Maximizing synteny blocks to identify ancestral homologs

Guillaume Bourque[1], Yasmine Yacef[2] and Nadia El-Mabrouk[2]

[1] Genome Institute of Singapore, bourque@gis.a-star.edu.sg
[2] DIRO, Université de Montréal, mabrouk@iro.umontreal.ca

Abstract. Most genome rearrangement studies are based on the assumption that the compared genomes contain unique gene copies. This is clearly unsuitable for species with duplicated genes or when local alignment tools provide many ambiguous hits for the same gene. In this paper, we compare different measures of order conservation to select, among a gene family, the pair of copies in two genomes that best reflects the common ancestor. Specifically, we present algorithms to identify ancestral homologs, or exemplars [1], by maximizing synteny blocks between genomes. Using simulated data, we validate our approach and show the merits of using a conservative approach when making such assignments.

1 Introduction

Identifying homologous regions between genomes is important, not only for genome annotation and the discovery of new functional regions, but also for the study of evolutionary relationships between species. Once orthologous genes have been identified, the genome rearrangement approach infers divergence history in terms of global mutations, involving the displacement of chromosomal segments of various sizes. The major focus has been to infer the most economical scenario of elementary operations transforming one linear order of genes into another. In this context, inversion (or "reversal") has been the most studied rearrangement event [2, 3, 4, 5, 6], followed by transpositions [7, 8, 9] and translocations [10, 11, 12]. All these studies are based on the assumption that each gene appears exactly once in each genome, which is clearly an oversimplification for divergent species containing paralogous and orthologous gene copies scattered across the genome. Moreover, even for small genomes (viruses, bacteria, organelles) where the hypothesis of no paralogy may be appropriate, the assumption of a one to one correspondance between genes assumes a perfect annotation step. However, in may cases, the similarity scores given by the local alignment tools (such as BLAST or FASTA) are too ambiguous to conclude to a homology, and using different parameters and cut-off values may lead to different sets of orthologs.

The approach to identify homology described above only relies on local mutations; it neglects the genomic context of each gene copy which might provide additional information. For example, if two chromosomes are represented by the two gene orders "$badc$" and "$badceaf$", the two first a are more likely to be the two copies derived from the common ancestor, as they are preserving the

gene order context in the two chromosomes. Sankoff [1] was the first to test this idea with the *exemplar approach*. The underlying hypothesis is that in a set of homologs, there commonly exists a gene that best reflects the original position of the gene family ancestor. The basic concept of Sankoff's algorithm is to remove all but one member of each gene family in each of the two genomes being compared, so as to minimize the breakpoint or the reversal distance. Context conservation has also been used in the annotation of bacterial genomes [13] to choose, among a set of BLASTP best hits, the true ancestral copies, also called *positional homologs*. We now want to extend these ideas to other measures of gene order conservation such as conserved and common intervals [14, 15, 16, 17]. These alternatives measures generalize the breakpoint distance and similarly allow to compare a set of genomes. Moreover, they allow to study global genome evolution without focusing on a specific rearrangement model.

In this paper, we use the common and conserved interval criteria to identify the ancestral homologs. Generalizing the fact that gene copies that are surrounded by the same genes in different genomes are more likely to be the true ancestral copies, we identify ancestral homologs by maximizing blocks of synteny between genomes. In Section 2, we review some gene order measures and their use for genome rearrangement with gene families. In Section 3, we describe our method and present algorithms for ancestral homolog assignment. In Section 4, we analyze the permformance of our method using simulated data and show the effect of homolog assignment on the induced rearrangement distance.

2 Related work

In the rest of this paper, a gene family a will refer to all homologs (orthologs and paralogs) of a gene a among a set of genomes. Paralogs are copies inside the same genome that have evolved by duplication, while orthologs are copies among different genomes that have evolved by speciation. A genome will be considered single chromosomal and represented as a linear order of signed genes, where the sign represents the transcriptional orientation of the gene. A chromosomal *segment* $[a, b]$ is just the subsequence surrounded by the two genes a and b.

2.1 Genome rearrangement with gene families

Gene orders can be compared according to a variety of criteria. The *breakpoint distance* between two genomes G and H measures the number of pairs of genes a, b that are adjacent in one genome (contains the segment '$a\ b$') but not in the other (contains neither '$a\ b$' nor '$-b\ -a$'). *Rearrangement distances* measure the minimal number of genome rearrangements (inversions, transpositions, translocations$\cdots$) necessary to transform one order of genes into another.

