Rearrangement-based phylogeny using the Single-Cut-or-Join operation

Priscila Biller, Pedro Feijão, João Meidanis

Abstract—Recently, the Single-Cut-or-Join (SCJ) operation was proposed as a basis for a new rearrangement distance between multichromosomal genomes, leading to very fast algorithms, both in theory and in practice. However, it was not clear how well this new distance fares when it comes to using it to solve relevant problems, such as the reconstruction of evolutionary history. In this paper, we advance current knowledge, by testing SCJ's ability regarding evolutionary reconstruction in two aspects: (1) how well does SCJ reconstruct evolutionary topologies?, and (2) how well does SCJ reconstruct ancestral genomes? In the process of answering these questions, we implemented SCJ-based methods, and made them available to the community. We ran experiments using as many as 200 genomes, with as many as 3000 genes. For the first question, we found out that SCJ can recover typically between 60% and more than 95% of the topology, as measured through the Robinson-Foulds distance (a.k.a. split distance) between trees. In other words, 60% to more than 95% of the original splits are also present in the reconstructed tree. For the second question, given a topology, SCJ's ability to reconstruct ancestral genomes depends on how far from the leaves the ancestral is. For nodes close to the leaves, about 85% of the gene adjacencies can be recovered. This percentage decreases as we move up the tree, but, even at the root, about 50% of the adjacencies are recovered, for as many as 64 leaves. Our findings corroborate the fact that SCJ leads to very conservative genome reconstructions, yielding very few false positive gene adjacencies in the ancestrals, at the expense of a relatively larger amount of false negatives. In addition, experiments with real data from the Campanulaceae and Protostomes groups show that SCJ reconstructs topologies of quality comparable to the accepted trees of the species involved. As far as time is concerned, the methods we implemented can find a topology for 64 genomes with 2000 genes each in about 10.7 minutes, and reconstruct the ancestral genomes in a 64-leaf tree in about 3 seconds, both on a typical desktop computer. It should be noted that our code is written in Java and we made no significant effort to optimize it.

Index Terms—Genome Rearrangement, Phylogeny

1 Introduction

Genome rearrangements are evolutionary events where large, continuous pieces of the genome shuffle around, and have been studied since shortly after the very advent of genetics [1], [2], [3]. With the increased availability of whole genome sequences, gene order data have been used to estimate the evolutionary distance between present-day genomes, and to reconstruct the gene order of ancestral genomes. The simplest form of inference of evolutionary scenarios based on gene order is the pairwise genome rearrangement problem: given two genomes, find a shortest sequence of rearrangement events that transforms one genome into the other. In some applications, one is interested only in the number of events of such a sequence — the *distance* between the two genomes.

For most rearrangement events proposed, this problem has already been solved, usually with linear or subquadratic algorithms. However, when more than two genomes are considered, inferring evolution scenarios becomes much more difficult. For instance, the *genome median problem* (GMP) — given three genomes, find a fourth genome that minimizes the sum of its pairwise distance to the three given genomes — is NP-complete in most rearrangement models [4].

If we allow more than three genomes, we have a problem called the *multiple genome rearrangement problem* (MGRP) — searching phylogenetic trees describing the most "plausible" rearrangement scenario for multiple genomes [5], [6]. Formally, given n genomes, we want to find a tree T with n extant genomes as leaf nodes and assign ancestral genomes to internal nodes of T such that the tree is optimal (in a parsimonious sense), i.e., the sum of rearrangement distances over all its edges is minimal. This problem is also called the *Big Parsimony Problem* (BPP), in contrast to the easier *Small Parsimony Problem* (SPP), when a tree is given, and we just need to find genomes for the internal nodes.

The first approach to solving the MGRP was proposed by Sankoff and Blanchette [7], and implemented in their software BPAnalysis. This method performs an extensive search over all possible tree topologies finding the tree with the minimum number of *breakpoints* (adjacencies present in one genome, but absent in the other). Since the number of topologies is exponential on the number of genomes, this method was restricted to very small instances. Later, Moret et al. developed a faster, alternative method called GRAPPA [8], based on BPAnalysis, that improved the speed in several orders of magnitude.

- *Priscila Biller, Pedro Feijão and João Meidanis are with the Institute of Computing, University of Campinas, Brazil. E-mail: priscila.biller@students.ic.unicamp.br, ra932015@ic.unicamp.br, meidanis@ic.unicamp.br*
- *Pedro Feijão and João Meidanis are also with Scylla Bioinformatics, Brazil.*

Also, breakpoint distance was replaced by reversal distance [9], with the availability of a linear algorithm for the latter [10].

Another approach to the MGRP using reversals was presented by Bourque and Pevzner in their MGR program [11]. The main difference with GRAPPA is that in MGR the GMP is not solved exactly; a faster heuristic is applied instead.

In an earlier work [12], we proposed a rearrangement operation called Single-Cut-or-Join (SCJ). Under this simple breakpoint-like event, both the GMP and SPP have polynomial time solutions. This is a major advantage of using SCJ models for the MGRP, since most proposed methods (MGR and GRAPPA, for instance) are based on solving the GMP and the SPP several times, and, as we have mentioned, these problems are NP-hard for most rearrangement distances (including breakpoint and reversal distances). With SCJ, the MGRP can be solved much faster.

In this paper, we study the small and big parsimony problems under the SCJ distance, running several experiments to assess the ability of SCJ to reconstruct evolutionary histories, both with real as well as simulated data. As it turns out, SCJ does very well, being able to achieve results comparable to the best available methods.

2 Representing Genomes

We will use a standard genome representation [12]. A *gene* is an oriented sequence of DNA that starts with a tail and ends with a head, called the *extremities* of the gene. The tail of a gene a is denoted by a_t, and its head by a_h. Given a set of genes $\mathcal{G}$, the associated extremity set is $\mathcal{E}(\mathcal{G}) = \{a_t : a \in \mathcal{G}\} \cup \{a_h : a \in \mathcal{G}\}$. An *adjacency* is an unordered pair of extremities that represents the linkage between two consecutive genes in a certain orientation on a chromosome, for instance $c_h b_t$ in Fig. 1. An extremity that is not adjacent to any other extremity is called a *telomere*, for instance, a_t in Fig. 1. A *genome* is defined by a pair $(\mathcal{G}, \Pi)$, where $\mathcal{G}$ is a gene set and Π is a set of disjoint adjacencies from $\mathcal{E}(\mathcal{G})$. Telomeres are uniquely determined by the set of adjacencies and the gene set $\mathcal{G}$. Two adjacencies are said to be *conflicting* when they have at least one extremity in common. Thus, given a gene set, a genome can be characterized as a set of mutually nonconflicting adjacencies. Sometimes, we will represent a genome just by Π, its adjacency set, with $\mathcal{G}$ being then implicitly defined as the smallest gene set needed for the adjacencies in Π, or by the context.

The *graph representation* of a genome $(\mathcal{G}, \Pi)$ is a graph $G_{(\mathcal{G},\Pi)}$ whose vertices are the extremities of $\mathcal{E}(\mathcal{G})$ and there is a grey edge connecting the extremities x and y when xy is an adjacency of Π, or a directed black edge from x to y when x and y are tail and head, respectively, of the same gene. A connected component in $G_{(\mathcal{G},\Pi)}$ is a *chromosome* of $(\mathcal{G}, \Pi)$, and it is *linear* if it is a path, and *circular* if it is a cycle. A *circular genome* is a genome whose chromosomes are all circular, and a *linear genome* is a genome whose chromosomes are all linear. For instance, given the set $\mathcal{G} = \{a, b, c, d, e, f\}$, and the adjacencies $\Pi = \{a_h b_h, b_t c_h, d_h f_h, f_t e_t\}$, the graph $G_{(\mathcal{G},\Pi)}$ is given in Fig. 1.

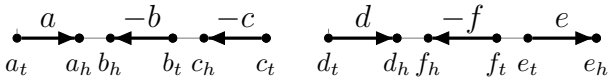

Fig. 1. Graph representing a genome with two linear chromosomes. Black, directed edges represent genes, while grey edges link consecutive extremities.

3 The SCJ operation

The SCJ operation is based on the two most basic rearrangement operations: a *cut*, an operation that breaks an adjacency in two telomeres (namely, its extremities), and a *join*, which is the reverse operation, pairing two telomeres into an adjacency. Any *cut* or *join* applied to a genome will be called a **Single-Cut-or-Join** (SCJ) operation. Since a genome is represented as a set of adjacencies, a *cut* can also be viewed as the removal of an adjacency from the set, while the *join* is the addition of a nonconflicting adjacency. The SCJ distance, denoted by d_{SCJ}, is defined as the smallest number of single-cut-or-join operations that transform one genome into the other.

The SCJ distance can be easily computed, as we see from the lemma below.

Lemma 3.1 (Feijão and Meidanis [12]): Consider two genomes represented by the sets Π and Σ, and let $\Gamma = \Pi - \Sigma$ and $\Lambda = \Sigma - \Pi$. Then, Γ and Λ can be found in linear time, and they define a minimum set of SCJ operations that transform Π into Σ, where adjacencies in Γ define cuts and adjacencies in Λ define joins. Consequently, $d_{SCJ}(\Pi, \Sigma) = |\Pi - \Sigma| + |\Sigma - \Pi|$.

