genes from the last three species.

count the phylogenetic relationship among the genomes. Furthermore, MultiMSOAR only considers those ortholog clusters that do not have gene losses in any species to be ortholog groups. This constraint might be acceptable for three closely related species, but it is too stringent when considering more species, since we expect to see many gene births and losses as well as duplications in the evolutionary history. As a consequence, we should allow gene losses within an ortholog group and ortholog groups to be composed of genes from a subset of the genomes.

Compared with MultiMSOAR, MultiMSOAR 2.0 allows gene losses within an ortholog group and ortholog groups involving genes only from a subset of the genomes. It also attempts to minimize the number of gene births, losses and duplications within a gene family when assigning ortholog groups. Moreover, compared with many other ortholog assignment tools for multiple genomes, MultiMSOAR 2.0 can provide more information about genome evolution in terms of gene births, losses as well as duplications.

4.2.2 An Outline of MultiMSOAR 2.0

An outline of MultiMSOAR 2.0 is shown in Figure 4.2. In short, MultiMSOAR 2.0 constructs gene families for all the genomes first by using sequence similarity search (*i.e.*, BLASTp) and the clustering algorithm MCL as done in Ref. [71]. Then it applies MSOAR 2.0 to find ortholog pairs between all pairs of genomes. After that, it builds a weighted multipartite graph using the pairwise orthology information and sequence similarity between each pair of orthologs and attempts to find a maximum weight matching for each gene family. Then it partitions each family into a set of disjoint sets of orthologous genes (called *super*

ortholog groups or *SOGs*) such that each SOG contains at most one gene from each genome. Each such SOG may potentially consist of several ortholog groups. In order to partition a SOG into ortholog groups, MultiMSOAR 2.0 labels the leaves of the species tree using 1 or 0 to indicate if the SOG contains a gene from the corresponding species or not. The resulting tree is called a *tree of ortholog groups* (or *TOGs*). MultiMSOAR 2.0 then employs one of the two algorithms devised in this chapter (called the *NodeCentric* and *TreeCentric* algorithms) to label the internal nodes of each TOG based on the parsimony principle and some biological constraints. Ortholog groups can then be trivially identified from each fully labeled TOG. The details of each of the main steps in Figure 4.2 are explained below. Note that each ortholog group found by MultiMSOAR 2.0 is contained in some TOG but a TOG may contain several ortholog groups. An example is shown in Figure 4.1(g), where the TOG contains two ortholog groups and the second ortholog group contains a gene loss.

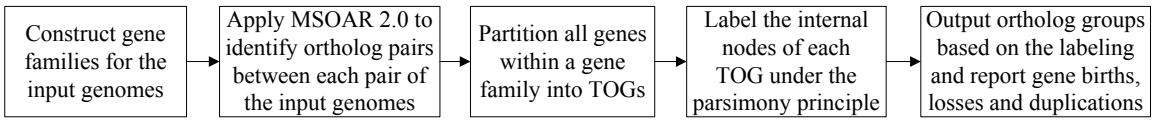


Figure 4.2: An outline of MultiMSOAR 2.0.

4.2.3 Homology Search and Gene Family Construction

Since we have multiple genomes, we define a gene family to consist of all homologous genes on all the genomes under study. As in Ref. [34, 71], only protein coding genes will be considered. For genes with alternative splicing variants, we use their longest transcripts. Similar methods have been used in previous studies [46, 71]. To cluster all the

genes into gene families, we combine all protein sequences from all genomes together, and perform an all-vs-all BLASTp homology search [48]. Then we use the popular clustering program MCL [49] to construct gene families. Similar methods have been used in many other papers [7, 22, 46].

4.2.4 Pairwise Genome Comparison

Since we try to identify ortholog groups among multiple genomes based on pairwise comparison, the prediction accuracy of ortholog pairs between two genomes is critical for the performance of our multiple genome system. MSOAR 2.0 has shown to be the most accurate prediction tool for assigning one-to-one ortholog pairs between two closely related genomes [71]. So, it is preferable to use the output of MSOAR 2.0 as the input to our current system. For a comparison among S genomes, we apply MSOAR 2.0 to all pairs of the S genomes, and use the $S*(S-1)/2$ pairwise comparison results to define a multipartite for each gene family to be partitioned in MultiMSOAR 2.0.

4.2.5 Partition of Each Gene Family into TOGs

In our definition of ortholog groups, each group may include at most one gene from each genome. However, a gene family may include many homologous genes from each genome (*i.e.*, paralogs), making it necessary to split the genes in a family into TOGs, such that each TOG contains at most one gene from every genome. This is done by employing a heuristic maximum weight S-dimensional matching algorithm as follows. Similar methods have been used in Ref. [70, 72].

Suppose we have S genomes, $G_1, G_2, ..., G_S$, where $S \geq 3$. For a given gene family, the number of genes from each genome are denoted as $n_1, n_2, ..., n_S$. We can construct an S-partite (or S-stage) graph G with n_i $(1 \leq i \leq S)$ vertices in the part corresponding to genome G_i (called stage i). We add edges to G by using the pairwise orthology information produced by MSOAR 2.0. Specifically, we add an edge between two vertices in G if and only if the corresponding two genes are from two different genomes and they are assigned as an ortholog pair by MSOAR 2.0. We assign a weight to such an edge, which is the BLASTp similarity score between the ortholog pair.

Since we would like to obtain a perfect S-dimensional matching with the maximum weight among the S stages, we need to add some dummy vertices to some of the stages in G to make them all have the same number of vertices. Let $N = \max_{1 \leq i \leq S} n_i$ be the maximum number of paralogs on any genome in the gene family. Then we add $N - n_i$ $(1 \leq i \leq S)$ dummy vertices to the i-th stage. The maximum (S-dimensional) matching problem for S-partite graphs (where $S \geq 3$) is known to be NP-hard [73], and N could be large for a real gene family when a large number of genomes are considered. So, we will use a heuristic optimization approach to find a good matching. Since the maximum weight matching for a bipartite graph can be computed by the Hungarian algorithm in cubic time [74], we first find a maximum weight bipartite matching for two stages in G, combine them into one stage, and apply the Hungarian algorithm iteratively on the remaining stages in G until only one stage is left. This results in a matching for the original S-partite graph G. This approach is very similar to the method used in MultiMSOAR [70], except that we use a post-order traversal on the species tree to decide the order that stages are combined. This

way, a stage is always combined with another stage that is close to it on the species tree. Another difference is that we use the bit score as the weight of an edge in G. If there is no edge between two vertices in different stages, we deem that there is an edge with weight 0 between them.

An example of the gene family partition is shown in Figure 4.1, where the figures in 4.1(c), 4.1(d), 4.1(e) represent 3 TOGs for the α gene family while Figure 4.1(f) represents a single TOG for the β gene family.

4.2.6 Labeling of TOGs

In order to identify ortholog groups within a TOG, we need to label the internal nodes (which correspond to ancestral genomes) using binary representations as well. Here, 1 means that the a gene is present in the corresponding ancestral genome while 0 means absence. Two constraints will be assumed:

1. *Intratree constraint*: If node u is labeled with a 0 and u has an ancestral node that is labeled with a 1, then every descendant node of u must be labeled with a 0.