Most work on rearrangement has been restricted to the comparison of genomes with no gene copies. A method that does takes into account duplications, but requires that the number of copies is the same in both genomes, has been presented by *Tang and Moret* [18]. Their approach relied on a straightforward enumeration of all possible assignments of homologs between two genomes. More

recently, Chen *et al.* [19] gave an NP-hard result for this problem under the reversal distance and presented an efficient heuristic based on a maximal cycle decomposition of the Hannenhalli and Pevzner breakpoint graph [10, 3]. Both of these studies are based on an evolutionary model assuming that all copies were present in the common ancestor and no duplication occurred after speciation (Fig. 1a). In many context, this assumption may be questionable.

Another approach relaxing the copy number constraint has been considered by Sankoff [1]. The *exemplar approach* consists in deleting, from each gene family, all copies except one in each of the compared genomes G and H, so that the two resulting permutations have the minimal breakpoint or reversal distance. The underlying evolutionary model is that the most recent common ancestor F of genomes G and H has single gene copies (Fig. 1b). After divergence, the gene a in F can be duplicated many times in the two lineages leading to G and H, and appear anywhere in the genomes. Each genome is then subject to rearrangement events. After rearrangements, the direct descendent of a in G and H will have been displaced less frequently than the other gene copies. Even though finding the positional homologs (called *exemplars* in [1]) has been shown NP-hard [20], Sankoff [1] developed a branch-and-bound algorithm that has been shown practical enough on simulated data. More recently, Nguyen *et. al.* [21] developed a more efficient divide-and-conquere approach.

The preceding model is based on the hypothesis of a unique ancestral copy for each gene family. However, in the more general case of an ancestral genome containing paralogs, for each gene family, not only one but many pairs of ancestral homologs have to be found (Fig. 1c). The exemplar approach can also be applied to this model. Indeed, by running the algorithm n times, n homolog assignments are made for the same gene family. Recently, Blin *et. al* [22] gave an NP-hard result and proposed a branch-and-bound exact algorithm to compute the breakpoint distance under this model.

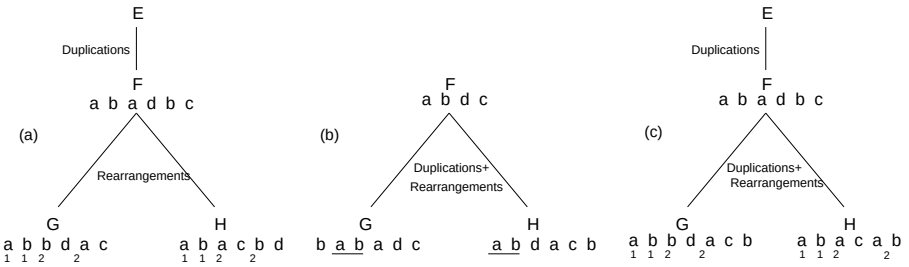


Fig. 1. (a) Evolutionary model considered in [19]; using the breakpoint distance, the chosen homologs are the one underlined by the same number in both genomes. (b) Model considered in [1]; using the breakpoint distance, the chosen exemplar are the underlined ones. (c) General model with duplications occurring before and after speciation; using the breakpoint distance and running the exemplar algorithm twice, the chosen homologs are the one underlined by the same number in both genomes.

2.2 Synteny blocks

The drawback of considering a rearrangement distance to compare genomes is the strong underlying model assuming evolution by one or two specific rearrangement events. A simpler measure of order conservation (synteny) is the breakpoint distance. Other more general measures of synteny have been proposed in the genome rearrangement literature [14, 16, 17] and are now being reviewed.

Conserved blocks The notion of *conserved intervals* or *blocks* that has been introduced in [14] is identical to the notion of a *subpermutation* introduced in the Hannenhalli and Pevzner theory [10]. It is defined for genomes with single gene copies as follows.

Definition 1. Given two genomes G and H, a *conserved block* is defined by two signed genes a and b and a set of unsigned genes U such that, in each genome, there exists a segment of the form $S = [a, b]$ or $S = [-b, -a]$, and the set of unsigned genes appearing between the two endpoints is U *(Fig. 2a)*. Such a conserved block will be denoted $[a, U, b]$.