The SCJ also admits a distance equation based on the Adjacency Graph, introduced by Bergeron et al. [13]. The adjacency graph $AG(\Pi, \Sigma)$ is a bipartite graph whose vertices are the adjacencies and telomeres of the genomes Π and Σ and whose edges connect two vertices that have a common extremity. Therefore, vertices representing adjacencies will have degree two, telomeres will have degree one, and this graph will be a union of paths and cycles.

Lemma 3.2 (Feijão and Meidanis [12]): Let Π and Σ be two genomes with the same set of genes $\mathcal{G}$. We have

$$d_{SCJ}(\Pi, \Sigma) = 2\big[N - (C_2 + P/2)\big], \tag{1}$$

where N is the number of genes, C_2 is the number of cycles of length two, and P the number of paths in $AG(\Pi, \Sigma)$.

Other important genome rearrangement problems, such as the generalized genome median (median of n genomes) and genome halving also have polynomial algorithms under the SCJ distance.

4 Multiple Genome Rearrangement Problem

In this section, we formally describe two MGRP problems: the Small Parsimony Problem (SPP) and the Big Parsimony Problem (BPP).

4.1 The Small Parsimony Problem

Given a tree T, where each leaf corresponds to a genome defined over the same set of genes $\mathcal{G}$, the *small parsimony problem* (SPP) consists of finding an ancestral genome Γ_v for each internal node v of T such that the total branch length of T (the sum of the weight of each edge, defined as the distance between the genomes of its incident vertices) is minimized. Formally, we want to find

$$M = \min_{G} \sum_{uv \in E(T)} d(\Gamma_u, \Gamma_v) \quad (2)$$

where $E(T)$ is the set of edges of T and G is the mapping from v to Γ_v.

A common way to solve this problem is the approach proposed by Blanchette et al. [14], where one iterates over each internal node of T, solving a *genome median problem* (GMP) until convergence to a local minimum is achieved. One difficulty with this technique is that the GMP is NP-hard for most rearrangement distances, notable exceptions being the SCJ distance [12], and the BP distance in some specific cases [4].

A significant advantage of the SCJ distance is that the SPP can be solved in polynomial time [12]. Stoye and Wittler had already used a similar strategy in their reconstruction of ancient gene clusters [15]. Basically, it applies Fitch's small parsimony algorithm on each adjacency, viewing it as a binary character, determining, in each pass, whether the given adjacency is present in each of the internal nodes of the tree, and ultimately building all ancestral genomes. This is the only known distance for which the SPP has a polynomial time solution, since for BP, even though the median problem is polynomially solvable, for the general case of four or more genomes the SPP is NP-hard [16].

4.2 The Big Parsimony Problem

The big parsimony problem under SCJ can be stated as follows. Given n genomes $\pi_1, \ldots, \pi_n$ defined on the same set of genes $\mathcal{G}$, find a tree T whose leaves are in one-to-one correspondence with the genomes $\pi_1, \ldots, \pi_n$, and find an ancestral genome Γ_v for each internal node v of T so that the total branch length of T (the sum of the weight of each edge, defined as the distance between the genomes of its vertices) is minimized.

Under the SCJ distance, the BPP is related to the Steiner tree problem in $\{0,1\}^N$, which is NP-hard [17], with the difference that the binary characters in our case are adjacencies, and therefore are not necessarily independent, since conflicting adjacencies cannot belong simultaneously to the same genome. However, with a simple coding scheme, it was proven that given an instance of the Steiner problem in $\{0,1\}^N$, it is possible to code it as an SCJ problem where the adjacencies behave as independent characters, thus effectively reducing this NP-hard problem to big parsimony under SCJ [12]. As a result, SCJ big parsimony turns out to be NP-hard as well.

5 Experimental Setup

In this paper, we are interested in assessing experimentally the quality of phylogenetic trees obtained using SCJ as the sole operation in a rearrangement model. To check how good a phylogenetic tree is, we examine several of its features, including topology, branch length (i.e., the number of evolutionary events on a branch), and ancestral genomes (genomes in internal nodes of the tree). Given a collection of leaf genomes as input, we analyze these features with metrics detailed in Section 5.3.

To compute the metrics, we applied the SCJ model in two different multiple genome rearrangement (MGR) problems. The first is the Big Parsimony Problem (BPP), where the input consists of n extant genomes and we try to find the "best" (minimum length) tree, with the n input genomes as leaves. In this case, we are interested in comparing the topologies of the tree obtained with SCJ. The other problem is the Small Parsimony Problem (SPP), where the extant genomes and the topology are given, and we need to find ancestors that minimize the length of the tree. In this case, we analyze several metrics related to the inferred ancestors.

Given that BPP is NP-Hard for SCJ, we use two methods to solve it: an exact, branch-and-bound approach and a heuristic method that adds one genome at a time (stepwise addition). For SPP, we implemented an adaptation of Fitch's algorithm, suggested by Feijão and Meidanis [12], that runs in polynomial time. This approach is very fast and gives an optimum solution. These methods are described in more detail in Section 5.2.

In addition to verifying the accuracy in inferring the features, we also analyze the performance of SCJ, measuring the computational time used in each experiment.

After defining the metrics and methods, we defined the input data to be used in each method. In these experiments we used real and simulated data, depending on the evaluated feature. We used real data to assess the topology, comparing the results with topologies inferred by other methods in literature. We chose two well-studied datasets: Campanulaceae, which is a difficult case because it is highly rearranged, and Protostomes, composed of 66 genomes, which is challenging because of its size. Real data could not be used to compare ancestors, because these are unknown in the datasets we selected.

All methods were also tested using simulated data. In the simulations, we vary the values of different parameters, such as number of input genomes, genome size, number of rearrangement events per tree edge,

frequencies of rearrangement event types, and evaluate their influence on the inferred tree. Section 5.1.1 has a more detailed description of how the simulations were conducted. Fig. 2 shows a diagram that summarizes the experiments.

5.1 Data Preparation

5.1.1 Simulated Dataset

Computer simulations are useful for verifying how well SCJ works under different evolutionary conditions, because they can exhaustively explore the impact of different parameters. In this paper, we simulate rearrangement evolution for a set of species, represented by their genomes. With these simulations, we evaluate how the following parameters influence the results:

- **Branch Length:** denotes the expected number of evolutionary events along an edge of the tree. Values are sampled from a uniform distribution on the set $\{1, 2, 3, ..., \text{MAX\_LEN}\}$, where (MAX_LEN) is defined as a percentage of the number of genes;
- **Tree size:** denotes the number of leaves in the phylogenetic tree.
- **Genome size:** consists of two parts: the number of genes and the number of chromosomes.
- **Rearrangement Distribution:** defines the frequency of each rearrangement event type, which remains the same during our simulated evolution. We consider only signed reversals, transpositions, and reciprocal translocations.

Since it is unfeasible to test all parameter combinations, we selected a default value for each parameter, and studied variations around this point. The default values are shown in Table 1. Notice that, in some cases, the default value depends on the dataset.

In the experiments, we vary just one parameter and fix the other parameters in their default values. For a given combination, we generate 200 simulated trees, as follows. First, the simulation creates the topology of a rooted binary tree with the specified number of leaves. Topology generation follows the beta-splitting model proposed by Aldous [18]. In this model, it is possible to modify the probability distribution of topologies using the β parameter, which ranges from -2 to ∞. The β parameter adjusts the probability of generating trees with a certain degree of balance, with larger β corresponding to greater balance. We use $\beta = -1$, which is the value obtained empirically by Aldous as the one that best represents the balance observed on real data.

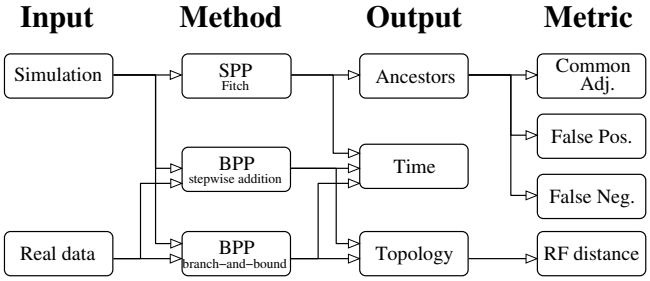

Fig. 2. Diagram with the experiments.

After defining the topology, the hypothetical ancestor at the root is created, with the specified number of genes and chromosomes, without duplications. Finally, from top to bottom, we work along each branch, evolving the parent genome with reversals, translocations, and transpositions, based on the given rearrangement distributions and branch lengths, until all leaves are reached.

It is common to use signed reversals, transpositions and reciprocal translocations in simulated evolution [19], [20], because these operations correspond to events actually observed in practice, unlike more general operations, such as DCJ or SCJ. In experimental studies there is a predominance of reversals over other events [21], [22], so the rearrangements in our simulations are distributed as 90% reversals and 10% translocations by default.

5.1.2 Campanulaceae Chloroplast DNA Dataset

To compare the SCJ method with existing ones, we applied SCJ to a dataset of chloroplast genomes from the flowering plant family Campanulaceae. This dataset was created by Cosner et al. [23] as a test case for their MPBE method. This is a well-studied dataset, consisting of 13 species (12 Campanulaceae and the outgroup Tobacco), where the genomes have one circular chromosome with 105 markers. The tree inferred by MPBE is shown in Fig. 5a.

