A Practical Algorithm for Ancestral Rearrangement Reconstruction

Jakub Kováč[1], Broňa Brejová[1], and Tomáš Vinař[2]

[1] Department of Computer Science, Faculty of Mathematics, Physics, and Informatics, Comenius University, Mlynská Dolina, 842 48 Bratislava, Slovakia, e-mail: kuko@ksp.sk, brejova@dcs.fmph.uniba.sk
[2] Department of Applied Informatics, Faculty of Mathematics, Physics, and Informatics, Comenius University, Mlynská Dolina, 842 48 Bratislava, Slovakia, e-mail: vinar@fmph.uniba.sk

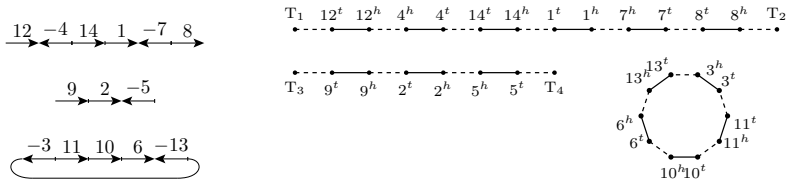
Abstract. Genome rearrangements are a valuable source of information about early evolution, as well as an important factor in speciation processes. Reconstruction of ancestral gene orders on a phylogeny is thus one of the crucial tools contributing to understanding of evolution of genome organization. For most models of evolution, this problem is NP-hard.
We have developed a universal method for reconstruction of ancestral gene orders by parsimony (PIVO) using an iterative local optimization procedure. Our method can be applied to different rearrangement models. Combined with a sufficently rich model, such as the double cut and join (DCJ), it can support a mixture of different chromosomal architectures in the same tree. We show that PIVO can outperform previously used steinerization framework and achieves better results on real data than previously published methods.
Datasets, reconstructed histories, and the software can be downloaded at http://compbio.fmph.uniba.sk/pivo/.

1 Introduction

Genome rearrangements, such as inversions, transpositions, chromosome fusions and fissions, are evolutionary events that change the order of genes in genomes. These events are rare compared to point mutations, and are thought to be precursors of speciation [1]. Due to their low evolutionary rates, rearrangements are a valuable source of information about early evolution, and various rearrangement-based distances can provide phylogenetic signal well beyond the saturation of traditional nucleotide and protein point mutation models [2].

On the other hand, rearrangements make comparative analysis much more complex. Even though it is possible to reconstruct ancestral sequences in regions without breakpoints [3], it is much more difficult to order individual segments of ancestral genomes correctly [4].

In this paper, we present a new method for analysis of rearrangement histories, using gene orders of present-day species on a given phylogeny. Our goal is to reconstruct gene orders in all ancestral nodes of the phylogenetic tree. In



(a) Genome π. Arrows represent markers with known orientation.

(b) Graph $G(\pi)$. Each marker consists of two extremities – head and tail; T denotes a telomere. Solid edges connect extremities of a single marker, dashed edges are adjacencies.

Fig. 1. Genome π consisting of two linear chromosomes and one circular chromosome (left) and its graph representation (right).

particular, we are looking for the most parsimonious ancestral gene orders, minimizing the overall rearrangement distance (this problem is also called the *small phylogeny problem*). Even though our algorithm is not guaranteed to find the optimal solution, it presents a framework that generalizes most search strategies applied to various versions of this problem to date, e.g. [5, 6].

Our method, `PIVO` (Phylogeny by IteratiVe Optimization), is one of the first practical tools applicable to analysis of real datasets spanning a complex phylogeny and accommodating a variety of genome architectures (single vs. multiple chromosomes, linear vs. circular chromosomes). In our experiments, we use the *double cut and join* (DCJ) model [7, 8] for measuring rearrangement distance, but our method can be easily applied to other rearrangement distance measures reviewed in the next section. We demonstrate the accuracy of our program on the well-studied dataset of *Campanulaceae* chloroplast genomes [9], and apply it to the reconstruction of rearrangement histories of newly sequenced mitochondrial genomes of pathogenic yeasts from *Hemiascomycetes* clade [10].