2. *Intertree constraint*: Suppose that u and v are two nodes such that each of them is labeled with a 1 in at least one TOG. Then every node on the path connecting u and v must be labeled with a 1 in at least one TOG.

The intertree constraint makes sure that no gene is born twice in evolution, which is a commonly accepted hypothesis in molecular evolution since double gene birth events are extremely rare. The intratree constraint follows from the definition of orthology (that orthologs evolved through speciation only).

Among all the labelings of the TOGs satisfying the above two constraints, we would like to find one that minimizes the number of gene births, duplications and losses in the evolution of the family. Since each edge of a TOG whose nodes are labeled with 01 or 10 represents a gene birth/duplication or a gene loss, we need to find a parsimonious way to label the internal nodes so that the number of 01 or 10 edges is minimized. For simplicity, let us call a 01 or 10 change on an edge a *flip*.

We can now formulate the TOG labeling problem as a combinatorial optimization problem as follows:

TOG Labeling: *Given N TOGs, find a binary labeling of all the internal nodes of the TOGs so that both intratree and intertree constraints are satisfied and the total number of flips is minimized.*

The problem can be solved by a trivial exhaustive search algorithm that considers all possible labelings of the TOGs. However, since a binary tree with S leaves has $S-1$ internal nodes, this algorithm runs in time $O(2^{N\cdot(S-1)})$, which is impractical even if $N = S = 10$. We need to find more efficient solutions to this problem.

Before we proceed with our algorithms, we first prove the following two lemmas, which will help accelerate the speed of our labeling algorithm.

Lemma 1 *If two child nodes are labeled as* 1*, then in any optimal labeling, their parent node must be labeled as* 1*.*

Proof. Suppose that in an optimal labeling L, an internal node P is labeled as 0 in some TOG but both of its children are labeled as 1. If we change the label of P to 1,

the two constraints will not be violated, and there will be two fewer flips on the two edges from P to its two children. Even if this change might incur a new flip on the edge from P to its parent node, the total number of flips will still be reduced. This is a contradiction to the assumption that L is an optimal labeling, which completes our proof. □

Lemma 2 *If two child nodes are labeled as* 0*, then there is an optimal labeling, where their parent node is labeled as* 0*.*

Proof. Suppose that an internal node P of some TOG T is labeled as 1 while both of its children are labeled as 0 in some optimal labeling. If we change the label of P to 0, it is easy to see that the intratree constraint will not be violated. However, the intertree constraint might be violated if the node P is also labeled as 0 in all other TOGs. Then, according to Lemma 1, the two child nodes of P cannot be labeled as 1 at the same time in each of the other TOGs. If each of the two child nodes of P is labeled as 0 in all other TOGs, then we are safe to change the label of P from 1 to 0 in the TOG T since the change will not violate the intertree constraint. Otherwise, there is at least one TOG T', in which the two child nodes of P are labeled as 0 and 1, respectively. In this case, we can change the label of P in T' to 1. From the proof of Lemma 1, we know that changing the label of P in T will decrease the number of flips by at least 1, while changing the label of P in T' may increase the number of flips by at most 1. If we change the labels of node P is TOGs T and T' simultaneously, the total number of flips will not increase and thus the labeling is still optimal. Moreover, such a simultaneous change will keep the intertree constraint satisfied. This completes the proof of Lemma 2. □

The TOG labeling problem is trivial to compute without the intratree and intertree constraints. If we only consider the intratree constraint, the problem can still be solved by using dynamic programming in polynomial time. However, the intertree constraint makes the problem much harder. Here, we propose two different algorithms to solve the TOG labeling problem: the *NodeCentric* algorithm and the *TreeCentric* algorithm. The algorithms are sketched below.

The basic idea behind the NodeCentric algorithm is to label all N TOGs simultaneously by dynamic programming. In other words, it labels each internal node of the species tree with a binary vector of N bits. In order to keep track of the validity of the two constraints, we will use label $0'$ (when considering some TOG) to indicate that (i) the current node is labeled as 0 in the TOG and (ii) some descendant of the current node is labeled as 1 in the TOG. Thus, the label 0 now means that all descendant nodes are also labeled as 0. The algorithm proceeds in post-order. For each internal node u in the species tree, it enumerates all possible label vectors at u and for each vector, it computes the minimum number of flips in the subtree under node u by considering all feasible label vectors of its two children without violating the two constraints. By Lemmas 1 and 2, we can quickly fix the label of u in a TOG if the labels of its two children in the same TOG are both fixed as 0 or both fixed as 1.

Since the left and right children can be considered separately, it seems that the above algorithm would run in $O(S \cdot (3^N \cdot 3^N)) = O(S \cdot 9^N)$ time, which could be impractical if N is large. However, with a careful analysis, we find that at most 3 (instead of 9) combinations of the parent-child labels are possible in a TOG. If the parent label is fixed

as 0, then the child label must be fixed as 0 as well. Otherwise, the parent label could be $0'$ or 1. If it is $0'$, then the child label could be either fixed as 0 or one of $0'$ and 1. If the parent label is 1, then the child label must be fixed either as 0 or as 1 due to the intratree constraint. So, in any case, at most 3 combinations of the parent-child labels should be considered in a TOG and hence, a total number of 3^N values need to be computed. The intertree constraint may reduce the number of legal combinations even further. This implies an efficient implementation of the NodeCentric algorithm with time complexity $O(S \cdot 3^N)$.

While the NodeCentric algorithm goes through each node sequentially, the TreeCentric algorithm goes through each TOG sequentially. For a subset of fully labeled TOGs on the same species tree, the *union TOG* is a fully labeled TOG obtained by taking the Boolean *or* operation on the labels of each given TOG at the same node of the species tree. Let us order the TOGs arbitrarily as $T_1, T_2, \ldots, T_N$. For each TOG T_i, the TreeCentric algorithm enumerates all feasible binary labelings of the TOG T_i by taking into account the intratree constraint. This can be done efficiently by dynamic programming. For each such labeling of T_i, it enumerates all possible union TOGs T^i covering $T_1, T_2, \ldots, T_i$, and then computes and records the minimum number of flips in the TOGs $T_1, T_2, \ldots, T_i$ for each union TOG T^i, by taking advantage of the previously recorded minimum number of flips in $T_1, T_2, \ldots, T_{i-1}$ for each union TOG T^{i-1}. Finally, the minimum number of flips in all TOGs $T_1, T_2, \ldots, T_N$ is obtained by considering all possible union TOGs covering $T_1, T_2, \ldots, T_N$ and taking into account the intertree constraint. Since the number of different union TOGs is 2^{S-1}, the above algorithm runs in $O(N \cdot 4^{S-1})$ time.

More detailed pseudocodes of both algorithms are given in Algorithms 1 and 2. For the convenience of the reader, we list the notations used in the algorithms and their brief explanations explicitly below.