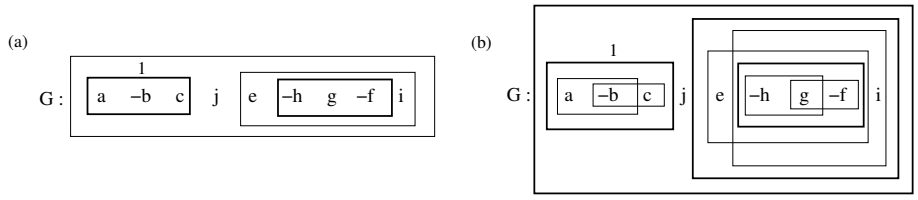


Fig. 2. The blocks of G and H, for H being the identity permutation $abcdefghij$. (a) Rectangles represent conserved blocks. For example, rectangle 1 represents the block $[a, U, c]$ with $U = \{b\}$. Bold rectangles are minimal blocks (not containing any other block); (b) Common blocks. For example, rectangle 1 represents the common block $\{a, b, c\}$. Bold rectangles are commuting blocks (either contained or have an empty intersection with any other block).

For genomes with gene copies, the problem of finding a pairing of gene copies that maximizes the number of conserved blocks (i.e. minimizing the conserved block distance) has been recently shown to be NP-complete [15].

Common blocks Even though conserved blocks have been shown useful for the genome rearrangement studies, the endpoint constraint contained in the definition is not directly linked to a specific biological mechanism. The notion of a common block introduced in [16] relaxes this constraint.

Definition 2. Let G and H be two genomes on the gene set $\{c_1, \cdots, c_n\}$. A subset C of $\{c_1, \cdots, c_n\}$ is a *common block* of G and H iff G (respec. H) has a segment which unsigned gene content is exactly C.

Common blocks have been considered as an additional criteria to improve the realism of genome rearrangement scenarios [17, 16].

3 Maximizing the blocks

Following the assumption that the true descendents of an ancestral gene in two genomes are the copies that have been less rearranged, the objective is to find a pairing of gene copies that maximizes gene order conservation. We use two measures of order conservation: the total number of conserved or common blocks.

There is a number of reasons to maximize the number of synteny blocks. First, the more blocks we can construct among a set of genomes, the farther they are from random permutations. Indeed, random orders would potentially contain no trace of gene order conservation, and have a single synteny block per genome. Second, they generalize the breakpoint criteria used in previous ancestral homolog assignment methods [1, 18, 19, 13]. Third, in contrast with rearrangement distances, they allow to model and compare, not only two genomes, but a set of genomes. Finally, although conserved blocks are not directly linked to a specific rearrangement event, they represent the components of the Hannenhalli and Pevzner graph [3, 10], and as such, are related to reversals.

It is preferable to measure similarity using the total number of blocks instead of the number of minimal or commuting blocks mostly because two overlapping blocks denote a better conservation than two disjoint blocks. Taking minimal blocks or commuting blocks alone does not reflect this difference. In contrast, maximizing the total number of blocks creates a bias towards overlapping blocks and tend to favour small local rearrangements, which is justified by a variety of biological and theoretical studies [23, 24].

3.1 Blocks for genomes with gene families

A *homolog assignment* is a procedure that connects, from each gene family, two particular gene copies, one from each genome. We generalize the notion of blocks (conserved or common) to two sequences containing gene copies as follows.

Definition 3. Let G be a genome on the gene family set $\{c_1, \cdots, c_n\}$. An *individual common* or *conserved* block of G is any subset C of $\{c_1, \cdots, c_n\}$ that can be obtained from any segment $S = [c_{i_k}, c_{j_l}]$ of G and any homolog assignment, where c_{i_k} (respec. c_{i_l}) is a member of the gene family c_i (respec. c_j). An *individual conserved block* is defined by its endpoints c_i, c_j and the gene subset U contained between these endpoints. Given two genomes G and H with possible gene copies, a *common* (respec. *conserved*) *block* is an individual common (respec. conserved) block of both G and H.