Preliminaries, definitions, and related work. We will represent a genome as a set of *markers* (e.g., genes or synteny blocks) with known order and orientation. In this setting, a genome can be represented as a graph in which each (oriented) marker corresponds to two vertices, called *extremities* of the marker; the ends of linear chromosomes are represented by special vertices called *telomeres*. The edge set of this graph consists of the *marker edges*, joining the two extremities of each marker, and the *adjacencies*, joining two consecutive extremities in the genome or an extremity with a telomere (Fig. 1). Each connected component of this graph is thus a cycle or a path (representing a circular or a linear chromosome, respectively).

In our program, we employ the double cut and join (DCJ) [7, 8] and reversal rearrangement model [11, 12]. The DCJ model encompasses a rich set of rearrangement operations, and is able to represent both linear and circular chromosomes. An evolutionary operation in the DCJ model takes two adjacencies, $\{p, q\}$ and $\{r, s\}$, and replaces them by either $\{p, r\}$ and $\{q, s\}$, or $\{p, s\}$ and $\{q, r\}$. This operation is quite general. A single DCJ operation can represent a reversal, translocation, chromosome fusion or fission, and excision or integration

of a circular chromosome. Two operations can simulate a transposition. The DCJ distance is defined as the minimum number of DCJ operations needed to transform one genome into another. The distance between two genomes can be computed in linear time [8].

The *reversal model* is another popular model of evolution by rearrangements [13]. Each genome is represented by a *signed permutation* (a sequence of chromosomal markers with their orientations), allowing only genomes with a single linear chromosome. Genomes can be modified by a reversal operation that takes a contiguous section of markers in the permutation and replaces it with the markers in reversed order and with reversed orientations. The distance between two genomes with the same marker content is then the minimum number of reversals needed to transform one genome into the other. Reversal distance can be computed in linear time [14], and can be easily adapted for circular genomes. The *Hannenhalli-Pevzner (HP) model* [15] generalizes the reversal model to genomes with multiple linear chromosomes, allowing in addition to reversals also translocations, fusions, and fissions. The HP-distance can also be computed in linear time [16, 17] and can be seen as a special case of the DCJ model restricted to operations that do not create circular chromosomes.

For completeness, we also consider a simple *breakpoint distance* [18]. A *breakpoint* between two genomes is a pair of markers that are consecutive in one genome but not in the other. The number of breakpoints between the two genomes is called the breakpoint distance. While this measure does not correspond to a particular set of evolutionary operations, it is clear that various rearrangement operations generally increase the breakpoint distance between the genomes, unless they reuse already created breakpoints.

While we can compute distances in all of these models efficiently, finding the most parsimonious solutions in more complex scenarios involving multiple genomes is more difficult. Perhaps the simplest scenario is the *median problem*, where we are given three extant genomes π_1, π_2, π_3 and a rearrangement distance measure d, and our task is to compute a single ancestral genome, a median π_M, that would minimize the sum of distances to the extant genomes $d(\pi_1, \pi_M) + d(\pi_2, \pi_M) + d(\pi_3, \pi_M)$.

Median problem has been shown to be NP-hard for almost every considered rearrangement model (unichromosomal reversal distance [19], unichromosomal breakpoint distance [20, 21], multichromosomal linear breakpoint distance [22], unichromosomal [19] and multichromosomal [22] DCJ distance) and is conjectured to be NP-hard for other rearrangement models. One notable exception is the breakpoint distance on multiple circular or mixed chromosomes for which the median can be computed in polynomial time [22].

On the other hand, practical algorithms were developed for particular models, allowing to find good solutions to the median problem in many instances. Blanchette, Bourque, and Sankoff [23, 18] reduce the breakpoint median problem to the travelling salesman problem (TSP) and then use a branch-and-bound algorithm to solve the resulting instance of TSP exactly. Siepel and Moret [24] and Caprara [25] propose exact practical solutions for the reversal median.