- T: the species tree.
- N: the number of TOGs in a gene family.
- T_i $(1 \leq i \leq N)$: the TOGs in the gene family.
- T^i $(0 \leq i \leq N)$: the union TOG covering TOGs $T_1, T_2, \ldots, T_i$.
- $l_u(T_i)$ $(1 \leq i \leq N)$: the label of node u in T_i.
- $l_u(T)$: the label vector of node u in T with N bits, where the i-th bit is $l_u(T_i)$ $(1 \leq i \leq N)$.
- $l(T_i)$ $(1 \leq i \leq N)$: the labeling of TOG T_i.
- $flip(l_u(T), l_v(T))$: the number of flips (*i.e.*, Hamming distance) between two labelings $l_u(T)$ and $l_v(T)$.
- $cost(l(T_i))$ $(1 \leq i \leq N)$: the number of flips in T when labeled as $l(T_i)$.
- $cost(u, l_u(T))$: the total number of flips in the subtree of T rooted at u with labeling $l_u(T)$.
- $cost(T^i)$ $(0 \leq i \leq N)$: the total number of flips in the first i TOGs when their labelings satisfy the intratree constraint and form the union TOG T^i.
- $l(T_i) \bigvee l(T_j)$: the boolean *or* operation between labelings $l(T_i)$ and $l(T_j)$.

Both algorithms NodeCentric and TreeCentric are exponential time algorithms. However, in practice, the number of genomes in comparison is expected to be small (usually $S \leq 15$). So we can use the TreeCentric algorithm to find an optimal TOG labeling

Algorithm 1 `NodeCentric` $(T_1, T_2, \ldots, T_N)$

1: Traverse T in post-order

2: **for all** node $u \in T$ **do**

3: **if** u is a leaf node **then**

4: $l_u(T) \leftarrow l_u(T_1)l_u(T_2)\cdots l_u(T_N)$

5: $cost(u, l_u(T)) \leftarrow 0$

6: **else**

7: **for all** possible labeling $l_u(T)$ at node u **do**

8: $cost(u, l_u(T)) \leftarrow \min\{cost(v, l_v(T)) + cost(w, l_w(T)) + flip(l_u(T), l_v(T)) + flip(l_u(T), l_w(T))\}$, where v, w are the two child nodes of u, and $l_v(T), l_w(T)$ are their labelings such that $l_u(T), l_v(T), l_w(T)$ satisfy the two constraints

9: **end for**

10: **end if**

11: **end for**

12: Traverse T in pre-order and retrieve the labeling of each node that gave rise to the minimum cost by a standard backtracing

Algorithm 2 `TreeCentric` $(T_1, T_2, ..., T_N)$

1: Initialize union TOG T^0 by labeling T with 0's

2: $cost(T^0) \leftarrow 0$

3: **for** $i \leftarrow 1$ to N **do**

4: **for all** union TOG T^i **do**

5: $cost(T^i) \leftarrow \infty$

6: **end for**

7: **for all** labeling $l(T_i)$ **do**

8: **if** $l(T_i)$ satisfies the intratree constraint **then**

9: **for all** union TOG T^i **do**

10: $cost(T^i) \leftarrow \min\{cost(T^i), cost(T^{i-1}) + cost(l(T_i))\}$, where $T^i = T^{i-1} \bigvee l(T_i)$

11: **end for**

12: **end if**

13: **end for**

14: **end for**

15: Let T^N_{opt} denote a union TOG that minimizes $cost(T^N_{opt})$ and satisfies the intertree constraint

16: Traverse the TOGs in reverse order (*i.e.*, $T_N, T_{N-1}, \ldots, T_1$) and retrieve the optimal labeling for each TOG T_i that gave rise to T^N_{opt} by a standard backtracing

efficiently. When the value of N is smaller, it is faster to apply the NodeCentric algorithm. Note that, the two algorithms may find different labelings for the same input, both of which are optimal.

4.2.7 Ortholog Group Identification

After labeling all TOGs, it is straightforward to identify ortholog groups. Starting from the root of each TOG, we can find the highest ancestral nodes labeled as 1. All genes at the descendent leaves of such an ancestral node form an ortholog group. An example is shown in Figure 4.1(g). In addition, with the labeling of each TOG, we can easily identify evolutionary events including gene births and losses as well as duplications. For each edge in the TOG, if the parent-child labeling is 1-0, then there is a gene loss. If the labeling is 0-1, and the parent node is labeled as 0 in all other TOGs, then it represents a gene birth. Otherwise, it represents a gene duplication.

4.3 Experimental Results

In order to test the performance of our system MultiMSOAR 2.0, we first apply it to simulated data, and compare it with the popular ortholog assignment tool MultiParanoid [17] for multiple genomes. For real data experiments, besides comparison with MultiParanoid, we also compare our results with Roundup [6], which is a well known multi-genome repository of orthology information and the Ensembl ortholog database.

4.3.1 Simulation Results

Our simulation test is an extension of the one in Ref. [71] for testing the performance of MSOAR 2.0. However, we now need to simulate more genome evolutionary events, including gene mutations, gene births, gene duplications, gene losses, genome rearrangements (including reversals, translocations, fusions and fissions, see examples in Figure 3.1) and speciations.

To make things easier, we only simulate the evolution of S ($S \leq 15$) single-chromosomal genomes as done in Ref. [71]. In order to generate S contemporary genomes, we first generate a random species tree T with S leaf nodes. Each internal node in T represents an ancestral genome while the leaf nodes represent the current genomes. Each edge in T represents a speciation event. We then randomly generate a genome with 100 genes consisting of 3,000 nucleotides each at the root of T. For each speciation event, we simulate E evolutionary events, which include α gene duplications, β gene births, γ gene losses, and $(1 - \alpha - \beta - \gamma)$ genome rearrangements. To generate the gene duplications, we randomly choose a gene, copy it and insert it into the genome next to the original copy or at a random position, depending on whether the duplication is tandem or random (here we assume 50% of all duplications are tandem, as done in Ref. [71]). To simulate the birth of a new gene, we create a new gene and randomly insert it into the genome. To simulate the loss of a gene, we randomly choose a gene and delete it from the genome. For genome rearrangements, since there is only one chromosome, only reversals are considered. Reversals are simulated by randomly choosing two positions on the genome and reverse all the genes between them.

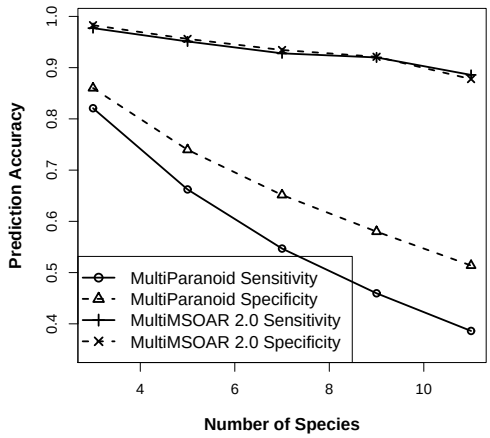
To simulate gene (point) mutations, we use a popular sequence simulation tool

evolver from the PAML package [75]. By running *evolver* with default options on the codon sequence at the root of a branch, we can obtain the mutated codon sequence over a pre-specified branch length μ. Since branch length can be measured in terms of the expected number of substitutions per site, we may use μ to control the mutation rate of a gene. We assume that between every two (genome-level) evolutionary events, all the genes on the existing genomes evolve at the same rate. In other words, a molecular clock is assumed.