For example, $\{a, b, c, f\}$ is an individual common block of the genome G in Fig. 3 obtained by choosing the copy f_1 from the gene family f. It is also an individual common block of H obtained by choosing the copy d_2 in the gene family d. Therefore, it is a common block of G and H. On the other hand, G contains two individual conserved blocks ending with a and c, depending on whether f_1 is the copy chosen from the gene family f, or not (Fig. 3.(1)). In the former case the block is B_1, ending with a, c and defined by $U = \{b, f\}$; in the

latter case, the block is B_2 ending with a, c and defined by $U = \{b\}$. B_1 is a conserved block of G and H, as it is also an individual conserved block of H (by choosing c_2, d_2 and any of the two copies b_1 or b_2) (Fig. 3.(2)). B_2 is also a conserved block of G and H, as it is an individual block of H (by choosing b_1 and c_1). However blocks B_1 and B_2 are incompatible in H as they require two different homolog assignments for the gene family c (Fig. 3.(3)).

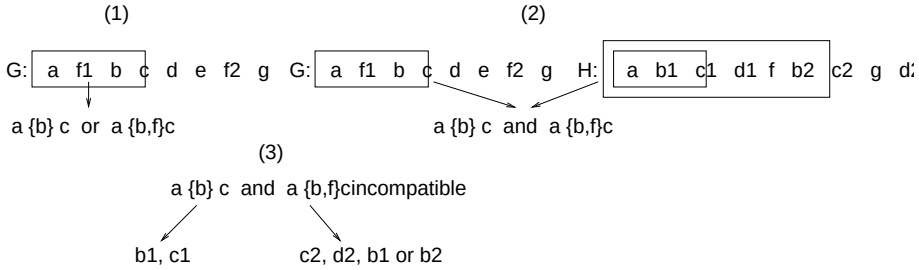


Fig. 3. Maximizing the nb. of conserved blocks for the genomes G and H with seven gene families represented by a, b, c, d, e, f, g. (1) Finding individual conserved blocks in G; (2) Finding conserved blocks; (3) Maximizing the compatible conserved blocks.

Our method will consist of three steps: 1) find all individual blocks of G and H respectively, 2) find the common or conserved blocks by superimposing the individual blocks of G and H and 3) select a maximal number of compatible conserved or common blocks. The method used at steps 2 and 3 is identical for common and conserved blocks. However, step 1 is slightly different for the two criteria. We will present the method for conserved blocks, and indicate the differences for common blocks. More details on the algorithms and proofs of theorems will appear in the full version of this abstract.

3.2 Finding individual conserved blocks

For each genome and each pair $\{a, b\}$ representing two gene families, we compute all individual conserved blocks $[a, U, b]$ by traversing the genome once, and constructing a tree-like structure $\mathcal{T}_{a,b}$ (Fig. 4a,b). The initial node is denoted by Φ. At the end of the construction, a *terminal node* t represents an individual block $[a, U, b]$ defined by the set U of labels in the path from Φ to t. As a block is not affected by the order of its elements between its two endpoints, for efficiency purposes we maintain a lexicographical order for each path in the tree.

During the tree construction, in addition to be terminal or not, each node is either marked or unmarked. The marked nodes correspond to partial individual blocks that can potentially form individual blocks later if a gene b is uncountered.

At the begining, $\mathcal{T}_{a,b}$ is restricted to a marked initial node Φ. The segment surrounded by the first copy of a and the last copy of b is then traversed from left to right. For each gene S_i in this segment, if $S_i = a$, we mark the initial state; if $S_i = b$, all marked states become terminal; otherwise, the tree $\mathcal{T}_{a,b}$ is incremented

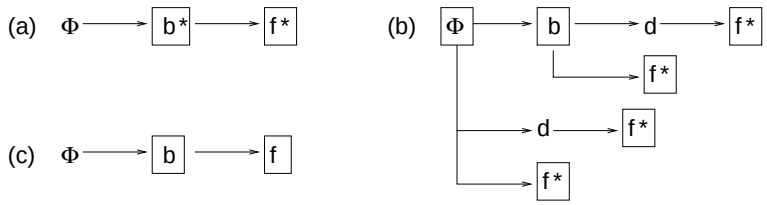


Fig. 4. The trees obtained for the pair $\{a, c\}$ for (a): genome G, and (b): genome H, first introduced in Fig. 3. Marked states are denoted by a '*', and terminal states are boxed. Superimposing trees (a) and (b) gives the tree (c), which represents the common blocks $[a, c]$ of G and H: the one containing b, and the one containing $\{b, f\}$.

by *Algorithm Add-Node* (Fig. 5). For simplicity, we do not distinguish between a node and its label. Moreover, the *lexicographical order* of *node* refers to the lexicographical order of the sequence of labels in the path from Φ to *node*.