Heuristic median solvers try to move the given genomes closer and closer to each other until they meet at an approximate median [6, 26–28]. A similar heuristic was also implemented for the DCJ model by Adam and Sankoff [29], while Xu and Sankoff have recently developed a DCJ median solver that is exact yet fast in practical instances [30].

The median problem is a special case of the *small phylogeny problem*, where we are given multiple extant genomes and their phylogenetic tree, and our goal is to compute genomes of their ancestors. According to the parsimony principle, the best reconstruction is the one involving the smallest number of rearrangement operations.

More formally, let G be the set of all possible genomes on a particular set of markers according to the chosen rearrangement model and let d be a distance measure on G. We are given a phylogenetic tree $T = (V, E)$ with the set of leaves L. For each leaf, we are also given a genome of the corresponding species, i.e., we are given a function $g : L \to G$. An *evolutionary history* is a function $h : V \to G$ extending g to the internal (ancestral) vertices. Our goal is to find an evolutionary history h which minimizes the overall evolutionary distance in the tree $d(h) = \sum_{\{u,v\} \in E} d(h(u), h(v))$.

The small phylogeny problem is trivially NP-hard for most rearrangement distance measures, since it is a generalization of the median problem. A notable exception is the breakpoint distance, for which the complexity of the small phylogeny problem is unknown.

Perhaps the most popular method for solving the small phylogeny problem is the *steinerization* method [31]. The main idea is to iteratively improve the evolutionary history until a local optimum is reached. In each iteration, we go through all internal vertices v. We take an ancestral genome π_v and its three neighbours π_a, π_b, π_c, compute the median π_M of π_a, π_b, π_c, and replace π_v with π_M if it yields a better overall score. If no vertex can be improved by taking the median of its neighbours, we have found a locally optimal evolutionary history. This approach is implemented in `BPAnalysis` software [23, 18] for the breakpoint model and in `GRAPPA` software [32–34] for both breakpoint and reversal models. The same approach was implemented for the DCJ model by Adam and Sankoff [29]. `MGR` [6], another small phylogeny solver for reversal model, uses the simple heuristic based on using operations bringing genomes closer to other genomes in the tree. Our new algorithm, presented in the next section, encompasses and extends all these existing approaches.

Note that rearrangement models can also be used to reconstruct phylogenetic trees based on the order of markers in genomes. Perhaps the easiest method is to compute distances between all pairs of extant genomes and then use traditional distance-based algorithms for phylogeny reconstruction [2]. Another option is to solve the full *large phylogeny problem*, where we are looking for both the phylogenetic tree and the evolutionary history h minimizing the overall evolutionary distance. The small phylogeny can be used as a subroutine in the large phylogeny problem solvers. For a small number of species, we can enumerate all possible trees and for each tree compute the small phylogeny score, as in `BPAnalysis`

Algorithm 1: Iterative local optimization

```
Data: evolutionary history h
Result: local optimum
s' ← score(h), s ← ∞ ;
while s' < s do
    cand ← generate lists of candidates (neighbourhood of h);
    h ← best(cand);
    s ← s', s' ← score(h);
end
return h
```

and `GRAPPA` [23, 34]. Alternatively, we can use the *sequential addition* heuristics, where we start from a trivial three-species star phylogeny and build the tree iteratively by adding new species, and reconstructing ancestral genomes in each step as in `MGR` or `amGRP` [6, 35]. Our new method can also be used in this context.

2 Methods

In this section, we introduce a new general approach to the small phylogeny problem based on iterative local optimization. The basic idea is that in each step, we propose multiple candidates for ancestral genomes in each internal node of the tree and choose the most parsimonious combination of the candidates by dynamic programming. We will formulate the method in general terms for any rearrangement distance measure d that can be efficiently computed.

Consider a phylogenetic tree $T = (V, E)$ with the set of leaves L and genomes of extant species $g : L \to G$. We start with some history h_0. For a particular history h and each internal vertex v, we propose a set of candidates $\text{cand}(h, v)$. We define a neighbourhood of history h as the set of all possible combinations of candidate genomes $N(h) = \{h' \mid \forall v \in V : h'(v) \in \text{cand}(h, v)\}$. We then find the best history in the neighbourhood $N(h)$ by a dynamic programming algorithm. If the new history is better than the previous one, we take it and repeat the iteration. Otherwise, we have found a local minimum and the algorithm terminates. We repeat the local optimization several times starting from different histories h_0. Algorithm 1 summarizes the local optimization method.