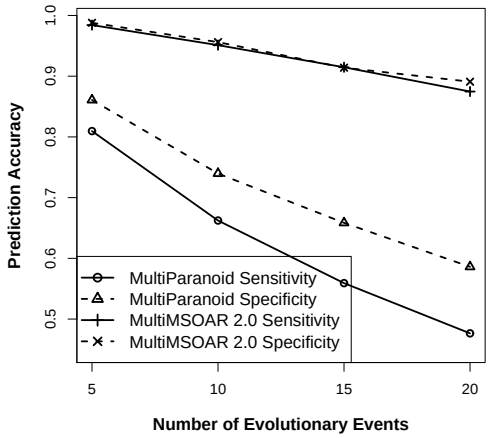
In summary, our simulation data is controlled by a 6-parameter set: $(S, E, \mu, \alpha, \beta, \gamma)$, where S is the number of species, E the total number of evolutionary events after each speciation, μ the gene mutation rate, and α, β, γ the percentages of gene duplications, births and losses among the E events, respectively.

To study the effects of different parameters on the performance of MultiMSOAR 2.0, we set the default values for each parameter as $S = 5, E = 10, \mu = 0.05, \alpha = 40\%, \beta = 10\%, \gamma = 10\%$, and we will vary one parameter at a time. To measure the prediction accuracy, we use two popular measurements: *sensitivity* and *specificity*. Here, sensitivity is defined as the number of the true ortholog groups (*i.e.*, true positives) identified by a program divided by the total number of known ortholog groups, and specificity is defined as the number of true ortholog groups identified divided by the number of ortholog groups output. We compare the ortholog groups found by MultiMSOAR 2.0 and MultiParanoid. In order for an identified ortholog group to be a true positive (*i.e., TP*), we require that all genes in the identified ortholog group match exactly with all the genes in a known ortholog group. For each parameter set, we generate 10 simulated data sets and run MultiMSOAR 2.0 and MultiParanoid on these data respectively. Finally we calculate the average prediction

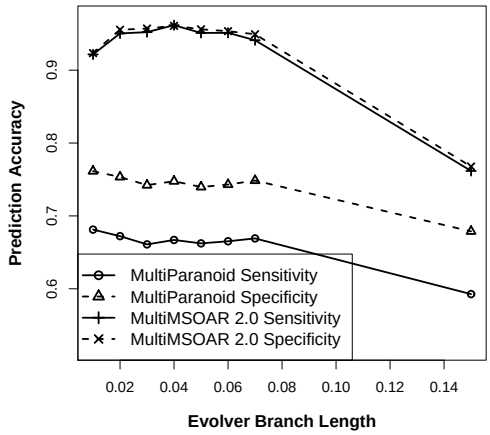
accuracies of the two programs on each parameter set. The prediction accuracies of the two programs are shown in Figure 4.3.



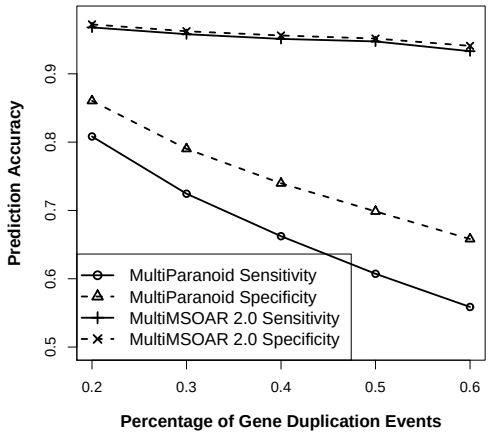
(a) Simulation results on the parameter set $(*, 10, 0.05, 40\%)$ where the parameter S is varied.



(b) Simulation results on the parameter set $(5, *, 0.05, 40\%)$ where the parameter E is varied.



(c) Simulation results on the parameter set $(5, 10, *, 40\%)$ where the parameter μ is varied.



(d) Simulation results on the parameter set $(5, 10, 0.05, *)$ where the parameter α is varied.

Figure 4.3: Comparison of MultiMSOAR 2.0 and MultiParanoid on simulated data.

Figures 4.3(a), 4.3(b), 4.3(d) show that with the increase of the number of species, the number of evolutionary events, and the number of gene duplications, the prediction accuracies of both programs decrease since it becomes harder for them to correctly identify ortholog groups. However, we notice that the decrease in accuracy for MultiMSOAR 2.0 is mild while the decrease is sharp for MultiParanoid, especially in Figure 4.3(d). This could be because when more genes are duplicated, it becomes increasingly difficult for MultiParanoid to decide if a duplication happened in an ancient genome or in a more recent genome. Thus, it might confuse some ancient duplications with recent duplications and miss calling some true ortholog groups. On the other hand, MultiMSOAR 2.0 infers the time of each duplication explicitly when labeling TOGs, and is thus more resilient to the increase of gene duplication events. However, since the labeling algorithm used in MultiMSOAR 2.0 is based on the parsimony principle and the optimal labeling might not be unique, the actual labeling given by MultiMSOAR 2.0 may not necessarily reflect the true evolutionary history. As a result, when the number of gene duplications increases, the prediction accuracy of MultiMSOAR 2.0 also decreases, but much more slowly than in the case of MultiParanoid.

Figure 4.3(c) is interesting and deserves some explanation. With the increase of the branch length μ defined in *evolver* from 0.01 to 0.04, both the sensitivity and specificity of MultiMSOAR 2.0 increase a little bit. This is because when μ increases, it becomes slightly easier for MultiMSOAR 2.0 to differentiate duplicated genes from their original copies based on sequence similarity. However, when μ goes from 0.07 to 0.15, the prediction accuracies of both programs sharply decrease. This is because the sequence similarity

between homologous genes originated from a common ancestral gene becomes weaker with the increase of μ. As a result, it becomes harder for MultiParanoid to identify ortholog groups solely based on sequence similarity, and for the MCL algorithm used in MultiMSOAR 2.0 to correctly cluster homologous genes into a gene family. Without correct gene families, we cannot expect MultiMSOAR 2.0 to find the ortholog groups correctly.

Generally speaking, from the four figures above, we can see that the prediction accuracy of MultiMSOAR 2.0 is significantly higher than that of MultiParanoid. With more species, more evolutionary events and more gene duplications, the advantage of MultiMSOAR 2.0 over MultiParanoid becomes more apparent. Besides, in the simulation, MultiMSOAR 2.0 is always able to achieve more than 90% prediction accuracy (in terms of sensitivity and specificity) as long as the gene mutation rate is not too high. This is pretty remarkable considering the large number of species and evolutionary events involved. Moreover, MultiMSOAR 2.0 can provide more information about gene births, losses and duplications in addition to identifying ortholog groups. In the simulation experiments, we also tested the accuracy of MulitMSOAR 2.0 in inferring gene births, losses and duplications, and compared its performance with Notung, a well-known software tool for reconciling genes trees with species trees by taking into account gene duplication and loss events [76,77]. Since Notung does not consider gene births, we only compare the sensitivity and specificity of MultiMSOAR 2.0 and Notung with respect to gene duplication and loss events. It turns out that the prediction accuracies of MultiMSOAR 2.0 on duplications and losses are generally much higher than those of Notung. Due to the page limit, the prediction accuracies concerning these events by MultiMSOAR 2.0 and Notung on simulated data are summa-

rized in Tables 4.1-4.4. Note that Notung fails to detect most gene losses because it prunes the species tree when an entire gene family is missing in a genome.

Table 4.1: Prediction accuracy when the parameter S (the number of species) is varied.