Finally, a terminal path denotes a path from Φ to a terminal node. A non-terminal path denotes any path from Φ or a terminal node to a leaf, that do not contain any terminal node.

Theorem 4. *$[a, U, b]$ is an individual conserved block of G if and only if U is the set of node's labels of a unique terminal path of $\mathcal{T}_{a,b}$.*

Complexity For each of the n^2 gene pairs $\{a, b\}$, where n is the number of genes, each genome G and H is traversed once. For each pair $\{a, b\}$ and each position i (from 1 to the size m of the genome), the ith character G_i of G has to be added to the current tree $\mathcal{T}_{a,b}$. This requires to traverse the tree once, and potentially perform subtree copies. Therefore, the worst time complexity of the algorithm is in $O(2n^2mS)$, where S is the size of the largest tree. In practice, subtree copies can be time consuming for large trees, making the algorithm unapplicable for large data. But, an easy way to circumvent this problem is to fix a tree depth threshold limiting the search to blocks of bounded length. We will show in Section 4.1 that using any reasonable tree depth threshold provides similar levels of accuracy.

Common blocks: In the case of common blocks, there are three main differences: 1) we construct a unique tree for each genome (unstead of constructing a tree for each pair $\{a, b\}$ and each genome), 2) the initial state Φ is always marked and 3) all tree-states are terminal.

3.3 Finding all conserved blocks

The conserved blocks $[a, U, b]$ of G and H are obtained by superimposing the two trees $\mathcal{T}^G_{a,b}$ and $T^H_{a,b}$ corresponding to G and H respectively (Fig. 4c).

Theorem 5. *$[a, U, b]$ is a common block of G and H if and only if U is the set of node's labels of a terminal path common to $\mathcal{T}^G_{a,b}$ and $T^H_{a,b}$.*

```
Algorithm Add-Node (S_i, T_{a,b})
 1.  For each node in lexicographical order Do
 2.      If  node = S_i
 3.          Mark node;
 4.      Else If  node < S_i and node is marked
 5.          If node has a child labeled S_i
 6.              Mark this child;
 7.          Else
 8.              Create a node new labeled S_i, and an edge from node to new;
 9.              Mark new;
10.          End If
11.      Else If  node > S_i
12.          nodePrec = node's father;   P = subtree rooted by nodePrec;
13.          If nodePrec does not have a child labeled S_i
14.              Create node new labeled S_i, and an edge from nodePrec to new;
15.          End If
16.          Attach P to the child of nodePrec labeled S_i;
17.      End If
18.      If node ≥ S_i
19.          Skip all nodes of the subtree rooted by node
20.      End If
21.  End For
22.  If S_i represents a single gene
23.      Remove all non-terminal paths that do not contain S_i;
24.      Unmark the nodes in all terminal paths that do not contain S_i;
25.      Unmark all the ancestors of the S_i nodes;
26.  End If
```

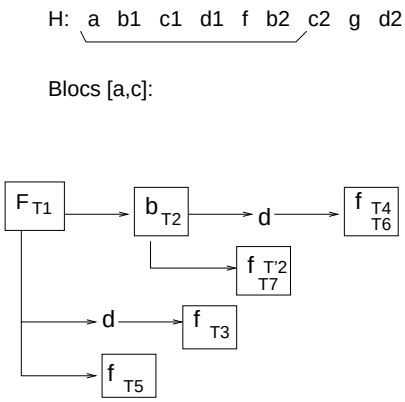
Fig. 5. Updating the tree $\mathcal{T}_{a,b}$ after reading the next gene S_i in the largest segment of genome G surrounded by the gene families a and b.

Notice that not all gene families are contained in conserved blocks. Consequently some gene families that have not retain sufficient positional context in both genomes may not be "resolved" with our approach. For example, the tree of Fig. 4c do not contain nodes for gene families d and c.

3.4 Maximizing compatible blocks

As illustrated in Fig. 3c, different blocks are obtained by different constraints that may be contradictory. In order to find compatible blocks, the constraints attached to each block have to be computed during the construction of individual trees. This is done with no additional complexity cost, by just labeling node marks, and keeping in a table all constraints attached to each mark. As soon as an endpoint is encountered, all marks become terminal and are reported with their constraints (Fig. 6).