Example #1: For each internal vertex v, the set of candidates $\text{cand}(h, v)$ can be the set of all the genomes within the distance 1 from $h(v)$. The neighbourhood of h is then the set of all histories we can obtain from h by performing at most one operation to each ancestral genome. Note that the size of $N(h)$ is exponential in the number of internal vertices, but as we will see later, we will never have to enumerate the entire neighbourhood.

Example #2: The steinerization approach mentioned in Section 1 is a special case of our method. Here, $\text{cand}(h, v) = \{h(v)\}$ for all vertices except for one vertex w with neighbours a, b, c, for which $\text{cand}(h, w) = \{h(w), \text{median}(h(a), h(b), h(c))\}$.

Note that the opposite is not true, and in fact, proposing multiple candidates and then choosing the best combination is a crucial feature of our algorithm.

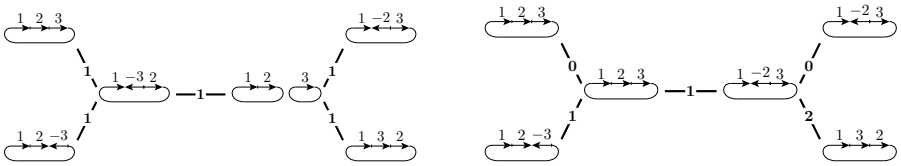


Fig. 2. A simple example showing a situation, where our more general approach outperforms the steinerization method. Given a quartet phylogeny and genomes at the leaves, the steinerization method has a local optimum with score 5 under the DCJ model (left), while there is a better solution with score 4 (right) which may be obtained when considering multiple candidates and choosing their best combination.

Consider a simple example on a quartet phylogeny in Figure 2. The steinerization approach may get stuck in a local optimum as in Figure 2(a) (both ancestral genomes are medians of the neighbouring vertices). To avoid such local optima, the steinerization method is repeated from different starting configurations. On the other hand, if we consider *all* solutions of the median problems as candidates or if we consider the neighbouring genomes (that are within one DCJ operation from the current ancestors) and then choose the best combination, we obtain a better solution, shown in Figure 2(b).

We can generalize this example to configurations that will result in arbitrarily bad local optima of the steinerization method, whereas the global optimum can be found by our method.

2.1 Finding the Best History in a Neighbourhood

Even though the size of the neighbourhood $N(h)$ can be exponential (it has $\prod_v |\operatorname{cand}(h, v)|$ elements), the best history can be found in polynomial time using dynamic programming.

Let c_i^u be the i-th candidate from $\operatorname{cand}(h, u)$ and let $M[u, i]$ be the lowest score we can achieve for the subtree rooted at u if we choose candidate c_i^u as an ancestor. $M[u, i] = 0$ if u is a leaf. If u is an internal vertex with children v and w, we first compute values $M[v, j]$, $M[w, k]$ for all j, k. Then

$$M[u, i] = \min_j\{M[v, j] + d(c_i^u, c_j^v)\} + \min_k\{M[w, k] + d(c_i^u, c_k^w)\}.$$

This algorithm can be easily generalized for non-binary phylogenetic trees.

If n is the number of species, m the number of markers in each genome, and k the number of candidates for each ancestor, the best history can be found in time $O(nmk^2)$ (provided that the distance between two genomes can be computed in $O(m)$ time).

2.2 Strategies for Proposing Candidates

A crucial part of our method is proposing good candidates. By proposing more candidates, we explore a larger neighbourhood, but finding the best combination

of candidates is slower. Furthermore, if we propose only candidates that are close to the genomes in the current history, the convergence to the local optimum may be slow. Here, we list several strategies for proposing candidates.

Extant species. In the initialization step, we can take genomes of the extant species as candidates in each internal node to get an evolutionary history to start with.