S	3	5	7	9	11
GeneBirth Sensitivity	95.00% / -	100.0% / -	100.0% / -	96.25% / -	87.00% / -
GeneBirth Specificity	71.00% / -	84.89% / -	84.64% / -	85.23% / -	93.90% / -
GeneDuplication Sensitivity	89.38% / 71.88%	89.06% / 61.56%	89.38% / 58.13%	87.66% / 55.47%	79.88% / 54.88%
GeneDuplication Specificity	99.33% / 59.61%	92.81% / 39.17%	92.70% / 22.65%	93.83% / 18.88%	94.64% / 17.10%
GeneLoss Sensitivity	47.50% / 0.00%	63.21% / 0.00%	66.67% / 0.83%	73.13% / 1.38%	84.63% / 2.00%
GeneLoss Specificity	93.33% / 0.00%	81.75% / 0.00%	77.63% / 0.48%	74.74% / 0.05%	52.74% / 0.11%

Table 4.2: Prediction accuracy when the parameter E (the number of evolutionary events) is varied.

E	5	10	15	20
GeneBirth Sensitivity	96.67% / -	100.0% / -	96.25% / -	96.88% / -
GeneBirth Specificity	76.38% / -	84.89% / -	76.78% / -	76.48% / -
GeneDuplication Sensitivity	91.88% / 60.00%	89.06% / 61.56%	85.83% / 62.08%	83.91% / 60.16%
GeneDuplication Specificity	96.91% / 37.61%	92.81% / 39.17%	95.17% / 32.69%	92.87% / 29.46%
GeneLoss Sensitivity	53.81% / 0.00%	63.21% / 0.00%	62.08% / 2.50%	45.42% / 2.59%
GeneLoss Specificity	74.17% / 0.00%	81.75% / 0.00%	69.92% / 0.13%	74.58% / 0.14%

Table 4.3: Prediction accuracy when the parameter μ (evolver branch length) is varied.

μ	0.01	0.03	0.05	0.07	0.15
GeneBirth Sensitivity	97.50% / -	97.50% / -	100.0% / -	98.75% / -	98.75% / -
GeneBirth Specificity	82.22% / -	79.75% / -	84.89% / -	78.44% / -	54.02% / -
GeneDuplication Sensitivity	86.56% / 61.88%	92.81% / 61.25%	89.06% / 61.56%	86.41% / 61.41%	95.00% / 58.75%
GeneDuplication Specificity	90.88% / 36.71%	93.80% / 37.96%	92.81% / 39.17%	92.94% / 37.48%	89.92% / 31.10%
GeneLoss Sensitivity	60.89% / 1.25%	65.00% / 0.00%	63.21% / 0.00%	59.55% / 0.00%	58.75% / 0.00%
GeneLoss Specificity	75.51% / 0.18%	89.11% / 0.00%	81.75% / 0.00%	82.79% / 0.00%	87.03% / 0.00%

4.3.2 Real Data Experiments

Since MultiMSOAR 2.0 is a tool to identify ortholog groups for multiple genomes that are closely related on a genome scale, to test its performance on real data, we choose

Table 4.4: Prediction accuracy when the parameter α (the ratio of duplication events) is varied.

α	20%	30%	40%	50%	60%
GeneBirth Sensitivity	98.75% / -	97.50% / -	100.0% / -	100.0% / -	97.50% / -
GeneBirth Specificity	79.00% / -	81.50% / -	84.89% / -	80.00% / -	80.44% / -
GeneDuplication Sensitivity	93.13% / 61.25%	93.33% / 61.67%	89.06% / 61.56%	86.00% / 61.25%	87.29% / 61.25%
GeneDuplication Specificity	94.44% / 37.80%	95.32% / 36.75%	92.81% / 39.17%	94.44% / 37.49%	95.31% / 38.21%
GeneLoss Sensitivity	71.25% / 0.00%	67.50% / 0.00%	63.21% / 0.00%	64.58% / 0.00%	60.71% / 1.25%
GeneLoss Specificity	96.90% / 0.00%	90.00% / 0.00%	81.75% / 0.00%	75.32% / 0.00%	78.18% / 0.56%

to use the mammalian genomes that have been completely sequenced. We downloaded seven mammalian genomes from the Ensembl genome browser (http://www.ensembl.org/): human (*Homo sapiens*), chimpanzee (*Pan troglodytes*), macaque (*Macaca mulatta*), mouse (*Mus musculus*), rat (*Rattus norvegicus*), cow (*Bos taurus*) and opossum (*Monodelphis domestica*) (version 57, March 2010). The species tree for the seven mammalian genomes is downloaded from Ensembl as well.

For the purpose of comparison, we choose to compare the results of MultiM-SOAR 2.0 with those of the popular tool MultiParanoid, Roundup and the Ensembl ortholog database. For MultiParanoid, we deem all the genes in the same cluster output by the program as an ortholog group assigned by MultiParanoid. We run MultiM-SOAR 2.0 and MultiParanoid on the real data sets respectively and compare their results. Roundup is a recently developed multi-genome repository of orthologs for over 250 genomes [6]. We download the ortholog groups for the concerned genomes from its website (http://roundup.hms.harvard.edu/). Since Roundup uses genomes from different sources, we need to map the genes used in Roundup to the corresponding genes used in Ensembl. For the Ensembl ortholog database, we download the reconciled EnsemblCompara gene trees, and extract the orthology information for the genomes being compared. Each group

of genes of the concerned genomes that descended from the lowest common ancestor of the concerned genomes defines an ortholog group.

Some other tools and databases are also available for ortholog assignment among multiple genomes, such as the OrthoFocus program [20] and the PhylomeDB ortholog database [13]. However, OrthoFocus is a program to identify orthologs in family-focused studies and it is inapproriate for genome-scale comparisons. PhylomeDB is a major source for phylogeny-based orthology and paralogy prediction, covering about 5 million proteins in 717 fully-sequenced genomes. However, since it involves a large number of genomes in the comparison, we are unable to retrieve reconciled gene trees concerning only genes from genomes of interest to us. Instead, we are only provided with orthology relationship with respect to a "seed" genome. This means that we would need to use a single-linkage method to combine ortholog groups via "seed" genomes, which is clearly undesirable. Besides, PhylomeDB generally presents orthology as a many-to-many relationship. Without reconciled trees, it is hard for us to refine the relationship into a one-to-one relationship, which makes the comparison with our results very difficult. Moreover, PhylomeDB uses a data source different from Ensembl, and the conversion of gene names between the two databases could be quite non-trivial.

Results on Human, Mouse and Rat

Since human, mouse and rat are the best annotated genomes, we can use gene symbols to validate the ortholog groups assigned among the three genomes by different programs. The same validation method has been used in many other papers [34, 70, 71].

Table 4.5: Performance of the four programs on human, mouse and rat.