In addition to comparing our SCJ methods with MPBE, we will also use the topology proposed by Bourque and Pevzner [11], who found a phylogeny with 65 reversals using MGR (see Fig. 5b). Later, the same

TABLE 1
Parameter Values

Parameter		Value Range	Default
Tree	Branch Length	{0.05, 0.10, 0.15, 0.20, 0.25}	0.2
	Leaves	$\{12, 32, 64, 128, 200\}$	$\{64, 12\}$[1]
Genome	Genes	{500, 1000, 1500, 2000, 2500, 3000}	2000
	Chromosomes	$\{1, 5, 10, 15, 20\}$	5
Rearrangement Distribution (Reversal, Transposition Translocation)		{(0.2, 0.0, 0.8), (0.2, 0.1, 0.7), (0.4, 0.0, 0.6), (0.4, 0.1, 0.5), (0.6, 0.0, 0.4), (0.6, 0.1, 0.3), (0.8, 0.0, 0.2), (0.8, 0.1, 0.1), (0.9, 0.0, 0.1), (1.0, 0.0, 0.0), (0.0, 1.0, 0.0), (0.0, 0.0, 1.0)}	(0.9,0.0,0.1)

[1] The default number of leaves depends on the method: 64 genomes for Fitch and stepwise addition methods, and only 12 genomes for branch-and-bound, because of its prohibitive computation time for larger data sets.

topology was used by Adam and Sankoff [24] to solve SPP under the DCJ model, obtaining a tree with 64 DCJs. In 2010, Kováč et al. [25] also used this tree to solve SPP under DCJ, but penalizing multiple chromosomes. They obtained several topologies with 59 DCJs, where all the ancestors have a single chromosome.

Another topology considered for the comparison was presented by Xu and Moret [22]. Recently, they have proposed a method called GASTS to solve the SPP, and found 294 trees with DCJ score equal to 63. These topologies are different from the topology obtained by MGR. From these trees, we collapsed interior branches of zero length, where no DCJ operation was done, and a consensus tree was built by PHYLIP 3.69 [26] (CONSENSE module). This GASTS consensus tree is shown in Fig. 5d.

The last tree to be compared was obtained by Cosner et al. [27], using 18 species of Campanulaceae and the outgroup Tobacco. We excluded from their trees six species that are not contained in our dataset. Cosner et. al conducted an extensive study of Campanulaceae phylogeny, using supplementary input data: sequence data from the rbcL gene and the ITS region, and three character matrices based on gene order. The authors present seven trees, but we chose the one which better represents the results discussed in their paper, in our opinion (Fig. 5c).

For the SCJ tree, we used the branch-and-bound method to find all optimal trees (4 in total), then calculated the consensus tree using PHYLIP. The resulting tree is shown in Fig. 5e.

5.1.3 Protostome Mitochondrial DNA Dataset

Besides the Campanulaceae dataset, we also used a larger dataset, containing 66 protostome mitochondrial DNAs, with 36 genes each, published by Fritzch et al. [28] as a test case for their alignment-based approach. After aligning the genomes, the phylogeny was inferred with maximum parsimony methods, such as stepwise addition and branch swapping heuristics. The resulting tree is shown in Fig. 6b. They used 112 genomes with 37 genes, but the dataset contains duplicated genomes, duplicated genes, and indels. Therefore, we applied a treatment similar to the one used by Bernt et al. [29], who obtained a smaller subset of 62 genomes and 36 genes. From the collection presented by Fritzch, we handled genomes with unequal gene content using the following steps:

1) Removed all duplicated genomes, leaving 78 genomes;
2) From these 78 genomes, we removed all genomes with duplicated genes (6 in total);
3) Removed the ATP8 gene. Before the removal, there were 18 genomes with unequal gene content, and only 6 after the removal;
4) Removed the 6 genomes that remained with unequal gene content.

We then executed the stepwise addition heuristic 100 times, saving the tree with the fewest SCJ operations, presented in Fig. 6c. We also used in our comparison the NCBI taxonomy tree (National Center for Biotechnology Information), shown in Fig. 6a. These species have been studied with several other methods. The phylogeny of protostomes based on rearrangements was first presented by Blanchette et al. [30], using a different dataset.

5.2 Phylogenetic Reconstruction Methods

5.2.1 Small Parsimony

To solve the SPP, we implemented the polynomial algorithm proposed by Feijão and Meidanis [12]. In broadest outline, this method is analogous to Fitch's algorithm [31], where each adjacency is a character and the possible states are presence or absence of a specific adjacency. A key advantage of SCJ, which sets it apart from other techniques, is that the SPP is easy, solvable in polynomial time, while in other models — like DCJ and Hannenhalli-Pevzner (HP) — it is NP-hard even when only three genomes are considered (the special case of SPP called Median Problem) [4].

5.2.2 Big Parsimony

To solve the BPP, we used two methods: an exact, branch-and-bound method and a greedy heuristic, called stepwise addition.

The stepwise addition heuristic is similar to previously used heuristics for this problem [32, pp. 216]. An outline of this algorithm is shown in Algorithm 1. It starts by solving a Median Problem with three of the input genomes and, at each step, an arbitrary unplaced genome is added to the tree, by solving a median problem at each edge. The genome is added to the tree in the edge that incurs the minimum branch length. After all genomes have been placed, the last step is to solve a SPP on the final tree, since the median solving during the iterative step does not guarantee that the internal nodes are optimal for the resulting tree. The SPP is solved in polynomial time, guaranteeing a minimum weight assignment of internal nodes on the tree, a major advantage of SCJ in comparison with other rearrangement distances, where the SPP is NP-hard.

When the size of the problem allows the search for an exact solution (typically 12 genomes or less), we also used a branch-and-bound algorithm. Branch-and-bound algorithms have been used for a long time in phylogenetic reconstruction [33]. As in the heuristic, we build a starting tree by solving a Median Problem with three of the input genomes and extensively build the whole space of possible phylogenetic trees.

In the search tree, the algorithm performs the operations of branching and bounding as follows. The branching operation constructs a child of a node based on its partial solution $P = (V^P, E^P)$ and the list L of the genomes not included in P. An arbitrary genome l

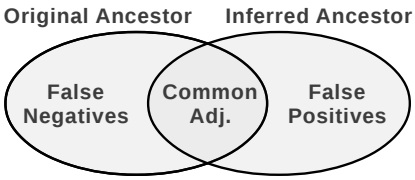

Fig. 3. Adjacencies of original and inferred ancestors. For each subset, there is a corresponding metric.

belonging to L, and an edge (i, j) of E^P are chosen, and the algorithm computes the median M of genomes i, j, and l, defining a new tree by removing edge (i, j) and including instead edges (i, M), (M, j), and (l, M). The children of a node are obtained applying the previous step for each edge (i, j) of E^P, generating all possible trees that can be obtained from the addition of l in the partial solution P.

The bounding operation initially computes an upper bound using neighbor-joining [34], and improves this upper bound when a better tree is found. For the lower bound of a node, the algorithm uses its partial solution $P = (V^P, E^P)$ and the list L of the genomes not included in P, constructing a new tree for each genome l belonging to L and each edge (i, j) of E^P, in the same way as described in the branching operation. From these trees, the algorithm finds the minimum cost that each genome of L adds to the tree, using as lower bound the greatest of them, which represents the lowest cost to include the most distant genome. Notice that the calculation of the tree cost is very fast, because SCJ provides for a polynomial algorithm for this task.

Finding the optimum trees is useful, for example, when we want to know whether SCJ infers a realistic tree structure.

Algorithm 1 Stepwise Addition Heuristic

Require: Genomes $\Pi_1, \Pi_2, \ldots, \Pi_n$
Ensure: Minimum weight phylogeny with leaves $\Pi_1, \Pi_2, \ldots, \Pi_n$

1: solve the median problem for Π_1, Π_2, Π_3 and call $\mathcal{T}$ the resulting tree;
2: **for** $l := 4$ to n **do**
3: **for** each edge $\{u, v\}$ in $\mathcal{T}$ with labels Π^u, Π^v **do**
4: compute a median Π^{uv}_M of Π^u, Π^v, Π_l
5: $C(u, v) := d(\Pi^u, \Pi^{uv}_M) + d(\Pi^v, \Pi^{uv}_M) + d(\Pi_l, \Pi^{uv}_M) - d(\Pi^u, \Pi^v)$
6: **end for**
7: $C\{u_0, v_0\} := \min\{C(u, v) | \{u, v\} \in E(\mathcal{T})\}$
8: remove edge $\{u_0, v_0\}$ from $\mathcal{T}$
9: add vertices Π^{uv}_M, Π_l to $\mathcal{T}$
10: add edges $\{\Pi^{uv}_M, u_0\}, \{\Pi^{uv}_M, v_0\}, \{\Pi^{uv}_M, \Pi_l\}$ to $\mathcal{T}$
11: **end for**
12: Solve the small parsimony problem on $\mathcal{T}$
13: **return** the tree $\mathcal{T}$

5.3 Comparison Strategies

Using the metrics detailed below, we analyze the main aspects of these reconstructions: tree structure and ancestral genomes. Besides accuracy, we analyze the efficiency of each algorithm.