Intermediates. For a vertex v with adjacent vertices u and w, we can take intermediate genomes as candidates, i.e. if π, γ are genomes at u and w, we can sample genomes θ such that $d(\pi, \theta) + d(\theta, \gamma) = d(\pi, \gamma)$.

Medians. The steinerization method uses a median of the genomes in the three adjacent vertices as a candidate. Note that often there are many medians with the same score. Furthermore, Eriksen [36] shows that medians of moderately distant genomes may be spread wide apart. In our method, we do not need to decide beforehand which median to use. Instead, we consider all the medians as candidates, as already advocated by Eriksen [37] and Bernt *et al.* in `amGRP` [35] (`amGRP`, however, backtracks over different choices).

If we compute the median by branch-and-bound, the time to list all medians is comparable to the time to find just one median (median solvers of Siepel [24], and Caprara [25] are capable of listing all medians). If we try to find the median heuristically by repeatedly moving the given genomes closer to each other, we can take the intermediate genomes as candidates. Another option is to find just a single median and search its neighbourhood for medians which can be added to the candidate list.

Neighbours. We can include neighbourhoods of individual genomes. In particular, if $h(v) = \pi$, we can add the set $N(\pi) = \{\gamma \in G \mid d(\pi, \gamma) \leq 1\}$ to $\mathrm{cand}(h, v)$. For most models, the size of $N(\pi)$ is roughly quadratic in the number of markers. Since for large genomes this becomes infeasible, we can include only genomes that do not increase the total distance to adjacent vertices, genomes closer to some genome in an adjacent vertex, or focus on a particular subset of neighbours.

Best histories. We can take several locally optimal histories and use the reconstructed ancestors as candidates. In this way we can "recombine" locally optimal solutions discovered previously.

2.3 Unequal Gene Content

A useful extension of our method is to allow a set of possible genomes in each leaf to be given on input instead of one fixed genome $g(v)$. This feature is useful if we are uncertain about the order of markers in some genomes. The algorithm will choose one of the alternative gene orders, so as to minimize the overall parsimony cost. Note that this choice can change between the iterations, and consequently we do not commit to a particular interpretation of the dataset until the end of the local optimization.

In addition to modeling uncertainty about the gene order in the extant species, we can also use this method for handling recent duplications or losses. Genome rearrangement models usually require equal gene content in all considered genomes. However, if one of the genomes contains a duplicated segment of

Table 1. The number of operations used to explain *Campanulaceae* dataset under different models and with different algorithms.

	reversal distance	unichr. DCJ	general DCJ
`GRAPPA` (Moret *et al.* [33])	67		
`MGR` (Bourque and Pevzner [6])	65		
`GRAPPA` (Moret *et al.* [38])	64		
`BADGER` (Larget *et al.* [39])	64		
`ABC` (Adam and Sankoff [29])		64	**59**
`PIVO` (this paper)	**62**	**62**	**59**

markers, we can try to delete one or the other copy, producing two alternative gene orders that are used as candidates for the corresponding leaf of the tree. The algorithm will choose one of them for the locally optimal history h, presumably the one corresponding to the ancestral state before the duplication happened. We can extend this idea and use a larger set of candidates in case of multiple duplications or a gene loss. However, list of candidates will become prohibitively large for genomes with many such events.

3 Results

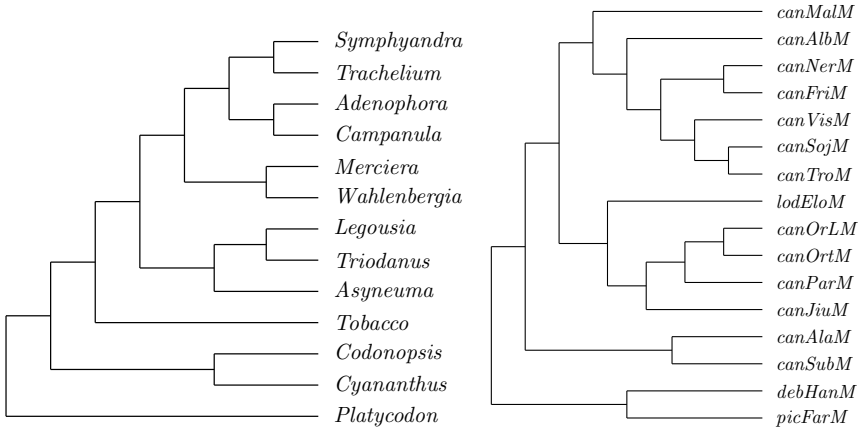
We have implemented our new method and the strategies for exploring neighbourhoods described in the previous section using the DCJ and reversal rearrangement models. We demonstrate utility of our method on two datasets.