Program	Assignable TPs	TPs	FPs	Unknowns	Total	Sensitivity	Specificity
MultiMSOAR 2.0	15,598	14,051	2,399	2,919	19,369	90.08%	85.42%
MultiParanoid	15,598	13,697	2,609	2,328	18,634	87.81%	84.00%
Ensembl	15,598	13,474	2,495	2,091	18,060	86.38%	84.38%
Roundup	14,616	10,094	2,424	6,790	19,308	69.06%	80.66%

Note that since some gene symbols were assigned using information from certain orthology databases, we should take the validation results based on gene symbols with a grain of salt. By using gene symbols, we can define true ortholog groups (TPs), false ortholog groups (FPs), and unknown ortholog groups as follows. If an ortholog group contains genes that have different gene symbols, then this group is counted as an FP. If at most one of the genes in the group have gene symbols, then this group is counted as an unknown. Otherwise, we treat the group as a TP. An ortholog group is defined as *assignable* if its genes appear in at least two genomes and have exactly the same gene symbol. We use the same measurements *sensitivity* and *specificity* as defined in the simulation to measure the prediction accuracies of the three programs. The performance of the programs is shown in Table 4.5.

The low sensitivity of Roundup in Table 4.5 may be caused by the mapping of gene IDs from Roundup to Ensembl since quite a few of the genes in Roundup were mapped to the unknowns in Ensembl. Nevertheless, we can see that MultiMSOAR 2.0 achieves the best sensitivity and specificity among all four programs. This is mainly because MultiParanoid only considers sequence similarity when assigning ortholog groups, while Ensembl ortholog groups tend to include lots of lineage-specific duplicated inparalogs. Though Roundup is

based on the reciprocal smallest distance algorithm, which is different from the reciprocal BLAST hits used in MultiParanoid, it fails to consider other information as well. In contrast, MultiMSOAR 2.0 combines gene order with sequence similarity, as well as phylogenetic information, and thus is able to make more accurate predictions.

Results on All Seven Mammalian Genomes

When comparing the seven mammalian genomes including human, chimpanzee, macaque, mouse, rat, cow, and opossum, we cannot validate the ortholog groups predicted by the three programs using gene symbols since not all of the genomes have been annotated with gene symbols. So, we only consider the common and different ortholog groups constructed by MultiMSOAR 2.0, MultiParanoid, Roundup and the Ensembl ortholog database. The comparison results are shown in Table 4.6 (since we are not able to find a good mapping from the data used in Roundup repository to the data used in Ensembl concerning all seven genomes, the comparison results with Roundup are not included in the table).

Table 4.6: Ortholog groups shared by MultiMSOAR 2.0, MultiParanoid and Ensembl on the seven mammalian genomes.

Programs	7 genomes	6 genomes	5 genomes	4 genomes	3 genomes	2 genomes
MultiMSOAR 2.0	12,034	3,772	1,337	584	875	3,195
MultiParanoid	11,397	3,311	1,127	609	800	2,728
Ensembl	13,566	2,002	493	270	363	991
MultiMSOAR 2.0 and MultiParanoid	9,075	2,237	633	239	348	1,483
MultiMSOAR 2.0 and Ensembl	8,722	1,003	225	104	131	524
MultiParanoid and Ensembl	8,438	983	237	117	143	587
All three programs	7,763	872	202	92	119	505

Table 4.6 shows the numbers of ortholog groups involving 2 to 7 genomes that were identified by MultiMSOAR 2.0, MultiParanoid and Ensembl. From Table 4.6, we can see that the numbers of ortholog groups found by all three programs are similar to each other for each number of genomes involved. Most of the ortholog groups identified by each of the three programs all involve seven genomes. Among such large ortholog groups identified by each program, more than a half (7,763) are shared by all three programs, which provides an indirect support for the ortholog groups found by MultiMSOAR 2.0. The large number of ortholog groups involving all seven genomes found by the three programs also manifests the evolutionary closeness of the seven mammalian species. The number of ortholog groups involving 4 genomes found by the three programs is pretty small here, since there is no subtree in the species tree consisting of exactly four species. Hence, an ortholog group of size four would have to involve gene losses. Since there is only one subtree consisting of three species (*i.e.*, human, chimpanzee, and macaque), most of the 875 ortholog groups of size 3 found by MultiMSOAR 2.0 (679, or about 77.6%) consist of genes from the three species. Similarly, 1,772/3,195 (55.46%) and 1,083/3,195 (32.49%) of the ortholog groups of size two consist of genes from mouse-rat and human-chimpanzee respectively, both of which are the closest pairs in the species tree.

4.4 Conclusion and Discussion

In this chapter, we have extended the pairwise ortholog assignment system MSOAR 2.0 to a multi-genome ortholog assignment system MultiMSOAR 2.0. By comparing with the well known multi-genome ortholog assignment tool MultiParanoid on simulated data,

we demonstrated that MultiMSOAR 2.0 achieves a significantly higher prediction accuracy. Our real data experiments on closely related mammalian genomes also show the superior performance of MultiMSOAR 2.0 over MultiParanoid, the multi-genome ortholog repository Roundup and the Ensembl ortholog database. Moreover, not only can MultiMSOAR 2.0 identify ortholog groups accurately, it can also provide accurate information about gene births, losses and duplications, which may shed additional insight on genome evolution.

Chapter 5

Conclusion

5.1 Main Contributions

In this dissertation, we presented new combinatorial approaches to accurate identification of orthologous genes between closely related genomes and developed two integrated systems MSOAR 2.0 [71, 78] and MultiMSOAR 2.0 [79, 80] for ortholog assignment.

MSOAR 2.0 is an improved system based on the previous system MSOAR by incorporating tandem gene duplication models explicitly. It combines phylogenetic approach to find tandemly duplicated inparalogs and filters them out before assigning ortholog pairs by MSOAR. It also catches some missing ortholog pairs by considering synteny information in the post-processing step.

MultiMSOAR 2.0 uses the pairwise orthology produced by MSOAR 2.0 as the input to construct a multiple graph for multiple genomes. Then a heuristic maximum weight matching algorithm is proposed to divide each gene family into a set of super ortholog groups

(*i.e., SOGs*) where each SOG contains at most one gene from each genome. For each such SOG, we label the leaves of the species tree using 1 or 0 to indicate if the SOG contains a gene from the corresponding species or not. The resulting tree is called a *tree of ortholog groups* (or *TOGs*). In order to reconstruct the evolutionary history for each TOG, two dynamic programming algorithms, namely NodeCentric and TreeCentric algorithms, are developed to label the internal nodes of each TOG under the parsimony principle and some biological constraints. Ortholog groups are then extracted from each fully labeled TOGs.

From our extensive experiments on both simulated and real data, we demonstrate that MSOAR 2.0 and MultiMSOAR 2.0 can achieve much better prediction accuracy than the other programs in comparison. In addition, MultiMSOAR 2.0 is able to provide more information about gene births, losses as well as duplications in evolution, which may be of independent biological interest.

5.2 Future Work

In MSOAR 2.0, we consider four basic genome rearrangement events, including reversal, translocation, fusion and fission. However, a unified genome rearrangement operation called *DCJ* (Double Cut and Join) has been proposed, which can be used as the basic operation to model all the other genome rearrangement events [81]. Given the unified operation of genome rearrangement, many algorithms to compute the rearrangement distance can be considerably simplified [82, 83]. Since MSOAR 2.0 also tries to minimize the rearrangement and duplication distance between two species when assigning orthologs, introducing the DCJ operation to MSOAR 2.0 may greatly simplify the heuristic algorithms

to calculate the RD distance currently used in the system.