To assess the structural quality of an inferred phylogenetic tree, we used the Robinson-Foulds distance [35], also known as split distance. Given a tree, the removal of an edge partitions the leaves into two disjoint subsets, forming what is called a split. The split distance is the difference in splits between the original and reconstructed trees, divided by the total number of splits. Split distance varies from 0 to 1, where 0 represents a "best case" scenario, when all splits are equal. We use the TOPD program [36] to compute the split distance.

We compare the ancestors considering their gene adjacencies. Given an original and an inferred genome, we categorize their adjacencies in three classes (see Fig. 3):

- False negatives: adjacencies present in the original ancestor, but not in the inferred ancestor;
- Adjacencies in common: present in both genomes, they are used to calculate the percentage of reconstruction of the original genome;
- False positives: incorrect adjacencies, present only in the inferred genome.

The percentage of reconstruction is defined as the number of adjacencies in common divided by the total number of adjacencies of the original genome. This metric is related to CARs (Contiguous Ancestral Regions) [21], but not identical. The CARs are different because they also consider the relationship between the adjacencies, trying to model orthology blocks. Based on the predicted ancestral adjacencies, the genes are connected into CARs if their predecessor and successor relationships are consistent with the original tree. Another way of analyzing the percentage of reconstruction is to obtain the percentage of the original genome covered by the CARs.

Let A and B the set of adjacencies of the original and inferred genome, respectively. The false positives metric is defined as $\frac{|B-A|}{|B|}$, and similarly the false negatives metric is determined by $\frac{|A-B|}{|A|}$. The false negatives set is the complement of the common adjacencies in the original genome, and for this reason the false negatives graphs are not shown in our analysis.

It is unfeasible to show the metrics for all nodes. To help understand comparison results on ancestral genomes, we decided to correlate them with the node position in the tree. This seems particularly appropriate for percentage of reconstruction, because all known information is in the leaves, and therefore more reconstruction effort is needed as we move up the tree. We considered several alternatives for a positional metric. The following measures were candidates for correlating the percentage of reconstruction: height (longest downward path from the node to a leaf), depth of (longest downward path from the root to the node), average distance

to a leaf, and minimum distance to a leaf. We ended up choosing height, because the standard deviations for the other measures were significantly higher.

Processing times were collected for all tests, and analyzed in the efficiency section (Section 6.3).

6 Results

6.1 Topology Accuracy

In this section, we present the results on topology accuracy under different evolutionary scenarios. We start by discussing the influence of simulation parameters on the results. We also assess the accuracy of methods: how does the heuristic approach the optimum result? And how does the optimum result approach the original tree? For data where we do not know the optimum (all those with more than 12 species), we compare the heuristic to the original tree directly. At the end of this section, we consider the results on real data, comparing the inferred topology with the topologies proposed in other studies.

We show below BPP results only. The SPP is not considered because the topology is given in this case. We use the methods described in Section 5.2 (branch-and-bound and stepwise addition heuristic) to solve the problem.

6.1.1 Simulated Dataset

For this section, we used as input the simulated datasets described in Section 5.1.1.

The graphs in Fig. 4 show how the deviation of the inferred topology (y axis) changes when we vary one parameter in the problem input (x axis). Each point on the graph has a histogram with the RF distance distribution, where a bar of histogram represents the frequency that a certain range of RF distance values was observed (see graph caption).

We can see in Figs. 4a and 4b that the average RF distance is not significantly affected by the number of genes, because the distance seems to remain nearly constant in both methods (exact and heuristic). In terms of the distribution, the SCJ reconstruction leads to the original tree structure in more than 50% of the simulations (RF distance equals zero). In the histograms in Fig. 4a, it is clear that inferences with the exact method have a lower probability of error, and that this probability decreases as the number of genes increases. A similar behavior is seen when the number of chromosomes increases (data not shown).

With the heuristic, the average RF distance is around 0.35, considerably higher than in the exact method (0.05). Keep in mind, though, that the exact method was tested on just 12 leaves.

How does the quality of the inferred topology vary when we use different rearrangement distributions? The graphs in Figs. 4c and 4d show an interesting result: when the frequency of each event increases or decreases relative to others, the accuracy of the inference is not affected. Probably this is because the same rearrangement distribution is used in all edges of the tree, being a common term. More specifically, let $n_{A,B}$ be the number of rearrangement events between two genomes A and B, and let $F = [f_\alpha, f_\beta, f_\gamma]$ be the relative frequencies of reversals, transpositions, and translocations. The number of events of each type between A and B is expected to be $n_{A,B} * [f_\alpha, f_\beta, f_\gamma]$.

If the number of SCJ operations required for each event is $S = [s_\alpha, s_\beta, s_\gamma]$, then the expected SCJ distance between A and B is:

$$d_{A,B} = n_{A,B}(f_\alpha s_\alpha + f_\beta s_\beta + f_\gamma s_\gamma).$$

That is, $d_{A,B}$ is roughly proportional to $n_{A,B}$. These quick calculations seem to imply that the *number* of events (branch length) matters much more than the *type* of event (rearrangement distribution).

The impact of branch length is shown in Figs. 4e and 4f. In Fig. 4e, the RF distance is similar to that of previous experiments (Figs. 4a and 4c). In Fig. 4f, which presents the results of the heuristic for 64 leaves, we see a decrease in accuracy with larger branch lengths.

In a similar study, Bernt et al. [29] compared three heuristic methods that search the most parsimonious reversal scenarios: amGRP, GRAPPA and MGR. In their experiment, they fixed the tree size in 50 leaves and the genome size in 50 genes, varying the branch length with the values {0.08, 0.12, 0.16}. Among the three methods mentioned, amGRP achieved better results both in terms of accuracy and efficiency, with RF distances varying between 0 and 0.05, while GRAPPA and MGR obtained RF distances between 0 and 0.15. Using a similar tree size (64 leaves) and considerably larger genomes (2000 genes), the SCJ heuristic obtained a relatively higher deviation, with RF distances from 0.2 up to 0.3 for the same variation of branch length (see Fig.4f). Nevertheless, the SCJ heuristic is much faster, solving BPP with large genomes (2000 genes) in just a few minutes, while the previous methods show a steep growth in the time required to compute the response, requiring a couple of hours even with genomes containing only 50 genes. Therefore, compared to other heuristics, the stepwise addition heuristic under SCJ is less accurate, but faster.

To summarize, it seems that tree size (measured by the number of leaves) and branch length are the only parameters that impact SCJ inference: both parameters increase the RF distance. We fitted a linear model with x (the parameter) and $\log y$, where y is the average RF distance, and found good correlations in all cases (data in Figs. 4e, 4f, and 4g). Our results agree with the study of Nakhleh et al. [37], where they used different distance-based methods to infer topologies and, despite their methods being less accurate, they also noticed a logarithmic growth of the RF distance when the number of leaves increases.

Our results show that SCJ inference by the exact method is more accurate, even with a high number of