The *Campanulaceae* cpDNA dataset. For comparison, we applied our program to a well-studied dataset of 13 *Campanulaceae* chloroplast genomes [9]. Each genome in this dataset consists of a circular chromosome with 105 markers. Using the phylogenetic tree in Fig. 3(a) reconstructed by Bourque and Pevzner [6] with `MGR`, the results are presented in Table 1.

Using `GRAPPA` software, Moret *et al.* [33] found 216 tree topologies and evolutionary histories with 67 reversals. Bourque and Pevzner [6] using `MGR` later found a solution with 65 reversals. Even better solutions with 64 reversals were found by Moret *et al.* [38] (using `GRAPPA`) and Larget *et al.* [39] (using `BADGER`).

Adam and Sankoff [29] used the more general DCJ model and the phylogenetic tree by Bourque and Pevzner. They found a history with 64 DCJ operations with ancestors having a single chromosome, and a history with 59 DCJ operations with unconstrained ancestors. However, as Adam and Sankoff note: "There is no biological evidence in the *Campanulaceae*, or other higher plants, of chloroplast genomes consisting of two or more circles." The additional circular chromosomes are an artifact of the DCJ model, where a transposition or a block interchange operation can be simulated by circular excision and reincorporation.

We penalized multiple chromosomes in the dynamic programming objective function to avoid such artifacts and found several histories with 62 DCJ operations, where all the ancestors had a single circular chromosome. Moreover, these



(a) Phylogenetic tree of *Campanulaceae* (b) Phylogenetic tree of *Candida*

Fig. 3. Phylogenetic trees used in the experiments.

histories only require 62 reversals, which further improves on the best previously known result of 64 reversals by Moret *et al.* [38] and Larget *et al.* [39].

The *Hemiascomycetes* mtDNA dataset. We have also studied evolution of gene order in 16 mitochondrial genomes of pathogenic yeasts from the CTG clade of *Hemiascomycetes* [10]. The phylogenetic tree (Fig. 3(b)) was calculated by `MrBayes` [40] from protein sequences of 14 genes and is supported by high posterior probabilities on most branches.

The genomes consist of 25 markers (synteny blocks): 14 protein-coding genes, two rRNA genes, and 24 tRNAs. Several challenges make this dataset difficult. First, it combines genomes with a variety of genome architectures: *C. subhashii, C. parapsilosis*, and *C. orthopsilosis* are linear, *C. frijolesensis* has two linear chromosomes, and the rest of the species have circular-mapping chromosomes.

Some of the genomes (*C. albicans, C. maltosa, C. sojae, C. viswanathii*) contain recent duplications which cannot be handled by the DCJ model. As outlined in Section 2.3, we have removed duplicated genes, and included both possible forms of the genomes as alternatives in the corresponding leaves. Similarly, the genomes of *C. alai, C. albicans, C. maltosa, C. neerlandica, C. sojae*, and *L. elongisporus* contain long inverted repeats that are often subject to recombination resulting in reversal of the portion of the genome between the two repeats. Both forms of the genome are routinely observed in the same species, and we include both of them in the corresponding leaf.

Finally, we penalized occurrences of multiple circular chromosomes and combinations of linear and circular chromosomes in ancestral genomes. Such combinations would likely represent artifacts of the DCJ model.

Our algorithm, using the *extant species, neighbours,* and *best histories* strategies, has found an evolutionary history with 78 DCJ operations. More detailed discussion of this dataset (including comparison to manual reconstruction in a subtree of closely related species) is included elsewhere [10].