Finally, after superimposing the two genomes's trees and amalgamating the



	b1	b2	c1	c2	d1	d2
T1	0	1	1	0		
T2	1	0	1	0		
T'2			0	1	0	1
T3			0	1	1	0
T4			0	1	1	0
T5	0	0	0	1	0	1
T6	0	1	0	1	1	0
T7	0	1	0	1	0	1

Fig. 6. Terminal states of the tree in Fig. 4b. The table represents states constraints: 1s are variables that have to be chosen, and 0s those that have to be avoided. Empty squares mean no constraints for the corresponding variables. State T_5 is irrelevent and has to be removed in a subsequent step, as b_1 and b_2 can not be avoided simultaneously.

corresponding constraints at each terminal state, a set of clauses representing all conserved blocks is obtained. Maximizing the number of compatible blocks is then reduced to a problem related to the extensively studied maximum satisfiability (MAX-SAT). It is stated as follows: given a boolean formula in cunjunctive normal form (CNF), find a truth assignment satisfying a maximum number of its clauses. Even though the MAX-SAT problem is NP-complete, it is well characterized, and many efficient heuristics have been developed. However, the clauses representing our blocks are not in CNF. Therefore, no direct MAX-SAT solver can be used in this case. We developed an appropriate heuristic based on the general method classically used to solve MAX-SAT problems: 1) Set an initial solution (variable assignment) and evaluate the clauses; 2) Explore a neighborhood of the initial solution, reevaluate the clauses and keep the best solution; 3) Stop at convergence or after a fixed number of iterations.

4 Experimental results

We used simulated data to assess the performance of the synteny blocks criteria to assign ancestral homologs. The data is generated as follows. Starting from a genome G with 100 distinct symbols representing 100 gene families, we obtain a second genome H by performing k rearrangements on G, and then randomly adding p_G gene copies in G and p_H gene copies in H at random positions. These copies may represent artifacts of an alignment tool. We simulated 5 different instances for each triplet (k, p_G, p_H), for $k \in \{10, 20, 30, \cdots, 90\}$, $p_G \in \{0, 10\}$ and $p_H \in \{10, 20\}$. We considered two rearrangement models: 1) inversions, transposition and inverted transpositions of size l following a Poisson distribution $P_\lambda(l)$ with $\lambda = 0.8$, to favour rearrangements of short segments (`ALL`) and 2) inversions of random size only (`INV`). We then run the algorithms and considered

the number of correct homolog assignments (resolved) and false predictions.

4.1 Impact of tree depth threshold

As explained in Section 3.2, in order to obtain an efficient time algorithm, we use a heuristic that constructs individual trees not exceeding a given tree depth threshold. Fig. 7 shows the results obtained for tree depth thresholds 5 and 10, using the evolutionary model `ALL`. For both common and conserved blocks, there is almost no tree depth effect on the quality of the results. In general, depth 10 provides slightly more resolved genes, but also slightly more false predictions. This result validates the fact that restricting the search to blocks of limited size is sufficient to capture the genomic context information.

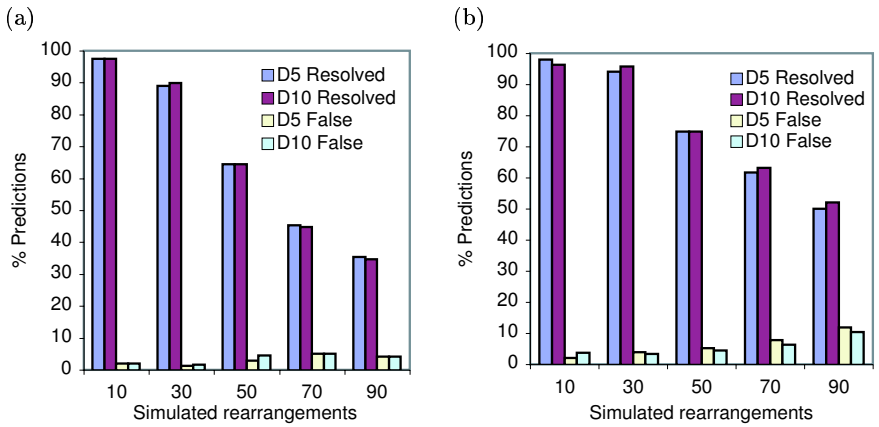


Fig. 7. Tree depth effect, for depth 5 (D5) and depth 10 (D10), on homolog assignment using: (a) the conserved block and (b) the common block criterion.