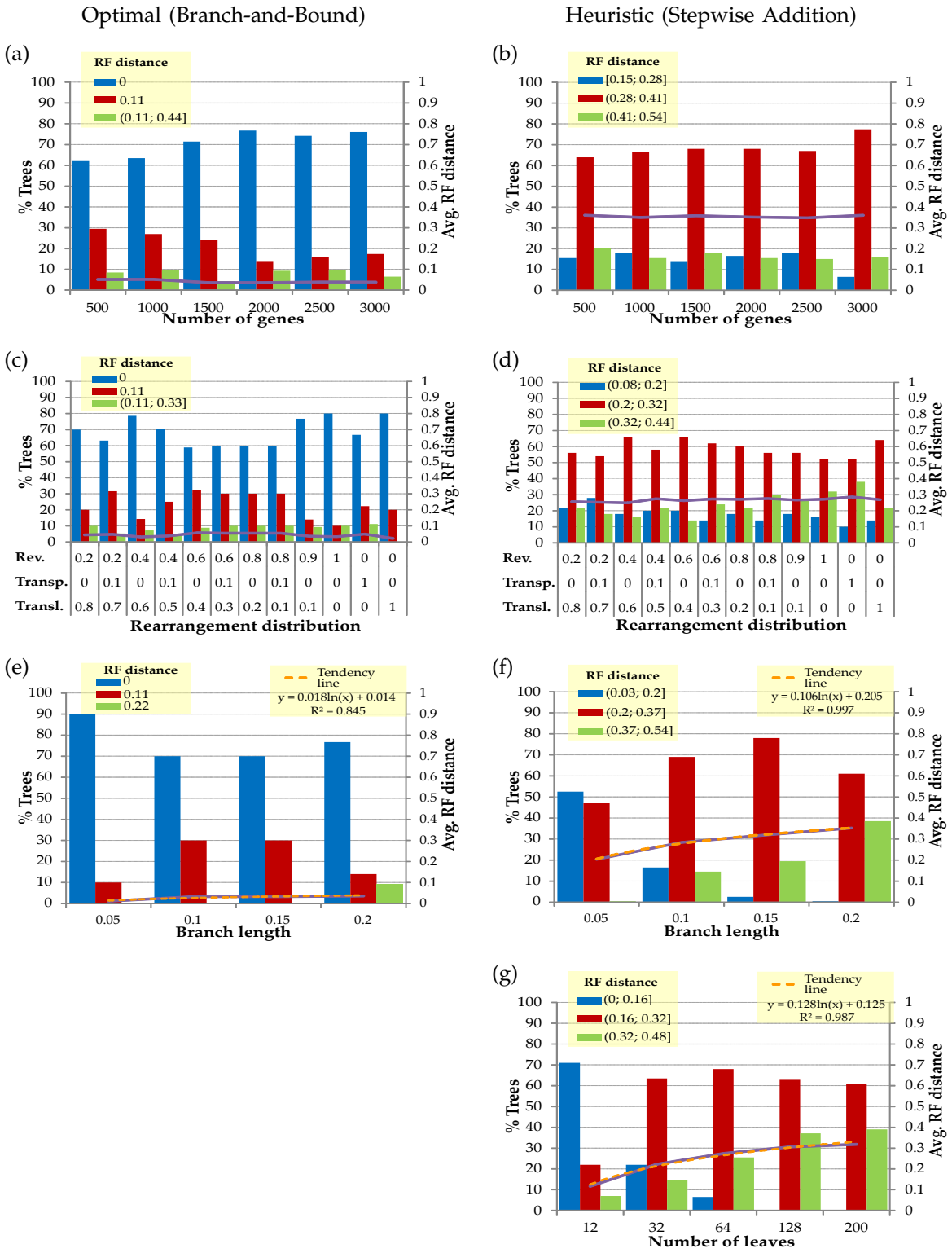

Fig. 4. Parameter influence on inferred topology, for BPP methods optimal and heuristic: (a–b), influence of genome size (number of genes); (c–d), influence of rearrangement distribution; (e–f), influence of branch length (percent of change); (g), influence of number of leaves. In each graph, when a parameter is being analyzed, all others retain their default values: 2000 genes (5 chromosomes of 400 genes each); (0.9, 0.0, 0.1) rearrangement distribution among reversals, transpositions, and translocations, respectively; and 0.2 for branch length. With respect to number of leaves, the default is 64 for the heuristic, but only 12 for the optimal method, because of time constraints, and there is no influence analysis for this method regarding number of leaves. All graphs have a double y axis: RF distribution histogram (left) and average RF distance (right). Average RF distances in (e), (f), and (g) were fitted with a log curve. In both cases (optimal and heuristic), the RF distance is computed with respect to the correct tree.

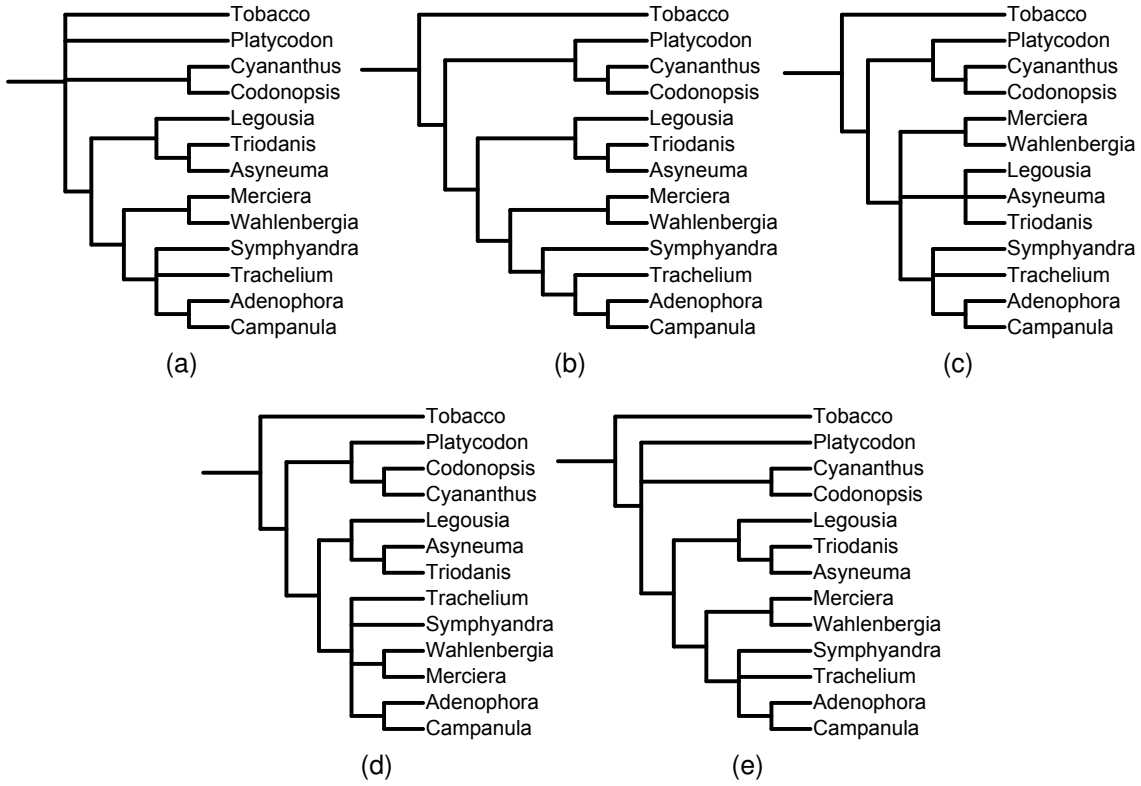

Fig. 5. Campanulaceae topology, as reconstructed by several methods: (a) MPBE [23]; (b) MGR [11]; (c) Cosner et al. [27]; (d) GASTS [22]; (e) SCJ. All methods, except Cosner et al., used as input the chloroplast genomes of 12 species of Campanulaceae and the outgroup Tobacco. Cosner et al. used a larger dataset based on gene order and sequence data, composed of 18 species of Campanulaceae and the outgroup Tobacco. Both SCJ and MPBE reconstructions are based on breakpoints, unlike the MGR method, which is based on reversals. The GASTS method solves SPP minimizing the number of DCJs. The analysis of Cosner et al. was based on parsimony, using breakpoints, events defined by breakpoints, and sequence data from the ITS region and the rbcL gene. In general, the topology obtained by SCJ agrees with the others, revealing some tendencies, as described in the main text.

evolutionary events (up to 20 percent of the number of genes in each edge). The heuristic method provides somewhat less accurate results, but it is able to solve much larger instances.

6.1.2 Real Dataset: Campanulaceae

Regarding real data sets, we ran the SCJ methods on Campanulaceae chloroplast genomes, and visually compared them with the trees presented by Cosner et al. [27], and also with the trees inferred by MGR, MPBE, and GASTS. Fig. 5 shows the five topologies used in this comparison, drawn with the online tool iTOL (Interactive Tree Of Life) [38]. We found all optimal trees (four in total) under the SCJ model. The four trees are very similar, except for two evolutionary relationships that were not fully resolved:

1) Relationship between Platycodon and the ancestor of Codonopsis and Cyananthus: in two trees, Platycodon is farther from Codonopsis and Cyananthus, as shown in Fig. 5e, while in the other two, the relationship is equal to the tree used by GASTS (Fig. 5d).
2) Relationship between Symphyandra, Trachelium, and the ancestor of Campanula and Adenophora: in two trees, Symphyandra is closer to Campanula and Adenophora than Trachelium (Fig. 5e), while in the other two, the opposite situation appears, as in the topology inferred by MGR (Fig. 5b).

In general, all trees are very similar. Looking at the trees mentioned above, it is possible to identify common relationships, listed below and well represented by the tree in Fig. 5c:

- **Small groups:**
  - **Group 1:** Campanula and Adenophora are siblings in all results;
  - **Group 2:** Merciera and Wahlenbergia are siblings in all results;
  - **Group 3:** Legousia, Asyneuma, and Triodanis form a subtree in all results;
- **Medium groups:**
  - **Group 4:** Platycodon, Codonopsis, and Cyananthus are always clustered together, which agrees with the evidence of them being basal within the family, presented in [27];

  - **Group 5:** formed by the groups 1, 2 and 3, and also by the Trachelium and Symphyandra genera;
- **Large group:**
  - **Group 6:** formed by the groups 4 and 5, represents the Campanulaceae family.

This shows that SCJ is capable of delivering high quality reconstructed trees from real datasets.

6.1.3 Real Dataset: Protostomes

In the case of the Protostome dataset, we ran the heuristic method 100 times, saving the trees with minimum total number of SCJ operations (3 in total). Then, a consensus tree was computed by PHYLIP, and the result is shown in Fig. 6c. We compare it to the NCBI topology (Fig 6a) and to the results of Fritzsch et al. [28] (Fig 6b).

Our approach correctly classified the species within the phyla Arthropoda, Nematoda, Echinodermata, Annelida, and Platyhelminthes. Mollusca presented more of a problem, which is justified by the observation that some Mollusca have higher frequencies of gene rearrangements in mitochondrial genomes, compared to other metazoans [28], rendering topology inference based solely on gene order difficult. Our approach was, however, able to classify more species into phyla, because the tree presented by Fritzsch et al. does not fully resolve Nematoda (see Fig. 6b).