4 Conclusion

We have developed a new method for reconstructing evolutionary history and ancestral gene orders, given the gene orders of the extant species and their phylogenetic tree. We have implemented our method using the double cut and join model and studied evolution of gene order in 16 mitochondrial yeast genomes, demonstrating applicability of our approach to real biological datasets. We have also analyzed the thoroughly studied *Campanulaceae* dataset and improved upon the previous results [33, 6, 38, 39, 29].

Our framework is compatible with a variety of rearrangement models and the optimization can be adjusted by introducing new strategies of generating candidate ancestral genomes. The use of the DCJ allowed us to study datasets that contained both linear and circular genomes and to contribute towards understanding mechanisms of genome linearization during the evolution [10].

In our experiments, we have explored only a small fraction of possible strategies offered by our framework. Systematic study of new initialization methods, candidate sets, and rearrangement measures may lead to even better results for a variety of practical problems. Our work also opened an avenue towards a systematic solution of the problems with unequal gene content. Similar directions can perhaps lead to the possibility of incorporating incompletely assembled genomes, a challenge posed by the next-generation sequencing technologies.

Acknowledgements. We would like to thank Jozef Nosek and Matúš Valach for helpful discussions on this problem and for providing *Hemiascomycetes* data. This research was supported by Comenius University Grant for Young Researchers UK/121/2011 to JK, Marie Curie grants IRG-224885 and IRG-231025 to TV and BB, and VEGA grant 1/0210/10.

References

1. Brown, K., Burk, L., Henagan, L., Noor, M.: A test of the chromosomal rearrangement model of speciation in Drosophila pseudoobscura. Evolution **58** (2004) 1856–1860
2. Wang, L., Warnow, T., Moret, B., Jansen, R., Raubeson, L.: Distance-based genome rearrangement phylogeny. Journal of Molecular Evolution **63** (2006) 473–483
3. Blanchette, M., Green, E., Miller, W., Haussler, D.: Reconstructing large regions of an ancestral mammalian genome in silico. Genome Research **14** (2004) 2412
4. Ma, J., Zhang, L., Suh, B., Raney, B., Burhans, R., Kent, W., Blanchette, M., Haussler, D., Miller, W.: Reconstructing contiguous regions of an ancestral genome. Genome Research **16** (2006) 1557
5. Moret, B., Warnow, T.: Toward new software for computational phylogenetics. Computer **35** (2002) 55–64
6. Bourque, G., Pevzner, P.: Genome-scale evolution: reconstructing gene orders in the ancestral species. Genome Research **12** (2002) 26
7. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics **21** (2005) 3340–3346