In MultiMSOAR 2.0, although we extend the pairwise genome comparison to multiple genomes, our system still relies on the pairwise orthology results produced by MSOAR 2.0 as the input. If there are S species in comparison, we need to perform $S*(S-1)/2$ pairwise genome comparisons first, which turns out to be not sufficient if S is large. It is of interest to find an appropriate approach to get rid of the pairwise comparison, and to compare all S genomes simultaneously while still achieving high prediction accuracy.

The TOG information computed in MultiMSOAR 2.0 provides a good view of the evolutionary history of the genes present in each TOG. According to the labeling of each TOG, we know which genes are present in an ancestral genome and which genes are absent. If we consider the conserved gene neighborhood in the current genomes and try to infer the gene neighborhood in an ancestral genome, just as Ma *et al.* did in [84, 85], then we will be able to reconstruct the ancestral genomes, which is of great interest in the evolutionary studies.

Bibliography

[1] Fitch WM (1970) Distinguishing homologous from analogous proteins. Syst Zool 19: 99–113.

[2] Huisman TH, Carver MF, Efremov GD (1996) In: A Syllabus of Human Hemoglobin Variants, The Sickle Cell Anemia Foundation, Augusta, GA, USA.

[3] Jiang Z, Michal J, Melville J, Baltzer H (2005) Multi-alignment of orthologous genome regions in five species provides new insights into the evolutionary make-up of mammalian genomes. Chromosome Res 13: 707-715.

[4] Remm M, Storm CE, Sonnhammer EL (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. Journal of Molecular Biology 314: 1041 - 1052.

[5] Tatusov RL, Natale DA, et al. (2001) The COG database: new developments in phylogenetic classification of proteins from complete genomes. Nucleic Acids Research 29: 22-28.

[6] DeLuca TF, Wu I, Pu J, Monaghan T, Peshkin L, et al. (2006) Roundup: a multi-genome repository of orthologs and evolutionary distances. Bioinformatics 22: 2044–2046.

[7] Li H, Coghlan A, Ruan J, et al. (2006) TreeFam: a curated database of phylogenetic trees of animal gene families. Nucleic Acids Research 34: D572-580.

[8] Vilella AJ, Severin J, et al. (2009) EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Research 19: 327-335.

[9] Jensen LJJ, Julien P, Kuhn M, von Mering C, Muller J, et al. (2008) eggNOG: automated construction and annotation of orthologous groups of genes. Nucleic acids research 36: D250–254.

[10] Muller J, Szklarczyk D, Julien P, Letunic I, Roth A, et al. (2010) eggNOG v2.0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations. Nucleic acids research 38: D190–195.

[11] Kriventseva EV, Rahman N, Espinosa O, Zdobnov EM (2008) OrthoDB: the hierarchical catalog of eukaryotic orthologs. Nucleic acids research 36: D271–275.

[12] Waterhouse RM, Zdobnov EM, Tegenfeldt F, Li J, Kriventseva EV (2011) OrthoDB: the hierarchical catalog of eukaryotic orthologs in 2011. Nucleic Acids Research 39: D283–D288.

[13] Huerta-Cepas J, Capella-Gutierrez S, Pryszcz LP, Denisov I, Kormes D, et al. (2011) PhylomeDB v3.0: an expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. Nucleic Acids Res 39: D556–D560.

[14] Altenhoff AM, Dessimoz C (2009) Phylogenetic and functional assessment of orthologs inference projects and methods. PLoS Comput Biol 5: e1000262+.

[15] Altenhoff AM, Schneider A, Gonnet GH, Dessimoz C (2011) OMA 2011: orthology inference among 1000 complete genomes. Nucleic acids research 39: D289–D294.

[16] Berglund AC, Sjölund E, et al. (2008) InParanoid 6: eukaryotic ortholog clusters with inparalogs. Nucleic Acids Research 36.

[17] Alexeyenko A, Tamas I, Liu G, Sonnhammer ELL (2006) Automatic clustering of orthologs and inparalogs shared by multiple proteomes. Bioinformatics 22: e9–15.

[18] Boyer F, Morgat A, Labarre L, Pothier J, Viari A (2005) Syntons, metabolons and interactons: an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data. Bioinformatics 21: 4209–4215.

[19] Deniélou YP, Boyer F, Sagot MF, Viari A (2008) Recovering isofunctional genes: a synteny-based approach. Actes des Journée Ouvertes de Biologie, Informatique et Mathématiques .

[20] Ivliev AE, Sergeeva MG (2008) OrthoFocus: program for identification of orthologs in multiple genomes in family-focused studies. J Bioinform Comput Biol 6: 811–824.

[21] van der Heijden R, Snel B, van Noort V, Huynen M (2007) Orthology prediction at scalable resolution by phylogenetic tree analysis. BMC Bioinformatics 8: 83+.

[22] Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: Identification of ortholog groups for eukaryotic genomes. Genome Research 13: 2178-2189.

[23] Wheeler DL, Barrett T, et al. (2006) Database resources of the National Center for Biotechnology Information. Nucleic Acids Research 34: D173-180.

[24] Goodstadt L, Ponting CP (2006) Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human. PLoS Comput Biol 2: e133.

[25] Han MV, Hahn MW (2009) Identifying parent-daughter relationships among duplicated genes. Pac Symp Biocomput : 114–125.

[26] Zheng XH, Lu F, Wang ZY, Zhong F, Hoover J, et al. (2005) Using shared genomic synteny and shared protein functions to enhance the identification of orthologous gene pairs. Bioinformatics 21: 703–710.

[27] Catchen JM, Conery JS, Postlethwait JH (2009) Automated identification of conserved synteny after whole-genome duplication. Genome research 19: 1497–1505.

[28] Jun J, Mandoiu II, Nelson CE (2009) Identification of mammalian orthologs using local synteny. BMC genomics 10: 630+.

[29] Hannenhalli S, Pevzner P (1995) Transforming men into mice (polynomial algorithm for genomic distance problem). In: FOCS '95. Washington, DC, USA: IEEE Computer Society.

[30] Kent WJ, Baertsch R, et al. (2003) Evolution's cauldron: Duplication, deletion, and rearrangement in the mouse and human genomes. PNAS 100: 11484-11489.

[31] Pevzner P, Tesler G (2003) Genome rearrangements in aammalian evolution: lessons from human and mouse genomes. Genome Research 13: 37-45.

[32] Semon M, Wolfe KH (2007) Rearrangement rate following the whole-genome duplication in teleosts. Molecular Biology and Evolution 24: 860-867.

[33] Chen X, Zheng J, Fu Z, et al. (2005) Assignment of orthologous genes via genome rearrangement. IEEE/ACM Trans Comput Biol Bioinformatics 2: 302–315.

[34] Fu Z, Chen X, Vacic V, et al. (2007) MSOAR: A high-throughput ortholog assignment system based on genome rearrangement. Journal of Computational Biology 14: 1160–1175.

[35] Kuzniar A, Vanham R, et al. (2008) The quest for orthologs: finding the corresponding gene across genomes. Trends in Genetics .

[36] Sankoff D (1999) Genome rearrangement with gene families. Bioinformatics 15: 909-917.

[37] Rasmussen MD, Kellis M (2007) Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes. Genome Research : gr.7105007+.

[38] Sharan R, Suthram S, Kelley RM, et al. (2005) Conserved patterns of protein interaction in multiple species. PNAS 102: 1974–1979.