Using either gene order or sequence data, neither Fritzch et al. nor SCJ classify the species in the subphylum of Arthropoda (Myriapoda, Chelicerata, Crustacea, and Hexapoda), disagreeing with the classifications based on morphological characteristics (data not shown). The taxonomy of the Arthropoda is not fully resolved, due to the diversity and size of this phylum.

Although the SCJ method correctly solves species in phyla, the evolutionary relationships between these phyla are quite different from that obtained by Fritzch et al. Comparing these trees with the NCBI taxonomy tree, we recognize three well-defined groups (see Fig. 6a):

- **Group 1:** Echinodermata;
- **Group 2:** Platyhelminthes and Nematoda;
- **Group 3:** Arthropoda, Annelida, and Mollusca.

The tree obtained by SCJ agrees in Groups 1 and 3, but not in Group 2. The species of the phylum Mollusca appear in various parts of the tree. In the tree of Fritzch et al., the phyla Annelida and Nematoda are monophyletic, and Platyhelminths are closer to the subphylum Gastropoda (Mollusca). Again, this shows that SCJ is capable of producing trees compatible with well-accepted ones.

6.2 Ancestral Genome Accuracy

Previously we analyzed how well the inferred topology agrees with the expected results. In this section we focus on the ancestral genomes inferred from simulated data.

To ensure that all inferred ancestors have a counterpart in the original tree, the topologies of inferred and original trees must be the same. Therefore we study the Small Parsimony Problem only, where the topology is given. In the Big Parsimony Problem there is no guarantee that the inferred topology will be the same as in the original tree.

To assess the quality of the reconstruction, we begin with a simple measure, which is the relative amount of adjacencies between consecutive genes that were recovered by the SCJ method. We call this amount the *percentage of reconstruction* below.

The line graphs in Fig. 7a show the correlation between the percentage of reconstruction of the original genome (y axis) and the height of the node in inferred tree (x axis). Each line corresponds to one specific rearrangement distribution.

We can observe a lower percentage of reconstruction in the nodes that have larger height in the tree (closer to root, farther from the leaves). We can see in the graphs that, regardless of the variation in the rearrangement distribution, the decrease in the percentage of reconstruction is similar in all lines of the graph, once again showing that the rearrangement distribution tends to impact negligibly the accuracy of SCJ methods. The genome size (number of genes and chromosomes) exhibits a similar behavior (not shown).

In all graphs in this section, the dispersion of the lines is bigger when the height increases. The dispersion increases because the number of nodes examined is lower: if the tree is balanced, the number of nodes from a height i is approximately the sum of all nodes with bigger heights; if the tree is unbalanced, the root will achieve a higher height. For example, in Fig. 7a we notice that the tree height reaches up to 25, whereas a balanced tree with 64 leaves has 6 as maximum height.

As in the analysis of topology, the parameter that affects the inference more strongly is the number of events on each edge. In Fig. 7b, the lines corresponding to larger branch lengths drop more sharply. Notice that, in experiments with rearrangement rates of up to 15%, all genomes had at least half of the genome reconstructed, regardless of height, which is a very good result.

The lines in Fig. 7c represent different numbers of leaves. Since trees with more leaves have a larger number of nodes, they can reach bigger heights, producing more elongated lines on the graph. The number of leaves affects how the reconstruction percentage decreases, just as in the case of Fig. 7a.

6.2.1 False positives

Considering all inferred adjacencies, the percentage of reconstruction defines how many adjacencies are correct, that is, are present in both genomes. Now let us look at the inferred adjacencies that are not present in the original genome, which we call false positives.

Fig. 8a shows the correlation between the number of false positives and the number of genes. These histograms classify the number of ancestral genomes ac-

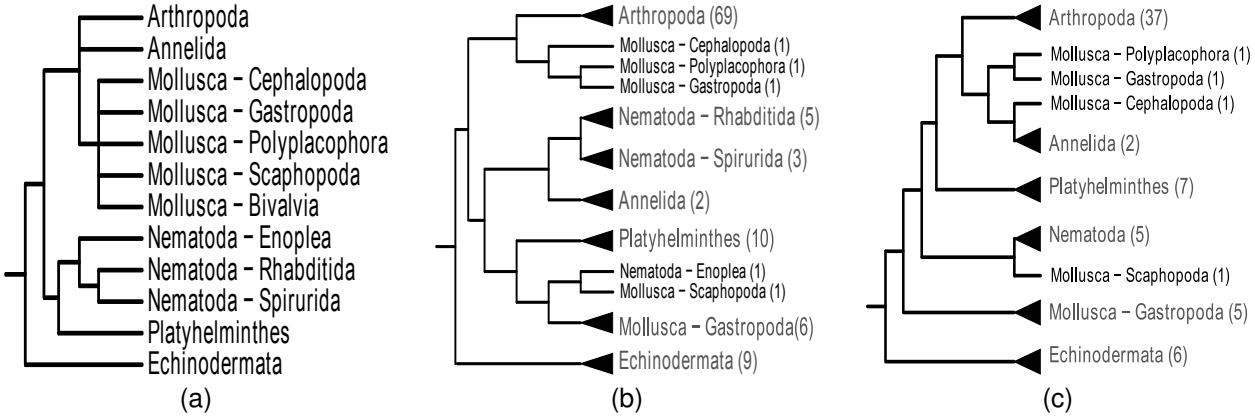

Fig. 6. Protostomes topology, as reconstructed by several methods: (a) NCBI; (b) Fritzsch et al. [28]; (c) SCJ. Note that the SCJ method correctly grouped all phyla, except the phylum Mollusca, which is highly rearranged and was also a problem in the inference of Fritzsch et al. The tree obtained by SCJ is closer to the tree obtained from the NCBI taxonomy, with a closer relationship between Annelida and Arthropoda.

cording to the number of incorrect adjacencies. We notice that 90% or more of the ancestral genomes have all their adjacencies reconstructed, even when we increase the number of genes. This same behavior is observed by varying the number of leaves (Fig. 8e), indicating no significant influence of these parameters on the number of false positives.

We also identify a second type of behavior, shown in Fig. 8b and Fig. 8d: when either the number of chromosomes or the rate of rearrangement increases, the number of false positives also increases. Ancestral genomes without false positives dominate in all histograms. In the case of branch length, we notice that this behavior is due to an increased chance of the original tree not being a most parsimonious tree, i.e., when more evolutionary events occur, a given region of the genome may be similar in species of different ancestry (as homoplasies). Therefore, it is possible to explain the evolution with a smaller number of events, if the common ancestors of these branches are also similar.

Finally, there is a third type of behavior observed in Fig. 8c: using different rearrangement distributions, we find a variation in the number of false positives, which are distributed in several ways. When there is only one type of event during evolution, the number of false positives is significantly higher than in evolutionary scenarios with more diverse rearrangements. In the case of just transpositions, it happens probably because the transposition event needs more SCJ operations, when compared to reversals and translocations. This event is therefore heavier, effectively increasing the branch length by 50%.

6.3 Efficiency

As far as running time is concerned, we distributed the experiments in different environments, according to the computational effort required to solve them.

Given a topology, the time SCJ takes to reconstruct the ancestors in a tree is about 3.3 seconds. The experiments were performed on a 2.67GHz Intel Core i5 processor, with 6GB RAM. We implemented a sequential version of this algorithm.

For both methods that solve BPP, we used parallel multithreading: the heuristic (stepwise addition) is repeated 100 times, distributed in three concurrent threads; the branch-and-bound algorithm is also run concurrently in three threads, sharing the bounds between them. Running the heuristic on the same computer described above, SCJ can reconstruct a topology of 66 protostomes in 3.9 seconds, while the topology of 64 genomes (simulated datasets) was obtained in 10.7 minutes, using computers with 4GB RAM and 2.40GHz Intel Core 2 Quad processors. The large time difference between them is due to the size of the input genomes: the simulated dataset has genomes with 2000 genes as default size, while the Protostome genomes have only 36 genes. We observe that both the number of genomes and the number of genes influence the efficiency of the heuristic.

The experiments solving BPP with the exact method spent significantly more time, due to the complexity of the problem. Because the Campanulaceae dataset is small (13 genomes), it was possible to run the exact method in about 6.9 hours, using a computer with 4GB RAM and an 2.40GHz Intel Core 2 Quad processor.

To improve the response time while maintaining the quality of results, we can use heuristics. The heuristic method has good accuracy when the set of genomes is small, often reaching the optimum value. Another alternative is to use distance-matrix methods, such as neighbor-joining, that usually infer the topology faster. However, SCJ methods, as well as other maximum parsimony approaches, provide more information about the evolutionary history, because they also infer ancestors.

For topology reconstruction with 12 genomes (simulated data), we used two different environments: one had 128 GB RAM and four IBM Power7 processors with 3.55GHz clock; and the other with 8GB RAM and 2.93GHz Intel Xeon.