4.2 Comparing synteny blocks and breakpoint distance criteria

We have compared the blocks criteria with the breakpoint distance criteria using the exemplar method developed in [21]. Fig. 8a shows the results obtained for the evolutionary model `ALL`. In general, the conserved and common blocks criteria allow to correctly resolve less genes than the exemplar method. However, the number of false predictions is notably reduced with our approaches. Comparing the two blocks criteria, common blocks correctly resolve more genes, while conserved blocks give less false predictions. We further compared the common and conserved blocks criteria using the evolutionary model `INV` (Fig. 8b). It appears that both criteria have almost the same proportion of true predictions, while the common blocks criterion produces more false predictions. In the case of reversals of large segments, using the conserved blocks criterion seems to be more appropriate, as it limits the noise introduced by irrelevant blocks.

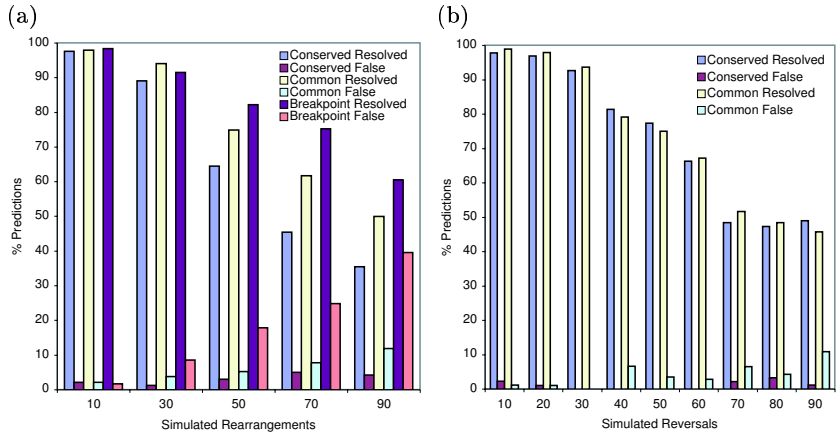


Fig. 8. (a) Comparison of the blocks and breakpoint distance criteria, using evolutionary model `ALL`; (b)Comparison of the blocks criteria using evolutionary model `INV`.

4.3 Impact of homolog assignment on the reversal distance

Various approaches have been considered in the past to preprocess duplicated genes for genome rearrangement studies. A common approach has been to remove all duplicated genes even though the missing data will typically lead to an underestimate of the rearrangement distances. An alternative approach could be to randomly assign corresponding pairs but that, in contrast, would lead to an overestimate of the actual distances. We were interested in measuring the extent of this under/over estimation and to compare it with the bias of our own methods for homolog assignment. The results are shown in Fig. 9. We observe a much stronger impact, especially at moderate levels of rearrangements, of the random assignment of homologs compared to the simple removal of all duplicated genes.

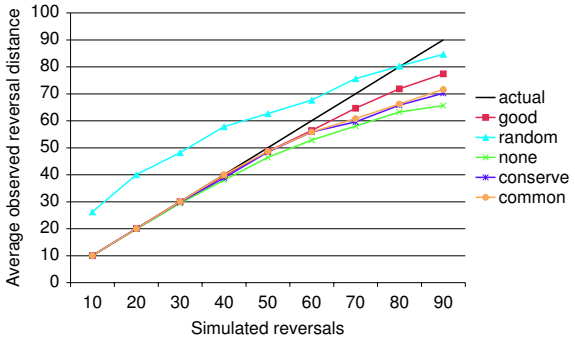


Fig. 9. The impact of homolog assignment on the observed reversal distance. 'Actual': number of simulated reversals. 'Good': ideal case with true ancestral assignment, 'Random': random selection of orthologs. 'Conserve' and 'common' are as before.

5 Conclusion

We have shown how synteny blocks can be used to accurately recover a large proportion of ancestral homologs. Based on the observation that incorrect assignment of homologs tend to have a more damageable impact on the induced rearrangement distances, we propose that a conservative approach, with a low level of false positives, is probably most desirable for this problem. Another strength of the approach is that it is directly generalizable to sets of multiple genomes. The next step of our work will be use the method, in replacement to the one used [13], to assign positional homology in bacterial genomes, and subsequently for the annotation of more complex genomes.