[39] Bandyopadhyay S, Sharan R, Ideker T (2006) Systematic identification of functional orthologs based on protein network comparison. Genome Research 16: 428–435.

[40] Wu F, Mueller LA, Crouzillat D, et al. (2006) Combining bioinformatics and phylogenetics to identify large sets of single-copy orthologous genes (cosii) for comparative, evolutionary and systematic studies: a test case in the euasterid plant clade. Genetics 174: 1407–1420.

[41] Mao F, Su Z, Olman V, et al. (2006) Mapping of orthologous genes in the context of biological pathways: an application of integer programming. PNAS 103: 129–134.

[42] Ohno S (1970) Evolution by gene duplication : 1–160.

[43] Maere S, De Bodt S, Raes J, et al. (2005) Modeling gene and genome duplications in eukaryotes. PNAS 102: 5454–5459.

[44] Zhang J (2003) Evolution by gene duplication: an update. Trends in Ecology & Evolution 18: 292–298.

[45] Hurles M (2004) Gene duplication: the genomic trade in spare parts. PLoS Biol 2.

[46] Shoja V, Zhang L (2006) A roadmap of tandemly arrayed genes in the genomes of human, mouse, and rat. Molecular Biology and Evolution 23: 2134-2141.

[47] Pan D, Zhang L (2008) Tandemly arrayed genes in vertebrate genomes. Comparative and Functional Genomics 2008.

[48] Altschul SF, Gish W, Miller W, et al. (1990) Basic local alignment search tool. Journal of Molecular Biology 215: 403–410.

[49] Enright AJ, Van Dongen S, Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30: 1575-1584.

[50] Alexeyenko A, Lindberg J, et al. (2006) Overview and comparison of ortholog databases. Drug Discovery Today: Technologies 3: 137–143.

[51] Katoh K, Misawa K, Kuma Ki, et al. (2002) Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. Nucleic Acids Research 30: 3059–3066.

[52] Katoh K, Kuma K, Toh H, et al. (2005) Mafft version 5: improvement in accuracy of multiple sequence alignment. Nucleic Acids Research 33: 511–518.

[53] Notredame C, Higgins DG, Heringa J (2000) T-coffee: A novel method for fast and accurate multiple sequence alignment. Journal of Molecular Biology 302: 205–217.

[54] Chenna R, Sugawara H, Koike T, et al. (2003) Multiple sequence alignment with the Clustal series of programs. Nucleic Acids Research 31: 3497-3500.

[55] Felsenstein J (1995) Phylip (phylogeny inference package), version 3.57 c. Seattle: University of Washington .

[56] Kishino H, Hasegawa M (1989) Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from dna sequence data, and the branching order in hominoidea. Journal of Molecular Evolution 29: 170-9.

[57] Felsenstein J, Churchill GA (1996) A hidden markov model approach to variation among sites in rate of evolution. Molecular Biology and Evolution 13: 93–104.

[58] Suyama M, Torrents D, Bork P (2006) PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. Nucleic Acids Research 34: W609-612.

[59] Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4: 406-425.

[60] Gascuel O (1997) Bionj: an improved version of the nj algorithm based on a simple model of sequence data. Molecular Biology and Evolution 14: 685–695.

[61] Guindon S, Gascuel O (2003) A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic Biology : 696–704.

[62] Huelsenbeck JP, Ronquist F (2001) Mrbayes: Bayesian inference of phylogenetic trees. Bioinformatics 17: 754–755.

[63] Yang Z (1997) PAML: a program package for phylogenetic analysis by maximum likelihood. Comput Appl Biosci 13: 555–556.

[64] Hess PN, De Moraes Russo CA (2007) An empirical test of the midpoint rooting method. Biological Journal of the Linnean Society 92: 669-674(6).

[65] Chauve C, Doyon JP, El-Mabrouk N (2008) Gene family evolution by duplication, speciation, and loss. Journal of Computational Biology 15: 1043–1062.

[66] Blanchette M, Kent WJ, Riemer C, et al. (2004) Aligning multiple genomic sequences with the threaded blockset aligner. Genome Research 14: 708–715.

[67] Kidd JM, Cooper GM, Donahue WF, et al. (2008) Mapping and sequencing of structural variation from eight human genomes. Nature : 56–64.

[68] Friedman R, Hughes AL (2003) The temporal distribution of gene duplication events in a set of highly conserved human gene families. Molecular Biology and Evolution 20: 154-161.

[69] Wain HM, Bruford EA, Lovering RC, et al. (2002) Guidelines for human gene nomenclature. Genomics 79: 464–470.

[70] Fu Z, Jiang T (2008) Clustering of main orthologs for multiple genomes. J Bioinform Comput Biol 6: 573–584.

[71] Shi G, Zhang L, Jiang T (2010) MSOAR 2.0: Incorporating tandem duplications into ortholog assignment based on genome rearrangement. BMC Bioinformatics 11: 10.

[72] Vashist A, Kulikowski CA, Muchnik I (2007) Ortholog clustering on a multipartite graph. IEEE/ACM Trans Comput Biol Bioinformatics 4: 17–27.

[73] Kann V (1991) Maximum bounded 3-dimensional matching is max snp-complete. Inf Process Lett 37: 27–35.

[74] Kuhn HW (2005) The hungarian method for the assignment problem. Nav Res Log 52: 7-21.

[75] Yang Z (2007) PAML 4: Phylogenetic analysis by maximum likelihood. Mol Biol Evol 24: 1586–1591.

[76] Durand D, Halldorsson BV, Vernot B (2006) A hybrid micro-macroevolutionary approach to gene tree reconstruction. J Comput Biol 13: 320–335.

[77] Vernot B, Stolzer M, Goldman A, Durand D (2008) Reconciliation with non-binary species trees. J Comput Biol 15: 981-1006.

[78] Shi G, Zhang L, Jiang T (2009) MSOAR 2.0: Incorporating tandem duplications into ortholog assignment based on genome rearrangement. In: Proc LSS Comput Syst Bioinform Conf. August, 2009. volume 8, pp. 13-24.

[79] Shi G, Peng MC, Jiang T (2010) Accurate Identification of Ortholog Groups Among Multiple Genomes. In: Proc LSS Comput Syst Bioinform Conf. August, 2010. volume 9, pp. 166-179.

[80] Shi G, Peng MC, Jiang T (2011) MultiMSOAR 2.0: An Accurate Tool to Identify Ortholog Groups among Multiple Genomes. PLoS ONE 6: e20892+.

[81] Bergeron A, Mixtacki J, Stoye J (2006) A Unifying View of Genome Rearrangements. Algorithms in Bioinformatics : 163–173.

[82] Braga M, Stoye J (2009) In: Comparative Genomics, Berlin, Heidelberg, volume 5817, chapter 4.

[83] Chen X (2010) On sorting unsigned permutations by double-cut-and-joins. Journal of Combinatorial Optimization : 1–13.

[84] Ma J, Zhang L, Suh BB, Raney BJ, Burhans RC, et al. (2006) Reconstructing contiguous regions of an ancestral genome. Genome research 16: 1557–1565.

[85] Ma J, Ratan A, Raney BJ, Suh BB, Zhang L, et al. (2008) DUPCAR: reconstructing contiguous ancestral regions with duplications. Journal of computational biology : a journal of computational molecular cell biology 15: 1007–1027.