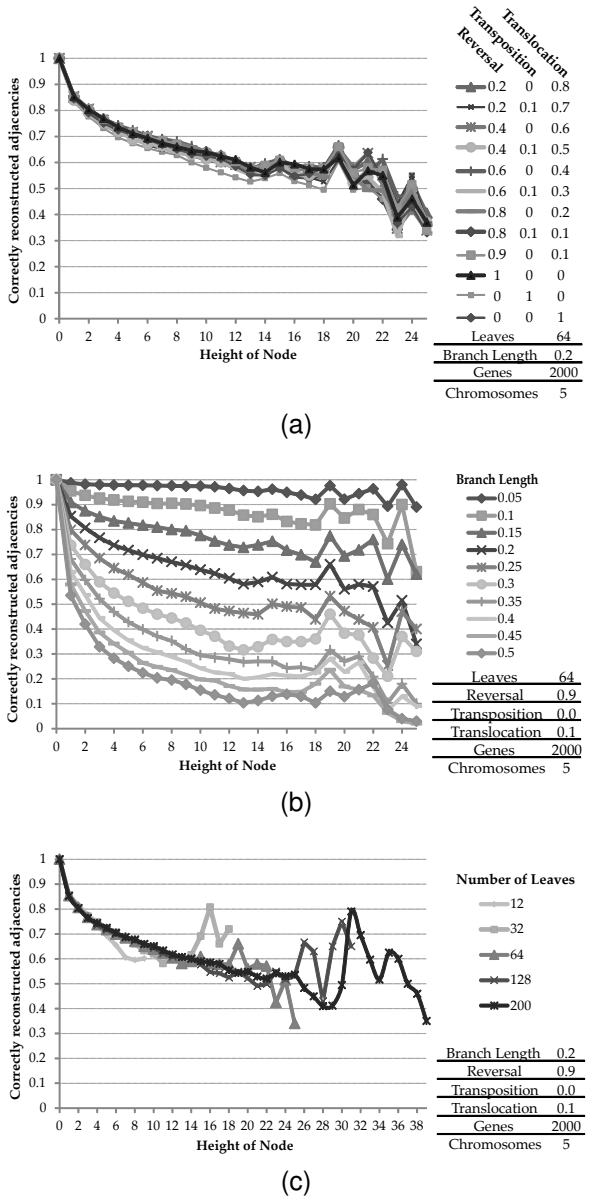

Fig. 7. Influence of parameters on ancestral genome reconstruction, considering only the percentage of correctly reconstructed adjacencies with respect to the total number of adjacencies in the original genome: (a) rearrangement distribution; (b) branch length; and (c) number of leaves. In all cases, the percentage of reconstructed genome decreases as node height increases. The number of nodes analyzed also decreases, yielding a greater data dispersion.

The experiments took time ranging from a few hours to up to 5 days, with an average time of 1.2 days. A more careful choice of upper and lower bounds may improve the performance. Note again the difference in time, caused by the size of the genomes, which heavily impact the complexity of calculating the median and other basic, frequently used operations.

7 Discussion and Future Directions

In this paper, we report on experiments to assess the ability of SCJ to reconstruct evolutionary histories, in two aspects: (1) how well does SCJ reconstruct evolutionary topologies, and (2) how well does SCJ reconstruct ancestral genomes.

It turns out that an SCJ-based heuristic is capable of recovering from 60% to 90% of the topology, as measured through the RF distance between original and reconstructed trees, in simulated data. This value was obtained with tests involving several random topologies, taken from a distribution closely approaching what we encounter in real life, and varying several parameters, including the amount of reversals, translocations, and transpositions on each branch, the number of input genomes (up to 200 genomes) and the number of genes (up to 3000 genes). We are not aware of other experiments done with datasets as large as these, other than using scaling methods such as the Disk Covering Method [39], which in turn could also be applied to SCJ, increasing its scalability even further. These large datasets were possible with SCJ because it yields extremely fast algorithms.

SCJ exact algorithms for the topology (Big Parsimony Problem) were also developed and tested. In this case, more than 95% of the topology can be recovered. However, this algorithm can only be used in smaller instances, up to 12 genomes, because of time considerations.

On real data, SCJ's ability to reconstruct tree topologies was also noteworthy. For the Campanulaceae dataset, SCJ was able to reconstruct the accepted topology, after averaging over several runs. In the Protostome dataset, SCJ was able to reconstruct several important clades, such as Arthropoda, Nematoda, Echinodermata, Annelida, and Platyhelminthes. Mollusca was a problem, but this clade is known to have especially difficult issues. The reconstruction achieved by SCJ was compatible with accepted trees reported in the literature.

With respect to ancestral genome reconstruction for a given topology, SCJ's success depends on how far from the leaves the ancestor is. For nodes close to the leaves, about 90% of the gene adjacencies can be recovered. This percentage decreases as we move up the tree, but, even at the root, about 50% of the adjacencies can be recovered. Our findings corroborate the fact that SCJ leads to very conservative genome reconstructions, yielding very few false positive gene adjacencies in the ancestors, at the expense of a relatively larger amount of false negatives.

We used this characteristic of SCJ to reconstruct very conserved gene clusters of arthropods and other clades, which may be of independent interest.

Code and data used in this paper can be found at:
www.ic.unicamp.br/∼meidanis/PUB/Proj/GRW

Acknowledgments

The authors would like to thank Brazilian research agency CNPq for scholarship grant 147990/2010-6 to

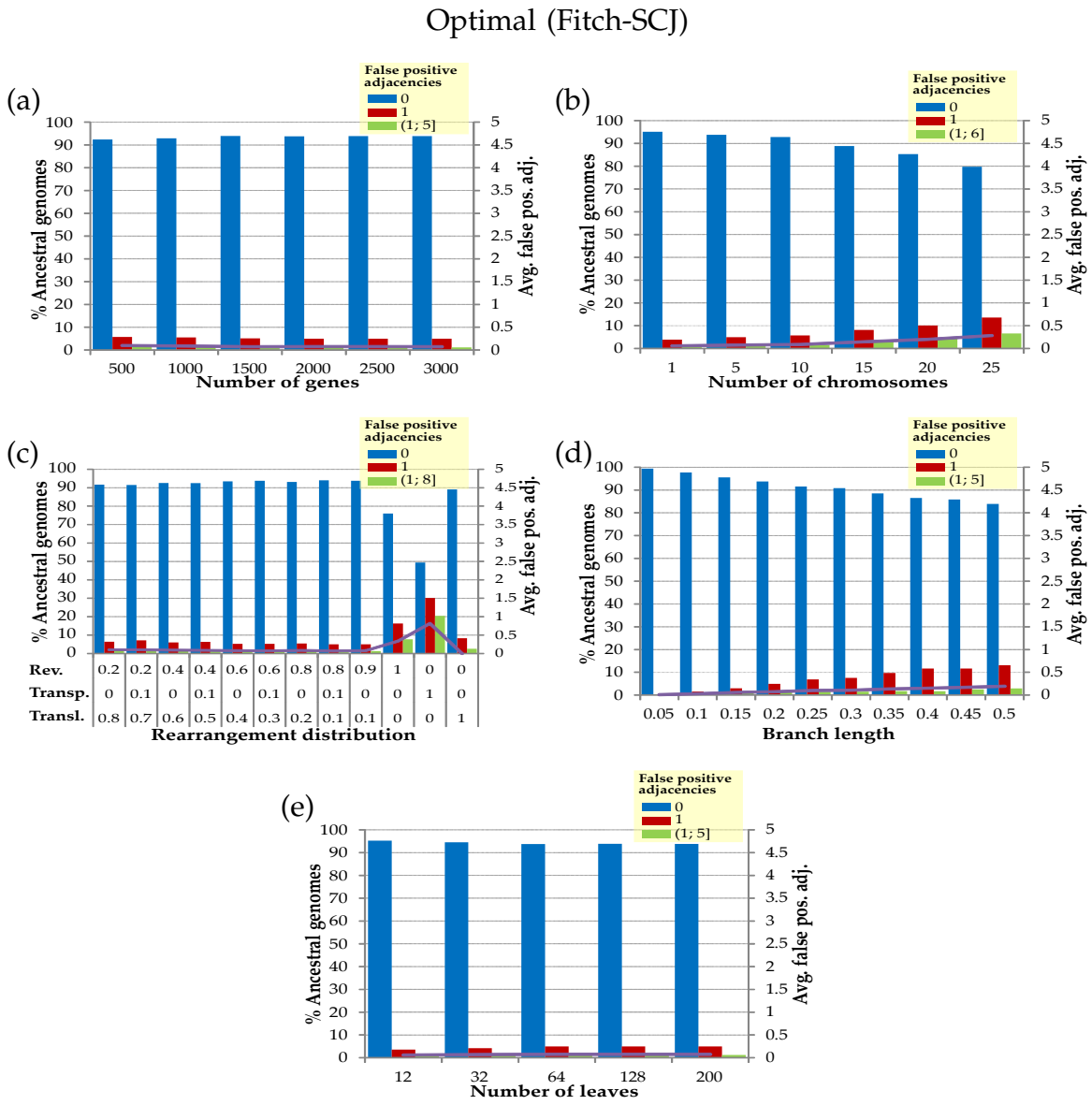

Fig. 8. Influence of parameters on ancestral genome reconstruction, considering only the false positive adjacencies. The false positive metric is defined as the number of reconstructed adjacencies not present in original genome divided by the total number of reconstructed adjacencies. The analyzed parameters are shown on the x axis: (a) number of genes; (b) number of chromosomes; (c) rearrangement distribution; (d) branch length; and (e) number of leaves. The left y axis plots the false positive distribution histogram and the right y axis plots the average of false positive adjacencies. In almost all cases, SCJ correctly infers the adjacencies, maintaining a small amount of false positives.