8. Bergeron, A., Mixtacki, J., Stoye, J.: A unifying view of genome rearrangements. In: Algorithms in Bioinformatics (WABI). (2006) 163–173
9. Cosner, M., Jansen, R., Moret, B., Raubeson, L., Wang, L., Warnow, T., Wyman, S.: An empirical comparison of phylogenetic methods on chloroplast gene order data in Campanulaceae. In Sankoff, D., Nadeau, J.H., eds.: Comparative Genomics, Kluwer Academic Publishers (2000) 99–121
10. Valach, M., Farkas, Z., Fricova, D., Kovac, J., Brejova, B., Vinar, T., Pfeiffer, I., Kucsera, J., Tomaska, L., Lang, B.F., Nosek, J.: Evolution of linear chromosomes and multipartite genomes in yeast mitochondria. Nucleic Acids Research (2011) Accepted.
11. Hannenhalli, S., Pevzner, P.: Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. In: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, ACM (1995) 178–189
12. Bergeron, A., Mixtacki, J., Stoye, J.: The inversion distance problem. Math. of Evolution and Phylogeny (2005) 262–290
13. Kececioglu, J., Sankoff, D.: Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. Algorithmica **13** (1995) 180–210
14. Bader, D.A., Moret, B.M.E., Yan, M.: A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. Journal of Computational Biology **8** (2001) 483–491
15. Hannenhalli, S., Pevzner, P.A.: Transforming men into mice (polynomial algorithm for genomic distance problem). In: Foundations of Computer Science (FOCS). (1995) 581–592
16. Bergeron, A., Mixtacki, J., Stoye, J.: HP distance via double cut and join distance. In: Combinatorial Pattern Matching (CPM). (2008) 56–68
17. Bergeron, A., Mixtacki, J., Stoye, J.: A new linear time algorithm to compute the genomic distance via the double cut and join distance. Theoretical Computer Science **410** (2009) 5300–5316
18. Sankoff, D., Blanchette, M.: The median problem for breakpoints in comparative genomics. In: Computing and Combinatorics (COCOON). (1997) 251–264
19. Caprara, A.: The reversal median problem. INFORMS Journal on Computing **15** (2003) 93
20. Pe'er, I., Shamir, R.: The median problems for breakpoints are NP-complete. Electronic Colloquium on Computational Complexity (ECCC) **5** (1998)
21. Bryant, D.: The complexity of the breakpoint median problem. Technical Report CRM-2579, CRM Universite de Montreal (1998)
22. Tannier, E., Zheng, C., Sankoff, D.: Multichromosomal median and halving problems under different genomic distances. BMC Bioinformatics **10** (2009) 120
23. Blanchette, M., Bourque, G., Sankoff, D.: Breakpoint phylogenies. In: Genome Informatics Workshop (GIW). (1997) 25–34
24. Siepel, A.C., Moret, B.M.E.: Finding an optimal inversion median: Experimental results. In: Workshop on Algorithms in Bioinformatics (WABI). (2001) 189–203
25. Caprara, A.: On the practical solution of the reversal median problem. In: Workshop on Algorithms in Bioinformatics (WABI). (2001) 238–251
26. Arndt, W., Tang, J.: Improving reversal median computation using commuting reversals and cycle information. Journal of Computational Biology **15** (2008) 1079–1092
27. Yue, F., Zhang, M., Tang, J.: Phylogenetic reconstruction from transpositions. BMC Genomics **9** (2008) S15

28. Rajan, V., Xu, A., Lin, Y., Swenson, K., Moret, B.: Heuristics for the inversion median problem. BMC bioinformatics **11** (2010) S30
29. Adam, Z., Sankoff, D.: The ABC of MGR with DCJ. Evolutionary Bioinformatics Online **4** (2008) 69–74
30. Xu, A.W., Sankoff, D.: Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In: Workshop on Algorithms in Bioinformatics (WABI). (2008) 25–37
31. Sankoff, D., Cedergren, R., Lapalme, G.: Frequency of insertion-deletion, transversion, and transition in the evolution of 5S ribosomal RNA. Journal of Molecular Evolution **7** (1976) 133–149
32. Moret, B., Wyman, S., Bader, D., Warnow, T., Yan, M.: A new implementation and detailed study of breakpoint analysis. In: Pacific Symposium on Biocomputing (PSB), Citeseer (2001) 583–594
33. Moret, B., Wang, L., Warnow, T., Wyman, S.: New approaches for reconstructing phylogenies from gene order data. Bioinformatics **17** (2001) S165
34. Moret, B., Tang, J., Wang, L., Warnow, T.: Steps toward accurate reconstructions of phylogenies from gene-order data. Journal of Computer and System Sciences **65** (2002) 508–525
35. Bernt, M., Merkle, D., Middendorf, M.: Using median sets for inferring phylogenetic trees. Bioinformatics **23** (2007) e129
36. Eriksen, N.: Reversal and transposition medians. Theoretical Computer Science **374** (2007) 111–126
37. Eriksen, N.: Median clouds and a fast transposition median solver. In: International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC), DMTCS Proceedings (2009) 373–384
38. Moret, B., Siepel, A., Tang, J., Liu, T.: Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. Algorithms in Bioinformatics (2002) 521–536
39. Larget, B., Kadane, J., Simon, D.: A Bayesian approach to the estimation of ancestral genome arrangements. Molecular phylogenetics and evolution **36** (2005) 214–223
40. Ronquist, F., Huelsenbeck, J.: MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics **19** (2003) 1572