Acknowledgments: We are grateful to Louxin Zhang for his participation in the experimental part of the project. We are also grateful to Elisabeth Tillier for fruitful discussions.

References

1. D. Sankoff. Genome rear. with gene fam. *Bioinformatics*, 15:909–917, 1999.
2. J. Kececioglu and D. Sankoff. Exact and approx. algo. for sorting by reversals, with application to genome rear. *Algorithmica*, 13:180–210, 1995.
3. S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). *J. ACM*, 48:1–27, 1999.
4. H. Kaplan, R. Shamir, and R. E. Tarjan. A faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal on Computing*, 29:880–892, 2000.
5. N. El-Mabrouk. Genome rearrangement by reversals and insertions/deletions of contiguous segments. In *CPM 2000*, volume 1848 of *LNCS*, pages 222- 234, 2000.
6. A. Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. In *CPM*, LNCS. Springer Verlag, 2001.
7. Vineet Bafna and Pavel A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, 1998.
8. M. E. Walter, Z. Dias, and J. Meidanis. Reversal and transposition distance of linear chromosomes. In *String Proc. Information Retrieval (SPIRE '98)*, 1998.
9. T. Hartman. A simpler 1.5-approximation algorithm for sorting by transpositions. In *LNCS 2676*, volume CPM'03, pages 156-169, 2003.
10. S. Hannenhalli and P.A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proc. IEEE 36th Ann. Symp. Found. Comp. Sci.*, pages 581–592, 1995.
11. G. Tesler. Efficient algorithms for multichromosomal genome rearrangements. *J. Comp. System Sci.*, 65(3):587-609, 2002.
12. M. Ozery-Flato and R. Shamir. Two notes on genome rearrangnements. *J. of Bioinf. and Comput. Biol.*, 1(1):71-94, 2003.
13. I.J. Burgetz, S. Shariff, A. Pang, and E. Tillier. Positional homology in bacterial genomes. manuscript, 2005.
14. A. Bergeron and J. Stoye. On the similarity of sets of permutations and its app. to genome comparison. In *COCOON'03*, volume 2697 of *LNCS*, pages 68- 79, 2003.
15. G. Blin and R. Rizzi. Conserved interval distance computation between non-trivial genomes. COCOON, 2005.

16. M. Figeac and J.S. Varré. Sorting by reversals with common intervals. In *LNBI*, volume 3240 of *WABI 2004*, pages 26 - 37. Springer-Verlag, 2004.
17. S. Bérard, A. Bergeron, and C. Chauve. Conservation of combinatorial structures in evolution scenarios. In *LNCS*, volume 3388 of *RECOMB 2004 sat-meeting comp. gen.*, pages 1 - 14. Springer, 2004.
18. J. Tang and B.M.E. Moret. Phylogenetic reconstruction from gene rearrangement data with unequal gene contents. In *8th Workshop Algo. Data Struct. (WADS'03)*, volume 2748 of *LNCS*, pages 37-46. Springer Verlag, 2003.
19. X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhing, S. Lonardi, and T. Jiang. Assignment of orthologous genes via genome rearrangement. *TCCB*, 2005. accepted.
20. D. Bryant. The complexity of calculating exemplar distances. In *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map alignment and the Evolution of Gene Families*, volume 1 of *Series in Computational Biology*. Kluwer Academic Press, 2000.
21. C.T. Nguyen, Y.C. Tay, and L. Zhang. Divide-and-conquer approach for the exemplar breakpoint distance. *Bioinformatics*, 2005.
22. G. Blin, C. Chauve, and G. Fertin. The breakpoint distance for signed sequences. In *Texts in Algorithm*, volume 3, pages 3- 16. KCL publications, 2004.
23. J.F. Lefebvre, N. El-Mabrouk, E. Tillier, and D. Sankoff. Detection and validation of single gene inversions. *Bioinformatics*, 19:190i-196i, 2003.
24. P. Pevzner and G. Tesler. Human and mouse genomic sequences reveal extensive breakpoint reuse in mamm. evol. *PNAS, U.S.A.*, 100(13):7672-7677, 2003.

This article was processed using the LaTeX macro package with LLNCS style