PB, Andrew Wei Xu and Bernard M. E. Moret for making available their GASTS phylogenies, the National Center for High Performance Computing in São Paulo (CENAPAD-SP) and the Laboratory of Optimization and Combinatorics (LOCo) of the University of Campinas for use of their computers, and Scylla Informatics for support.

REFERENCES

[1] A. H. Sturtevant and T. Dobzhansky, "Inversions in the third chromosome of wild races of *Drosophila pseudoobscura*, and their use in the study of the history of the species," *PNAS*, vol. 22, no. 7, pp. 448–450, 1936.

[2] B. McClintock, "The origin and behavior of mutable loci in maize," *PNAS*, vol. 36, no. 6, pp. 344–355, 1950.

[3] J. H. Nadeau and B. A. Taylor, "Lengths of chromosomal segments conserved since divergence of man and mouse," *PNAS*, vol. 81, no. 3, pp. 814–818, 1984.

[4] E. Tannier, C. Zheng, and D. Sankoff, "Multichromosomal median and halving problems under different genomic distances." *BMC Bioinformatics*, vol. 10, no. 1, p. 120, Apr 2009.

[5] S. Hannenhalli, C. Chappey, E. V. Koonin, and P. A. Pevzner, "Genome sequence comparison and scenarios for gene rearrangements: A test case," *Genomics*, vol. 30, no. 2, pp. 299–311, 1995.

[6] D. Sankoff, G. Sundaram, and J. D. Kececioglu, "Steiner points in the space of genome rearrangements," *Internat J Found Comput Sci*, vol. 7, no. 1, pp. 1–9, 1996.

[7] D. Sankoff and M. Blanchette, "Multiple genome rearrangement and breakpoint phylogeny." *J Comput Biol*, vol. 5, no. 3, pp. 555–570, 1998.

[8] B. M. Moret, L. S. Wang, T. Warnow, and S. K. Wyman, "New approaches for reconstructing phylogenies from gene order data." *Bioinformatics*, vol. 17 Suppl 1, pp. S165–S173, 2001.
[9] B. M. Moret, A. C. Siepel, J. Tang, and T. Liu, "Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data," in *Proc. WABI 2002*, ser. LNCS, vol. 2452, 2002, pp. 521–536.
[10] D. A. Bader, B. M. Moret, and M. Yan, "A linear-time algorithm for computing inversion distance between signed permutations with an experimental study." *J Comput Biol*, vol. 8, no. 5, pp. 483–491, 2001.
[11] G. Bourque and P. A. Pevzner, "Genome-scale evolution: reconstructing gene orders in the ancestral species." *Genome Res*, vol. 12, no. 1, pp. 26–36, 2002.
[12] P. Feijão and J. Meidanis, "SCJ: A breakpoint-like distance that simplifies several rearrangement problems," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 8, pp. 1318–1329, September 2011.
[13] A. Bergeron, J. Mixtacki, and J. Stoye, "A unifying view of genome rearrangements," in *Proc. WABI 2006*, ser. LNCS, vol. 4175, 2006, pp. 163–173.
[14] M. Blanchette, G. Bourque, and D. Sankoff, "Breakpoint phylogenies." *Genome Informatics*, vol. 8, pp. 25–34, 1997.
[15] J. Stoye and R. Wittler, "A unified approach for reconstructing ancient gene clusters," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 6, no. 3, pp. 387–400, 2009.
[16] J. Kováč, "On the complexity of rearrangement problems under the breakpoint distance," *Arxiv preprint arXiv:1112.2172*, pp. 1–15, 2011.
[17] L. R. Foulds and R. L. Graham, "The Steiner problem in phylogeny is NP-complete," *Adv. Applied Math.*, vol. 3, pp. 43–49, 1982.
[18] D. Aldous, "Stochastic models and descriptive statistics for phylogenetic trees, from Yule to today," *Statist. Sci.*, vol. 16, pp. 23–34, 2001.
[19] Y.-L. Huang, C.-C. Huang, C. Y. Tang, and C. L. Lu, "SoRT2: a tool for sorting genomes and reconstructing phylogenetic trees by reversals, generalized transpositions and translocations," *Nucleic Acids Research*, vol. 38, no. Web-Server-Issue, pp. 221–227, 2010.
[20] J. Shi and J. Tang, "An experimental evaluation of corrected inversion and DCJ distance metric through simulations," in *Proc. iCBBE 2010*, 2010, pp. 1–4.
[21] J. Ma, L. Zhang, B. B. Suh, B. J. Raney, R. C. Burhans, W. J. Kent, M. Blanchette, D. Haussler, and W. Miller, "Reconstructing contiguous regions of an ancestral genome." *Genome Res*, vol. 16, no. 12, pp. 1557–1565, Dec 2006.
[22] W. Xu and B. Moret, "GASTS: Parsimony scoring under rearrangements," in *Proc. WABI 2011*, ser. LNCS, vol. 6833, 2011, pp. 351–363.
[23] M. Cosner, R. Jansen, and B. Moret, *An Empirical Comparison of Phylogenetic Methods on Chloroplast Gene Order Data in Campanulaceae*. Kluwer Academic Publishers, Dordrecht, 2000, pp. 99–121.
[24] Z. Adam and D. Sankoff, "The ABCs of MGR with DCJ." *Evol Bioinform Online*, vol. 4, pp. 69–74, 2008.
[25] J. Kováč, B. Brejová, and T. Vinař, "A new approach to the small phylogeny problem," ArXiv e-prints, Tech. Rep., 2010.
[26] J. Felsenstein, *PHYLIP (Phylogeny Inference Package) version 3.69*, Distributed by the author, 2005.
[27] M. E. Cosner, L. A. Raubeson, and R. K. Jansen, "Chloroplast DNA rearrangements in Campanulaceae: phylogenetic utility of highly rearranged genomes." *BMC Evol Biol*, vol. 4, p. 27, 2004.
[28] G. Fritzsch, M. Schlegel, and P. F. Stadler, "Alignments of mitochondrial genome arrangements: applications to metazoan phylogeny." *J Theor Biol*, vol. 240, no. 4, pp. 511–520, Jun 2006.
[29] M. Bernt, D. Merkle, and M. Middendorf, "Using median sets for inferring phylogenetic trees." *Bioinformatics*, vol. 23, no. 2, pp. e129–e135, 2007.
[30] M. Blanchette, T. Kunisawa, and D. Sankoff, "Gene order breakpoint evidence in animal mitochondrial phylogeny." *J Mol Evol*, vol. 49, no. 2, pp. 193–203, Aug 1999.
[31] W. M. Fitch, "Toward defining the course of evolution: Minimum change for a specific tree topology," *Syst Zool*, vol. 20, pp. 406–416, 1971.
[32] G. Fertin, A. Labarre, I. Rusu, E. Tannier, and S. Vialette, *Combinatorics of Genome Rearrangements*, ser. Computational Molecular Biology. MIT Press, 2009, 312 pp.
[33] J. Felsenstein, *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates, 2004.
[34] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees." *Mol Biol Evol*, vol. 4, no. 4, pp. 406–425, Jul 1987.
[35] D. F. Robinson and L. R. Foulds, "Comparison of phylogenetic trees." *Math. Biosci.*, vol. 53, no. 1–2, pp. 131–147, 1981.
[36] P. Puigbò, S. Garcia-Vallvé, and J. O. McInerney, "TOPD/FMTS: a new software to compare phylogenetic trees." *Bioinformatics*, vol. 23, pp. 1556–1558, 2007.
[37] L. Nakhleh, B. Moret, U. Roshan, K. John, J. Sun, and T. Warnow, "The accuracy of fast phylogenetic methods for large datasets," in *Proc. of PSB'02*, 2002, pp. 211–222.
[38] I. Letunic and P. Bork, "Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation." *Bioinformatics*, vol. 23, pp. 127–128, 2007.
[39] J. Tang and B. M. E. Moret, "Scaling up accurate phylogenetic reconstruction from gene-order data." *Bioinformatics*, vol. 19 Suppl 1, pp. i305–i312, 2003.

Priscila Biller received a Bachelor's degree in Computer Science from the University of Campinas, Brazil, in 2009. She joined the Harpia project (Computational Intelligence Applied to Customs Risk Management) in 2008. She was a software test analyst at Sofist, and later worked on bioinformatics projects in a genomics laboratory. Currently, she is a PhD student at the Institute of Computing, University of Campinas, focusing on genome rearrangement problems.

Pedro Feijão completed his Applied Mathematics degree in 1997, at the University of Campinas, Brazil. He had his first contact with bioinformatics in 2004, when he started working in the Center of Molecular Biology and Genetic Engineering (CBMEG) on genome assembly projects and biological databases. He completed his PhD degree at the Institute of Computing, University of Campinas, in 2012, studying genome rearrangement models.

João Meidanis completed his PhD in Computer Sciences from the University of Wisconsin-Madison in 1992. He is a faculty member with the University of Campinas since 1986. He received the Science and Technology Medal from the State of São Paulo for his achievements in several Brazilian genome projects. His interests include computational biology, algorithms, and graph theory. Member of the Brazilian Computer Society.